23
29.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN

COMUNICACION EN PARALELO DE LA MICROCOMPUTADORA CROMEMCO Y EL MKE-Z80

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A N
ARMANDO RERGIS RAMIREZ
GERARDO MALDONADO TRUJILLO
PASCUAL ZARATE BANDA
ROBERTO MARTINEZ LOPEZ

DIRECTOR DE LA TESIS
ING. LUIS GARCIA GUTIERREZ

CUAUTITLAN IZCALLI, EDO. DE MEX. 1 9 8 7

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE

PROLOGO

El trabajo que se llevó a cabo en esta tésis consistió en elaborar un programa que satisface la necesidad para entablar comunicación en paralelo entre la microcomputadora CROMENCO y el KIT MKE-Z80. El SOFTWARE diseñado satisface la siguiente necesidad: el KIT es capaz de recibir archivos de programas ensamblados en hexadecimal de la CROMENCO archivados en disco.

La idea de realizar este tema de tésis, surge basada en la necesidad de obtener el máximo provecho del equipo existente en los laboratorios de electrónica. Actualmente se cuenta con un sistema CROMENCO y diez MKE-Z80. El trabajar en los KITS resulta muy difícil y tardado debido a lo inoperante que es el teclado de estos, por el contrario el trabajar en el sistema CROMENCO resulta más sencillo, por lo cual es de gran ayuda poder interconectar los KITS MKE-Z80 con el sistema CROMENCO.

El panorama actual de la tecnología en México nos muestra que el camino por el que se puede obtener mejores resultados es el diseño de tecnología apoyada en la ya existente. Esto es, se debe tratar de sacar el mayor provecho a los sistemas y equipos con que se cuenta tratando de perfeccionarlos ó adaptarlos de la mejor manera posible desarrollando una tecnología propia.

El país está urgido de tecnología avanzada dada la gran dependencia de tecnología extranjera. Es patente que nos vemos obligados a acudir a mercados de tecnologías con las cuales no se puede competir, esta tecnología, así como los conocimientos que provengan de sus investigaciones, se originan en una realidad diferente a la nuestra, por lo tanto no podrán satisfacer las necesidades del pueblo mexicano, que seguirá siendo un simple expectador en el acelerado progreso de la ciencia y la tecnología.

La situación económica por la que atraviesa el país, hace necesario concentrar la atención en el diseño y adaptación de sistemas y equipos ya existentes, este imperativo cobra especial importancia en la industria electrónica, en la que el suministro exterior tiene costos muy elevados.

Por lo tanto debemos de empezar a crear nuestra propia tecnología, necesaria para lograr una independencia económica, esto significa un reto para las tareas de investigación y desarrollo que deben producir resultados congruentes con las necesidades de la nación.

La tésis consta de once capítulos y una sección de apéndices, los cuales comprenden la teoría requerida para este trabajo. En los dos primeros capítulos, se tratan los

antecedentes históricos del desarrollo de la electrónica, así como de una reseña de la evolución de los sitemas computacionales y se tratan puntos de teoría básica sobre electrónica digital. El capitulo tercero y cuarto tratan sobre las diferentes formas ó métodos para entablar una comunicación de datos entre sistemas computacionales, así como de las técnicas existentes, para la detección y corrección de errores generados en dichas comunicaciones.

El capitulo quinto y sexto enuncian la definición y características de las diferentes interfaces de entrada y/o salida utilizadas en los sistemas computacionales. Se hace un estudio detallado de la interfaz 8255, porque este dispositivo fué el que se utilizó en la interconexión con el KIT. En el capitulo siete se estudia ampliamente el funcionamiento del microprocesador Z80A, su arquitectura y cada una de sus instrucciones para su correcta programación. En el capitulo octavo se presentan el Microkit Educacional MKE-Z80, generalidades, estructura, manejo y programación del mismo. El capitulo nueve, presenta un estudio del sistema CROMENCO, sistema operativo, estructura y programación. En el capitulo diez se presenta la teoría acerca del término "SOFTWARE", el cual es el programa ó microprogramación de los sistemas de cómputo.

Un análisis sobre la arquitectura ó dispositivos (HARDWARE) utilizados en el proyecto, es presentado en el

capitulo once. Para finalizar se presentan una serie de apéndices formados por tablas e información recibidas necesaria para apoyar los diferentes temas de esta tésis. Así como una guía y listado del programa de este proyecto.

Queda un buen margen de investigación para temas que surgen a lo largo del trabajo y que no se trataron con la amplitud que se quisiera, queda a juicio de los lectores, el estudio separado de los mismos, para un mejor aprovechamiento de esta tésis.

Hacemos patente nuestro agradecimiento a todas aquellas personas que nos apoyaron y colaboraron directa e indirectamente en la realización de este trabajo de tésis. Especialmente a el Ing. LUIS GARCIA GUTIERREZ. Al M. en C. JUAN ANTONIO NAVARRO MARTINEZ. Por sus aportaciones como maestros, guias y orientadores.

OBJETIVO :

DISEÑAR UN SISTEMA DE HARDWARE Y SOFTWARE, CON EL CUAL EL MICRO KIT EDUCACIONAL Z80 FUNCIONE COMO UN CONCENTRADOR DE INFORMACION

CAPITULO I

INTRODUCCION

Las matemáticas, desde tiempos remotos, han provisto al hombre de una gran herramienta, que le permite sistematizar y ordenar la solución de problemas en forma eficiente.

Conforme avanzaron las matemáticas, nació la ingeniería, que en un principio se dedicó sólo a medir áreas, distancias, deslindes, temperaturas, etc. La ingeniería evolucionó, hasta llegar con el paso del tiempo, a la existencia de una gran variedad de aplicaciones, para lo cual la ingeniería se dividió en varias ramas, las cuales varían de nombre según el tema, disciplina ó parte de las ciencias que se trate, como por ejemplo la ingeniería E.ectrónica, Mecánica, Civil, etc.

La ingeniería en general, es la rama de las ciencias, encargada de proporcionar a la humanidad, los instrumentos que emanen del análisis matemático, y de hacer uso de materiales y leyes naturales para solucionar un problema planteado.

La ingeniería Electrónica, nace a principios del

siglo XX, gracias a los recursos con que cuenta, puede decirse que estaba llamada a ser el vínculo entre otras ingenierías y las matemáticas.

La ingeniería electrónica se define como; la parte de la ingeniería que utilizando los distintos transductores, elementos y dispositivos, es capaz de realizar cualquier función matemática con señales eléctricas. Mediante el uso de transductores es posible cambiar cualquier variable (presión, temperatura, peso, flujo, velocidad, aceleración, pH, etc.) en señales eléctricas.

Un circuito electrónico, procesa la señal matemáticamente y de nuevo, mediante transductores apropiados, la transforma en variable ó variables de salida ó acción deseada. De esta manera se puede vislumbrar a la electrónica, como un vínculo entre las distintas ramas de las ciencias y la ingeniería.

La electrónica, es el estudio del comportamiento de los electrónes, en especial de sus movimientos en diversos campos de fuerzas a los que pueden ser sometidos como son: eléctrico, magnético, ó el efecto conjunto de ambos campos. En una forma simple la electrónica es el manejo de conexiones eléctricas y su control.

La palabra electrónica, proviene del vocablo griego electrón usado para nombrar el ámbar, ya que los antiguos griegos observaron que los trozos de ámbar atraían a otros cuerpos cuando se frotaban.

Actualmente se afirma, que la electrónica es la rama de las ciencias que ha evolucionado más rapidamente en lo que va del siglo. Esta evolución tan vertiginosa de la electrónica conjugada con las técnicas especiales, hará que pronto ningún país, ninguna sociedad por más aislada que esté, pueda sustraerse a este proceso y desarrollo de la ciencia.

Las aplicaciones de la electrónica, que en un principio se reducían a algunas comunicaciones por cable e inalámbricas, se han extendido de tal forma que resulta difícil encontrar en la actualidad alguna actividad industrial, comercial, educativa, ó de caracter doméstico, en la cual no exista algún elemento ó sistema en cuyo funcionamiento participe un circuito electrónico.

Prácticamente, el desarrollo de la electrónica se inicia alrededor de 1920, con la aparición de la radiodifución a gran escala, posteriormente la televisión acelera este desarrollo, pero el paso más importante, sin duda, es la aparición de los elementos llamados

semiconductores. En especial la creación del transistor, marca una de las etapas más importantes, en la evolución de la electrónica y en todas las aplicaciones de la misma.

Los materiales semiconductores, que hasta hace poco tiempo sólo interesaban a unos cuantos investigadores especializados, son en la actualidad de gran importancia práctica.

Anteriormente los materiales eléctricos se dividían en conductores y aislantes, según su resistencia, sin embargo se conocían materiales que no eran ni conductores ni aislantes por lo que se les llamó semiconductores. Más de un siglo de trabajo ha hecho posible demostrar que los semiconductores constituyen la clase más importante de los materiales eléctricos.

En su inicio, los semiconductores tuvieron un lento desarrollo, debido principalmente a dos razones: primero, que las técnicas para obtener los semiconductores monocristalinos eran muy limitadas y la pureza que se lograba, era muy inferior a la actual, y segundo, que la explicación del comportamiento de los semiconductores está basada en la teoría de la mecánica cuántica, la cual fue desarrollada en los años treinta.

La historia de los semiconductores, puede remontarse a principios del siglo pasado, cuando se descubrió que al pasar una corriente eléctrica a través de una barra de sulfuro de plata, ésta presentaba una conductividad baja, la cual se incrementaba al elevarse la temperatura. Debido a que este material conducía la corriente eléctrica mejor que un aislante, pero no tan bien como un conductor, se concluyó que debía existir un tercer grupo de materiales eléctricos, a los que se designó como semiconductores.

En 1873 se descubrió que haciendo incidir luz sobre selenio, su conductividad eléctrica aumentaba, y se encontraron sustancias que al estar en contacto con un metal tenían una resistencia que no obedecía a la ley de Ohm, y que conducía la corriente en un sólo sentido, a estos arreglos se les llamó rectificadores. A principios del presente siglo se logró realizar la rectificación de corriente alterna, utilizando rectificadores hechos de metal en contacto con carborundo (carbón de silicio), galeana (sulfato de plomo), silicio ó telurio, sin embargo con la invención del tubo al vacío (bulbo), que era más confiable y podía manejar mayor corriente, los semiconductores dejaron de utilizarse prácticamente.

Durante la segunda guerra mundial, se hicieron evidentes las limitaciones de los bulbos: su tamaño, su

fragilidad, el desgaste de material que producían, y su consumo de potencia, los volvían inadecuados para fabricar sistemas portátiles de comunicación, de radar y aéreos. Por esta razón los semiconductores se utilizaron nuevamente pero con más auge, como punto de partida se trató de encontrar los elementos semiconductores más adecuados para la fabricación de dispositivos electrónicos.

El fruto de estas investigaciones, fue la introducción al comercio de rectificadores, hechos de óxido de cobre recubierto de una capa de cobre ó de selenio con una aleación de estañocadmio, el cual tiene la propiedad de manejar potencias muy elevadas, razón por la cual todavía se utilizan. Posteriormente se descubrió, que el germanio y el silicio presentaban mejores características para utilizarlos en la fabricación de dispositivos semiconductores.

En 1948, Schockley, Bardeen y Brattain inventaron el transistor con el que empieza prácticamente la historia moderna de los semiconductores.

Historicamente, los primeros semiconductores que se utilizaron, fueron el selenio, el germanio y sobre todo el silicio. Inicialmente fue el germanio el material más utilizado en la elaboración de dispositivos semiconductores, y en la actualidad el silicio es el elemento principal en esa

actividad, se usan tambien compuestos semiconductores tales como: el arseniuro de galio (GaAs), el sulfuro de zinc (ZnS), el antimoniuro de indio (InSb), el fósforo de galio (GaP), los cuales tienen propiedades de semiconductores muy importantes.

El hecho de que finalmente se haya preferido, al silicio sobre el germanio en la industria electrónica se debe al considerar lo siguiente:

1.- El germanio presenta un defecto, su extrema sensibilidad a la temperatura.

2.- El silicio es el material que más se encuentra en todo el planeta, de hecho, es el segundo elemento más abundante después del oxigeno.

3.- La formación natural de una capa de dióxido de silicio en el proceso de oxidación, es la caracteristica que s aprovecha en el proceso de fabricación de dispositivos semiconductores.

Los semiconductores son, por lo tanto, el material base de los dispositivos electrónicos en miniatura (dispositivos microelectrónicos).

En 1948, la invención del transistor bipolar, en los laboratorios BELL, proporcionan un sustituto pequeño, barato y sencillo del tubo al vacio (bulbo), lo cual da un impetuo

renovado a la industria electrónica, este desarrollo continua con la integración de varios elementos en un circuito Impreso, que junto con la miniaturización de elementos pasivos, como son las resistencias y condensadores, esto condujo a la llamada microelectrónica, en la que en un material, se va construyendo dispositivos que reunen varios componentes activos y pasivos, conociendose esto como la tecnología planar, cuya invensión se da alrededor de los 60's, esta tecnología planar dio como resultado la aparición de los Circuitos Integrados (C.I.), inventados por la compañía Fairchil, lo que provocó una explosión industrial y un nacimiento de equipos sin precedente en el mercado.

Todo esto da como resultado una disminución en los costos de fabricación, tanto de los componentes, como de los equipos necesarios para esta fabricación. Y muchos progresos en lo que se refiere a su seguridad de funcionamiento, pues estos componentes tienen una mayor duración.

Desde que en 1904, cuando se creo el primer dispositivo electrónico, el tubo al vacio (bulbo), despues el transistor bipolar y el circuito integrado; de estos elementos desde el más simple hasta el más complejo, han tenido diversas repercusiones en ls vida de el hombre e incluso han afectado sus costumbres y hábitos sociales.

Con el circuito integrado, se tiene ahora un elemento funcional, que cambiará la filosofía clásica de el diseño, pues con el circuito integrado es posible lograr una considerable economía en las líneas de producción, ya que al manejar un número menor de componentes y disminuir el número de conexiones en el circuito, la eficiencia se incrementará y la posibilidades de error en el conexionado se reducen. La calidad y confiabilidad también resultan grandemente beneficiadas con el empleo de los circuitos integrados.

En unos cuantos años, los C.I. han experimentado un crecimiento explosivo, pasado de lo que en un principio se consideraba una "curiosidad de laboratorio" a lo que es actualmente: una industria a escala mundial, que ha destinado enormes recursos económicos, técnicos y humanos para continuar su desarrollo y ampliar el campo de aplicación de estos componentes. La incursión del C.I. en el área digital, ha dado como resultado la gran proliferación de las computadoras y el gran progreso de las mismas, en este campo es donde se introduce el término de Microelectrónica, que se define como la parte de la electrónica que trata de la realización de componentes, circuitos y sistemas extremadamente pequeños.

Es conveniente, recordar que los circuitos integrados, que se emplean en los diferentes equipos

electrónicos, pueden ser Lineales (Analógicos) ó Digitales. Los primeros operan básicamente mediante una señal que varia continuamente con el tiempo, llamada información analógica, en los sistemas digitales ó numéricos, la información consiste de combinaciones de tensiones ó corrientes, en donde su magnitud no es lo más importante para transmitir la información sino más bien el hecho de que esté ó no a la salida del sistema en este caso la información será numérica, es decir, número de metros, número de grados centigrados, etc.

Los grandes adelantos en las tecnologias de fabricación de circuitos integrados, han hecho que el desarrollo de estos, los lleve a incrementar cada vez más el número de componentes, incrementando su eficiencia y reduciendo notablemente su consumo de potencia y la relación de potencia dicipada en forma de calor.

El más grande de los logros alcanzado en el desarrollo de los circuitos integrados es el MICROPROCESADOR. Tradicionamente, las computadoras eran máquinas muy grandes y costosas, sin embargo, en la actualidad, la tecnología de los circuitos integrados hizó que fuera posible fabricar circuitos que contienen miles de transistores en un pequeño volumen de silicio. Los C.I. tuvieron efecto sobre la tecnología de las computadoras e hizó que el Microprocesador

se hiciera realidad. Puesto que el microprocesador es una CPU (Unidad Central de Proceso) en un integrado, los términos de CPU y Microprocesador se utilizan a menudo sinónimos.

El primer Microprocesador, presentado en el mercado, fue el 4004 por la compañía Intel en 1971, éste era un integrado muy sencillo que sólo podía manipular datos en un grupo de 4 bits, y fue a consecuencia de desarrollar calculadoras de un sólo integrado, la calculadora MCS-4 tenía como componente principal el microprocesador 4004, una memoria de programa 4001, una memoria de lectura y escritura 4002 y un registro de corrimiento 4003.

En 1972, Intel produjo el microprocesador 8008 de 8 bits, el cual substituía a el 4004, al mismo tiempo aparecieron el PPS-4 de la Rockwell International y el IMP-16 de la National Semiconductor. El principal uso que se les dió, fue como parte principal en algunas calculadoras electrónicas y en aplicaciones muy sencillas, estos circuitos integrados contituyeron lo que es la primera generación de microprocesadores.

Poco tiempo después, Intel diseño en substituto del 8008, con una mayor velocidad y potencia y fue el 8080, sus competidores de Motorola lanzaron al mercado el 6800, ambos de 8 bits, si bien en el diseño ambos eran bastante

diferentes, resultaban igualmente potentes y adecuados a su función de servir de base al diseño de una microcomputadora, esta fue la segunda generación de microprocesadores, abarcando el periodo de 1973 a 1975 y se caracteriza por la avanzada tecnología utilizada en la fabricación de estos circuitos integrados.

En el periodo de 1976 a 1977 un equipo de ingenieros, formado por Gary Kildall, John Torode, Frederic Faggin y Masatoshi Shima que anteriormente habían trabajado en Intel, aprovechando el detallado conocimiento que poseían acerca del 8080, pudieron ampliar su conjunto de instrucciones, diseñando así el microprocesador Z-80, el nombre deriva de Zilog Incorporated.

Paralelamente Intel actualiza su 8080 y diseña el 8085, compatible con el primero, con funciones adicionales y menos circuitos de soporte. En este periodo, sale al mercado una nueva empresa llamada MOS-Technology que con la ayuda de Chuck Peddle, crea el microprocesador 6502, el cual estaba basado en el 6800 de Motorola pues Peddle había trabajado para Motorola, aunque este 6502 sigue un criterio de diseño muy próximo al 6800, no son compatibles en nada pero el 6502 aventaja en mucho al 6800, siendo este 6502 el más utilizado en la primera década de la microcomputación.

Este periodo de 1976 a 1977, comprende la segunda y media generación de microprocesadores.

La tercera generación es a partir de 1978, donde el primer microprocesador representativo es el 6809, de ocho bits diseñado por Motorola, basado en su 6800, con una mayor sofisticación, probablemente él más capaz dentro de los microprocesadores de ocho bits, sin embargo apareció muy tarde para causar impacto y sólo ha sido utilizado en algunas máquinas.

Texas Instruments lanza al mercado el primer microprocesador de 16 bits, éste es el 9900, los 16 bits es la principal característica de esta tercer generación. Posteriormente Intel presenta el 8086 y el 8088, mientras Zilog presenta el Z8000 el cual pretende ser del todo compatible con el Z80 y por ende con el 8080, el más poderoso de los microprocesadores de este tipo es el 68000 de Motorola teniendo tambien la versión reducida que es el 68008.

La lucha por el dominio del mercado, entre los microprocesadores de 16 bits, es una repetición de los de 8 bits. Apesar de el desarrollo de los de 16, la mayoría de las microcomputadoras se basan actualmente en los de 8 bits, como el Z80 y el 6502.

Actualmente se trabaja en microprocesadores de la cuarta y quinta generación, pero aún no existe una división bien definida, algunas dispositivos representativos serían el microprocesador Hewlett-Packkard de 32 bits, el 80386 de Intel, el MC68020 de Motorola.

En los últimos años el microprocesador se ha transformado en una herramienta muy útil, ya que por su versatilidad permite utilizarlo en los campos más diversos, como son: en la computación principalmente, en la industria, y en aplicaciones especiales, como en la educación, en la medicina, en el área espacial, en la milicia y en el hogar.

La aplicación más importante de los microprocesdores es en la computación, siendo empleados como circuitos de soporte y programados para realizar funciones específicas dentro del sistema, disminuyendo el tamaño del sistema y aumentando su vida útil. Podríamos decir que hoy en día las computadoras forman parte importante en nuestra vida, su cultura es cada vez más nuestra cultura, están en las escuelas, en las fábricas y oficinas, desempeñan un importante papel en las comunicaciones, conducen y controlan equipos de transporte, realizan funciones de biblioteca, archivan, traducen, y un sinfin más de aplicaciones.

Nos hallamos, en una nueva era, en la cual las

computadoras, hasta hace poco elitistas, están a disposición de todo el mundo. En la actualidad se construyen computadoras personales ó microcomputadoras de bajo costo, con un microprocesador como unidad central de proceso. Según estadísticas, las ventas de microcomputadoras aumentan en un 50% cada año, y se prevee que para el año 2000 habrá al menos 300 millones de microcomputadoras.

Probablemente, dentro de unos años una microcomputadora casera valdrá lo que ahora un aparato de TV normal y estará por lo tanto ampliamente difundido, mediante este sistema doméstico podremos por lo tanto no sólo trabajar, estudiar, jugar, consultar datos y programar la vida diaria, sino que el sistema mismo de la información y de la enseñanza, a todos sus niveles, experimentará una auténtica revolución, debido a la difusión de las computadoras.

La invasión de microprocesadores y controladores programables en el hogar, se da casi sin darnos cuenta, en una máquina de lavar, en las nuevas tornamesas, en un horno de microondas, en los relojes digitales, los nuevos sistemas telefónicos con botones en vez de discos para marcar, y un banco de memoria para números, estos sin lugar a dudas son los ejemplos más significativos de la aplicación de los microprocesadores en el hogar.

En la industria, los microprocesadores automatizan y substituyen circuitos cableados muy complejos, disminuyendo los costos y aumentando la eficiencia de los sistemas en la industria, haciendola más "inteligente" y más eficaz, actualmente el diseño de una gran cantidad de maquinaria, se basa en el microprocesador como unidad central, en resumen la tecnología se esta automatizando por completo y en todas las ramas industriales, esta tendencia seguirá aumentado en una gran escala.

Las aplicaciones médicas son las más interesantes dentro de el área de utilización especial, existiendo dos clases en las que se utilizan los microprocesadores, y son los sistemas implantados y los sistemas para uso externo.

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

A nivel experimental, se ha desarrollado dispositivos, a base de microprocesador, para inducir una reacción de respuestas del sistema nervioso como consecuencia de una actividad anormal del cerebro.

Desde hace muchos años el hombre se ha dedicado al estudio de la rehabilitación, al perder alguna extremidad, ha usado materiales para substituirla, como son madera y los metales, con el advenimiento de los productos derivados de el petróleo, actualmente se emplean materiales más ligeros y resistentes como la fibra de vidrio, teniendo entonces prótesis más ligeras y de fácil manejo, dando como resultado una mayor comodidad. Con el progreso de la microelectrónica, los países desarrollados han mejorado considerablemente el diseño de las prótesis de extremidades superiores para amortiguar las molestias y dificultades de las prótesis mecánicas. Los problemas de espacio que se presentaba anteriormente ya no son tan graves debido al actual tamaño de los dispositivos electrónicos.

La prótesis electrónica funciona como un sistema neuronal simple; el paciente acciona un contacto localizado a un lado del muñon y el resto lo hace el sistema electrónico y mecánico que manda información al sistema electrónico, de manera automática, éste apagará ó accionará un motor según se

requierá, generandose así movimiento en la prótesis. Los movimientos individuales de los dedos son generados por micromotores y además las prótesis pueden ser tan sofisticadas que se puede incluir sensores de temperatura para ser colocados en la yema de los dedos aumentando así su funcionalidad.

La aplicación de la electrónica en la medicina es muy grande, y de manera importante es el uso de los microprocesadores, los cuales se pueden encontrar en un Marcapasos, un Audiómetro, un Intercambiador de Oxigeno, una Prótesis ó en una Laringe Electrónica.

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 monitoriadas 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.

La aplicación de los microprocesadores en el área militar es de suma importancia, pues en la actualidad se tiene que las fuerzas armadas son las "industrias" más

inteligentes del mundo, se desarrolla una gran tecnologia para ser aplicada en las más modernas armas. En la actualidad se posee la nueva generación de armamento controlado por microprocesadores, en donde la precisión y velocidad son factores más esenciales.

Los microprocesadores controlan factores de importancia primordial, como son la dirección del viento, la posición del objetivo, la clase de munición utilizado y el ángulo de tiro, a partir de esta información se calcula la ruta correcta y se apunta con precisión al objetivo.

Un ejemplo de este armamento serían los modernos misiles, en donde empleando diversos microprocesadores se controla el vuelo y la trayectoria del mismo, el microprocesador realiza de manera continua ligeros ajustes en la trayectoria y cuando el misil se está aproximando al objetivo se activa un sistema de guía final e identifica la imagen del objetivo a partir de una "imagen" compuesta por millones de números que tiene en una memoria, una vez que los detectores del misil han "visto" el objetivo, le envían al microprocesador otra imagen digital la cual es comparada con la de la memoria, dirigiendo el misil hacia la imagen del objetivo.

Es importante señalar, que debido al desarrollo de los

microprocesadores para la aplicación militar ha sido un factor primordial para acelerar el progreso de la electrónica y la computación. Sin las enormes sumas de dinero que se invierten en los laboratorios militares, es probable que no se diera el desarrollo tan fuerte en la tecnología a base de microprocesadores, siendo un factor decisivo en el desarrollo de la sociedad.

Por lo antes expuesto, se observa la gran importancia que tiene la Electrónica, la cual no ha hecho más que empezar. Puede decirse que en los próximos años los nuevos inventos electrónicos serán tantos y tan profundos que cambiarán radicalmente la forma de vida de las sociedades.

CAPITULO II

HISTORIA Y TEORIA BASICA

2.1 ORIGENES Y DESARROLLO DE LAS COMPUTADORAS

2.1.1 SISTEMAS DE NUMERACION PRIMITIVOS

Los pueblos primitivos aprendieron a contar con los dedos. por supuesto que no podían alcanzar cifras elevadas, pero sí las suficientes para satisfacer sus necesidades. Si querían recordar algunos números hacían incisiones en un palo ó marca sobre una piedra.

Cuando la gente aprendió a escribir, pronto halló también una manera de registrar números. Al principio se hacían solamente algunas marcas simples, como / para el uno, //// para el cuatro. Pero de esta forma no podían escribir cifras muy grandes. La escritura más antigua que se conoce es la de los egipcios que inventaron un símbolo para el diez, signo semejante a una U invertida. Hacían marcas hasta el nueve y luego escribían la U invertida volviendo a comenzar: /∩ =11, ////∩ =14, ////∩∩∩ =34, etc.

Eran signos sencillos, que cualquiera podía interpretar aún cuando no supiera ni leer ni escribir. No

todos los pueblos primitivos eran claros en el sistema de numeración que escogierón. Los babilonios tenian un sistema de numeración algo más complicado, acostumbraban contar de sesenta en sesenta. A éllos se debe el sistema actual que tenemos de contar los minutos y los segundos. Los babilonios escribian haciendo marcas en forma de cuñas sobre arcilla blanda.

Los griegos aprendieron de los egipcios el sistema de numeración con el número 1Ø como número básico, pero en vez de usar simples marcas para escribir los números, usaban letras de su alfabeto. La palabra griega que significa " diez" comenzaba con la letra "delta", la palabra griega que significa "cinco" comenzaba con la letra "pi", y estas letras pasaron a representar el "diez" y el "cinco" respectivamente y lo mismo se procedió para los demás números, empleándose combinaciones para representar cifras grandes.

Los romanos usaron signos simples combinados con algunas letras para construir su sistema de numeración, para el uno, dos y tres usaban rayas verticales; para el cinco usaban una V y para el diez usaban dos de estos signos, uno de ellos invertido en forma: χ, con el tiempo este signo se transformó en X, para representar cincuenta empleaban una L, para el cien C, para el quinientos una D y para el mil una M, estos pocos signos bastaban para sus necesidades.

Las letras numerales romanas eran mejores que las antiguas maneras de contar que se conocían y permanecieron en uso durante casi dos mil años. A fines de la Edad Media, la gente aún podía sumar y restar con los signos creados por la antigua Roma.

Sin embargo, mucho antes de esa época, se disponía ya de signos numerales mucho más apropiados, que son aquellos que se usan en la actualidad. Estos provienen de los árabes, pero los signos arábigos no reemplazarón a los romanos inmediatamente. En realidad pasaron varios siglos para que esto ocurriera, durante los cuales existieron uno junto al otro, pero mientras que un sistema perdía terreno lentamente, el otro lo ganaba.

La tremenda ventaja del invento árabe fué el de la convención de que cada cifra tiene dos valores, el absoluto y el relativo.

El valor absoluto de la cifra es el que tiene por sí misma, y el valor relativo es el que se adquiere por el lugar que ocupa en el número.

Por consiguiente un número es igual a la suma de los valores relativos de sus cifras.

Al sistema de números arábigos por lo tanto se le denominan sistema de posición y al sistema de números romanos se le denomina sistema sin posición.

Los números romanos han quedado remplazados, aunque aún perduran en la numeración de los capitulos de los libros y en unos cuantos lugares más.

2.1.2 EL ABACO

La computadora actual tiene como antecesor lejano el ábaco de arena, que consistía en un tablero rectangular con bordes a su alrededor, donde se colocaba una capa fina de arena y el operador podía realizar sus cuentas ayudandose con piedrecillas (calculi en latín) que acompañaban el tablero (abax = Tabla).

El ábaco chino (swan pan) aparece hacia el siglo VIII A.C. consiste en un bastidor, generalmente de madera, en las que van ensartadas las cuentas, con las cuales es posible realizar las cuatro operaciones fundamentales. Los Romanos usaban también el ábaco, aunque no de cuentas ensartadas, sino una tabla de hendiduras horizontales en las que se insertaban pequeños discos móviles, en donde la primera hendidura representaba las unidades, la segunda las decenas, etc. Los españoles al desembarcar en nuestras costas, se

encontraron un conjunto de varillas paralelas unidas a una pieza de madera en la que se podía ensartar cuentas, siendo este el ábaco Azteca.

Los ábacos se usan actualmente en muchas partes del mundo, pero destaca el ábaco japones moderno llamado Soroban, con el que pueden hacerse operaciones a una velocidad impresionante, en Japón se le da tanta importancia que existe un Instituto de Investigación de Abaco, en donde existen operadores tan hábiles que pueden obtener con él incluso raíces cuadradas.

2.1.3 LAS PRIMERAS MAQUINAS

El matemático y filósofo francés Blaise Pascal, a los 19 años de edad diseño la primera máquina aritmética ó calculadora mecánica del mundo, esto fué en 1642, con lo cual ayudo en una gran forma a su padre, pues este era recaudador de impuestos. La máquina funcionaba a la perfección, basada en ruedas dentadas y numeradas del 0 al 9, transportando los números de las unidades hasta la columna de las decenas mediante un mecanismo de trinquete, más ó menos de la misma forma que transporta los números el velocimetro de un automóvil, Blaise la denominó "PASCALINA", la idea original era una máquina de sumar, restar, multiplicar y dividir, pero sólo quedo como una sumadora, despertando el interés a nivel

científico y durante unos años se introdujeron muchas mejoras en la calculadora original.

El siguiente innovador fue Gottfried Wilhelm Leibniz, quien descubrió al mismo tiempo que Newton, el cálculo infinitecimal, y dado su interés en las matemáticas, trabajó en el perfeccionamiento de la máquina de Blaise pascal, intentando mejorarla de forma que fuera capaz de multiplicar y dividir por el método de sumas y restas repetidas, Leibniz perfeccionó el mecanismo de acarreo automático de Pascal con el llamado cilindro de Leibniz, utilizado en la actualidad en las calculadoras mecanicas.

Hasta este momento, en que Leibniz perfeccionaba su máquina, se trataba sin embargo de calculadoras no uilizables con fines prácticos, siendo estas máquinas sólo curiosidades hasta inicios del sigio XIX.

En 1802 el ingeniero francés, Joseph Marie Jacquard, construyó un telar automático, el cual estaba basado en tarjetas perroradas que corriendo sobre una cadena ó banda sinfin daban forma al diseño permitiendo entrar a las agujas en las partes perforadas.

La primera máquina con memoria y con posibilidades de efectuar operaciones multiples fue ideada por el matemático

Ingles Charles Babbage, en 1822, la cual denominó "MAQUINA DIFERENCIAL" desarrollada para calcular e imprimir tablas matemáticas. Trabajando en equipo con Ada Lovalece, Babbage se abocó rápidamente a un proyecto más ambicioso, el diseño de la "MAQUINA ANALITICA", con la cual esperaba alcanzar todos los objetivos para los cuales había diseñado la Máquina Diferencial, en muchos sentidos su nuevo diseño se parecía al microprocesador moderno, contenía un almacén de memoria y un sistema aritmético, proporcionaba una salida impresa e incluso era posible programarla, mediante el empleo de bifurcaciones condiccionadas. Las instrucciones se controlaban mediante clavos largos, como en un organillo, posteriormente se adoptó el sistema de tarjetas perforadas que Joseph Jacquard había introducido en la industria textil.

Desafortunadamente, aunque Babbege fue el primero en desarrollar estos conceptos y tratar de ponerlos en práctica, no llegó a terminar ninguna de sus dos máquinas, debido a las limitaciones técnicas de la ingeniería del siglo XIX.

La primera máquina de calcular que alcanzó éxito comercial fue la del financiero francés T. de Colmar quien la inventó en 1820 y la fabricó para su venta a partir de 1831, esta máquina utilizaba los conceptos de Pascal y de Leibniz, pero resueltos con una máquina de superior calidad.

En los Estado Unidos, W.S. Burroughs diseña en 1892 una máquina sumadora de palanca que alcanza una gran popularidad por lo práctico y confiable en su funcionamiento. La fabricación en serie y la variedad de modelos, hacen de esta firma una de las más importantes, en ella se pasa gradualmente de la máquina de palanca manual a la eléctrica.

El procesamiento de datos propiamente dicho comienza con las máquinas clasificadoras y tabuladoras del ingeniero norteamericano Herman Hollerith, quien aplicando la tarjeta perforada de Jacquard, los relevadores eléctricos programables desde un tablero de clavijas y sistemas electromecánicos para la exhibición de datos, construye las primeras computadoras electromecánicas. Su aplicación inicial fue para la obtención de los resultados del censo norteamericano de 1890, en el que se clasificaron la edad, el sexo, dirección, nombres, etc, de más de sesenta millones de personas en un tiempo record.

Esta lectora de tarjetas, tenia la desventaja de que era necesario perforar las tarjetas agujero por agujero, pero en el año de 1916 él mismo inventó un dispositivo que solucionaba el problema de la perforación de tarjetas. Con las modificaciones necesarias a la computadora de Hollerith, muchas máquinas irrumpen en la industria y en las finanzas a principios del siglo XX. así, las grandes empresas como las

compañias telefónicas, los ferrocarriles, etc, consiguen acumular gran cantidad de información y manipular a su conveniencia con rapidez y exactitud. En México, se instala la primera de estas máquinas para los ferrocarriles Nacionales en el año de 1927. Finalmente Hollerith es el fundador de la IBM (International Business Machines) en 1924.

Vannevar Bush, fisico norteamericano, en 1931 inventa el "ANALIZADOR DIFERENCIAL", el cual era una calculadora electromecánica que resolvia ecuaciones diferenciales, la máquina constaba de una compleja estructura de engranes, ejes y motores eléctricos, los trabajos de Bush se inspiran en el método sugerido por primera vez por lord Kelvin, el cual hacia referencia a la alimentación de la salida de un "integrador" conectada a la entrada de otro, sin embargo la potencia de salida era demasiado débil y no podia ser utilizada como entrada, este método no pudo ser aplicado hasta que fueron inventados los amplificadores.

En la década de los cuarentas un ingeniero alemán Konrad Zuse, trabajó en una calculadora programable, que se considera como el primer computador de la historia, su primera máquina, el Z1, era un dispositivo mecánico que podia efectuar las cuatro operaciones matemáticas elementales, calcular raices cuadradas y convertir números decimales a

notación binaria y viceversa. En el año de 1941 construyó un computador electromagnético, al cual llamo Z2, el gobierno de Alemania proporcionó fondos a Zuse para desarrollar el Z3, éste habria de ser un computador eléctrico, con un tendido de cables extenso que posibilitaron un diseño más compacto y elegante, en lugar de los enlaces mecánicos que utilizaron las máquinas anteriores.

El Z3 podia almacenar 64 palabras de 22 bits de longitud, a la información se le daba entrada a través de un teclado y los resultados se exibían visualmente en un conjunto ordenado de lámparas montadas sobre un tablero. El último computador que produjo Zuse durante la guerra, fuè el Z4, el cual había incrementado la longitud de palabra a 32 bits, todos estos computadores fueron destruidos en 1945 en el bombardeo a Berlin.

En 1943 el matemático británico Alan Turing, desarrolla su procesador llamado "COLOSSUS", el cual era una enorme máquina que empleaba 1500 bulbos, procesaba 5000 caracteres pur segundo y tenía la misión de decifrar los mensajes alemanes en la segunda guerra mundial, se dice que el Colossus es el primer procesador eléctrico del mundo.

A lo largo de esta evolución los dispositivos seguían siendo de naturaleza mecànica, por lo tanto

extremadamente lentos y engorrosos, el siguiente paso fué la intoducción de mecanismos electromagnéticos movidos mediante dispositivos eléctricos, esta inovasión llevó a la construcción, en 1944, del "MARK I", una máquina que para su época tenia una potencia enorme: 10 operaciones por segundo, sus dimensiones eran notables, 18 metros de longitud por 2.5 de altura. Esta máquina fue diseñada por H. Aiken, de la universidad de Harvard, en los Estados Unidos, en colaboracion con la IBM, siendo esta la última gran máquina electromecánica que se ha construido.

En 1946 se construye la primera máquina totalmente electrónica, la cual fue denominada ENIAC (Electronic Integrator and Calculador), desarrollada en la universidad de Pennsylvania po J.P. Eckert. El ENIAC erá más de 1000 veces más rápido que las computadoras electromecánicas, usaba un total de 18 mil bulbos y estaba constituída por 30 unidades, requería un espacio de aproximadamente 9x30 m. y pesaba 30 toneladas.

Los primeros problemas fueron la escasa memoria y la falta de fiabilidad. El ENIAC sólo podia almacenar 20 numeros de 10 digitos y todo el programa tenia que hacerse reordenando las conexiones, cabe decir que hasta entonces, no se habia introducido el concepto de programa, el operador debia teclear en cada momento el calculo a efectuar.

Debido a que la máquina sólo podía funcionar unos dos minutos antes de que los bulbos empezaran a fundirse, la vida del ENIAC fue corta, siendo retirado de funcionamiento en 1952.

En 1950 aparece la noción de "programa", por el matemático John Von Neumann, quien dió a conocer su trabajo referente al concepto de "programa almacenado", en donde las instrucciones y los datos usados durante el procesamiento podía ser almacenados dentro de la computadora, y siempre que fuera necesario, la computadora tendría la capacidad de modificar estas instrucciones de programa durante la ejecución del mismo. Trabajando en la universidad de Princeton, Von Neuman diseño, basado en sus investigaciones sobre el "programa almacenado", una computadora la cual recibió el nombre de JOHNIAC.

En 1951 la UNIAC I (Universal Automatic Computer), es presentada como la primera computadora comercial, construida a base de bulbos y ocupaba un gran volúmen. El calor generado por esta computadora debía ser controlado mediante sistemas de aire acondicionado.

Los desarrollos en el campo de las computadoras se multiplicaron rápidamente a partir de 1950, desarrollandose principalmente los tipos de almacenamiento interno de datos

los cuales eran proporcionados mediante tiras y tarjetas de papel. Las computadoras construidas hasta este tiempo constituyeron la primera generación de computadoras.

2.1.4 LAS COMPUTADORAS ACTUALES.

Como en toda nueva tecnología industrial, los modelos iniciales de computadoras fueron resultado de un extraordinario cúmulo de investigaciones y experimentos; fueron modelos hechos a mano, y frecuentemente impresionaban mas que nada por su tamaño físico.

Dado los precios tan elevados de estos equipos, había sólo unas cuantas instituciones, principalmente militares, de investigación y gubernamentales, que podían afrontar estas nuevas maravillas de la tecnología. El proceso entero se complicaba aún más por la necesidad de preparar programas que pudieran dar resultados esperados. El lenguaje de programación por aquella época era simple y puro lenguaje de máquina.

En su corta vida, la industria de la computación ha tenido que pasar por cuatro generaciones. La primera generación formal de las computadoras, aparece apróximadamente en 1950, utilizandose sólo como instrumentos

de calculo, su principal característica es el uso de bulbos. Se utilizan sistemas magnéticos de memoria central, lectoras y perforadoras de tarjetas y cintas de papel, se utilizan lenguajes de máquina y ensambladores primitivos, su alfabeto es basicamente numérico, manejaban hasta 10000 operaciones por segundo, la memoria central almacena hasta 8000 palabras. Como modelos tipicos de esta primera generación tenemos: IBM-650, Bendix-G15, UNIVAC SS90, Bull-pT, IBM-790.

La invención del transistor, en 1948 en los laboratorios Bell, por los doctores John Bardeeen, William Shockley y Walter Brittain, anunció una nueva generación de computadoras. Teoricamente, el funcionamiento del transistor es similiar al del bulbo, pero su rendimiento es superior, es más pequeño y más barato de fabricar, tienen una mayor velocidad, con mayor fiablidad y sin generar calor. El empleo del transistor que substituye al bulbo en 1957, origina la segunda generación de computadoras con una reduccion apróximada de voldmen de 100 a 1, y un aumento tal en la relación capacidad/precio, que permite su introducción en la industria y en el comercio. En 1957 la compañía CDC (Control Data Corporation) produce la computadora CDC-1604 totalmente transistorizada.

Las computadoras de la segunda generación entran al mercado formalmente alrededor de el año 1960, sus

características son: se utilizan como instrumentos de cálculo y procesamiento de datos, sistemas de memoria magnéticos y memorias virtuales de ferritas, impresoras y cintas magnéticas, utilizan los primeros lenguajes de compilación (FORTRAN y ALGOL), su alfabeto es a base de números y letras, su capacidad de memoria es de 32000 palabras, procesan hasta 100000 operaciones por segundo. Los modelos representativos de esta generación son: CDC-160, IBM-1401, IBM-7090, RCA-305, Burroughs 5500, Bendix G20, CDC-3600, CEC-6600, CDA-160A, Larc Stretch, Philco 2000, RCA-301. Una de las primeras computadoras de este tipo, una IBM-7070, se instaló en México en 1960 al servicio del Instituto Mexicano del Seguro Social.

El siguiente adelanto es el circuito integrado, el cual en un principio consistía en la fabricación simultánea de ocho ó diez transistores en el mismo sustrato, así nace el microcircuito y se logra una reducción en el volúmen, iniciandose la tercera generación de computadoras, que se caracteriza por la integración a gran escala de circuitos integrados hibridos y monoliticos microminiaturizados. Esta generación aparece apróximadamente de 1964 a 1970, utilizandose en sistemas de información, su principal característica es el uso de circuitos integrados (LSI) y memorias de película magnética, se introducen las terminales de video y teletipos.

Las computadoras de la tercera generación dieron lugar al desarrollo de una gran variedad de lenguajes de alto nivel como el FORTRAN, COBOL, PL y BASIC, lo cual contribuyó a su adaptación en los más diversos campos, tales como la industria. Aumenta la velocidad de procesamiento a 1000000 de operaciones por segundo y un aumento de la memoria de 64 a 256 K de palabras, se introducen los primeros sistemas de cintas magnéticas como unidades de memoria. Los modelos representativos de esta generación son: IBM-360 y 370, Burroughs 6700, CDC-6000 y 7000, PDT-10, PDP-20, UNIVAC 1106, CYBER-170.

Con el desarrollo de la microelectrónica surge el microprocesador, desarrollandose así la cuarta generación de computadoras caracterizandose por el diseño de sistemas pequeños ó minicomputadoras, utilizandose en sistemas de comunicación, en sistemas de información para negocios pequeños y personales. Esta generación basada en la microelectrónica (VLSI), utiliza memorias de tipo MOS, terminales inteligentes, discos y cintas magnéticas, equipos de graficación y lectores ópticos. En esta generación de computadoras se presenta la comunicación entre equipos, la velocidad de procesamiento aumenta considerablemente a un rango de 10000000 de operaciones por segundo, aumentando tambien la capacidad de memoria e innovando diferentes sistemas de almacenamiento de información.

Dentro de los modelos de computadoras de esta generación se tienen a la IBM-4330, UNIVAC 1100, Burroughs B6900 y B7900, Prime 550, MP 3100VAX y dentro de las minicomputadoras aparece Apple II, TR80, IBM-PC, Altair.

En la actual etapa de la cuarta generación, el proceso de acorcamiento continuó, se convierte en un excelente apoyo para experimentacion y la enseñanza.

Por lo tanto es notorio que en esta cuarta generación, la reducción de costos, la confiabilidad y reducción de requerimientos para su operación, son las principales aportaciones en el desarrollo de las computadoras.

Actualmente, empiezan a definirse en los centros de investigación estadounidenses y europeos, los conceptos que darán lugar a la quinta generación de computadoras. Paralelamente, la industria japonesa desarrolla un proyecto de gran magnitud para poner en operación una nueva generación de computadoras, sustancialmente distintas a las existentes, en los primeros años de la década de los noventas.

Estas nuevas máquinas habrán de caracterizarse por la utilización de enjambres de procesadores microscópicos, operando simultáneamente para recibir y clasificar

información. El sistema de control habrá de seguir y utilizar principios ya conocidos hoy en día.

Finalmente, una de las principales características que tendrán las computadoras de la quinta generación será una programación casi total por las mismas computadoras.

2.2 TEORIA BASICA.

2.2.1 SEMICONDUCTORES

Existen tres grupos de materiales según su conductividad eléctrica: metales, semiconductores y aislantes. Los metales se caracterizan por tener una conductividad muy alta, mientras que los aislantes presentan una conductividad muy baja casi nula, los semiconductores bajo ciertas condiciones se pueden comportar como aislantes y presentar una baja conductividad ó bien aumentarla gradualmente hasta tener el comportamiento de un metal.

El mecanismo de conducción que se da en una red cristalina, (un cristal consiste en un arreglo tridimensional de átomos y moléculas) lo llevan a cabo los electrones de Valencia, siendo estos el número de electrones que giran en la órbita exterior y son estos electrones los que determinan las características físicas y químicas de los materiales.

Para que los electrones de Valencia puedan conducir una corriente dentro del material, es necesario proporcionarles una energia adicional que los libere del átomo al que originalmente pertenecia. Estos electrones libres se convierten en electrones "libres" ó de conducción.

La distancia de los electrones de Valencia, con respecto al núcleo, determina el nivel de energia en que se encuentra cada electrón, este concepto de nivel de energia es la relación entre la energia total del electrón, cinética y potencial, al radio de su órbita. Estos niveles de energia se dan en un átomo aislado pero en un grupo de àtomos ó en una red cristalina, en donde la unión de los electrones crea las bandas de energía, bandas que son formadas por niveles de energía estrechamente unidos entre si.

La fig. 2.1 muestra los diferentes niveles de energia de los electrones.

En donde:

EV: es la máxima energia que pueden tener los electrones de Valencia.

EC: es la energia mínima que tienen los electrones de conducción.

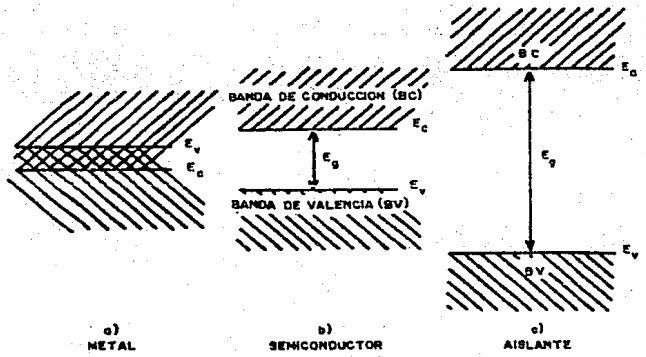


FIGURA 2.1 NIVELES DE ENERGIA

Generalmente por la acción de la temperatura (ó de cualquier otro estímulo externo, como la luz) se dice que los electrones pasan de la banda BV a la banda BC. En el caso de los metales, estas dos bandas se traslapan y se tienen electrones de conducción prácticamente a cualquier temperatura. En los semiconducores y los aislantes la banda de conducción esta separada de la de Valencia, a la región intermedia se le denomina banda de energía prohibida, y su altura, característica de cada material, se designa como Eg. En el caso de los aislantes, esta Eg es tan grande que se necesita demasiada energia para tener electrones de conducción.

El ancho de banda prohibida (Eg) del germanio y del silicio es, respectivamente, de 0.67 eV y 1.1 eV (electrón-volt, se define como unidad de energía que "gana" un electrón cuando es acelerado a través de un potencial V=1.0 volt, se conoce como 1.0 e.V.).

El silicio tiene 4 electrones en su órbita de Valencia ligados levemente al núcleo. Si se ponen en contacto un conjunto de átomos de Si. comparten entre si sus cuatro electrones exteriores; cada átomo comparte 2 electrones con 4 de sus átomos vecinos en enlaces covalentes ó par de electrones, enlaces que mantienen a los átomos fijos en el espacio formando una red, ó estructura regular llamada cristal como se ve en la fig. 2.2.

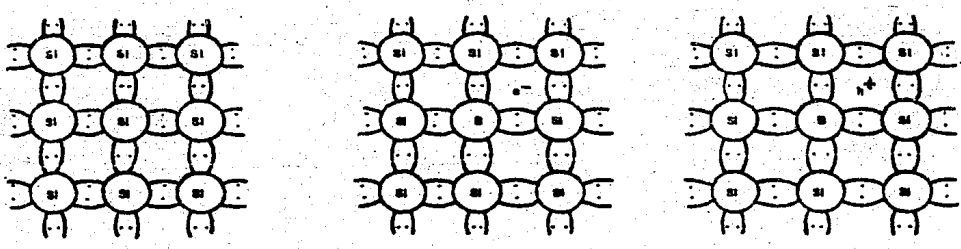


FIGURA 2.2 RED CRISTALINA

Cuando en la red cristalina no intervienen átomos de impureza, se tiene un semiconductor intrínseco, cuando por alguna razón, un electrón e(-) abandona su enlace, se hace portador de carga negativa; este electrón libre origina o deja en su lugar una imperfección ó hueco (+) con exceso de carga positiva por lo que se ha roto el equilibrio eléctrico del átomo original. Ahora es fácil para otro electrón e(-), en un enlace covalente próximo abandonar su posición para trasladarse al hueco h(+) dejado por el electrón libre y dejando a su vez otro hueco (+) en su lugar, asimismo otro electrón ocupará el hueco (+) y así sucesivamente. De este modo el hueco se va "moviendo" a través del cristal, llevando una carga positiva de igual valor a la del electrón libre pero de signo contrario.

La conducción puede entonces llevarse a cabo por electrones e(-) en la banda de conducción, ó por huecos h(+) en la banda de Valencia.

En un semiconductor intrínseco la cantidad de e(-) en la BC y de h(+) en la BV es la misma, en donde a temperatura ambiente, el semiconductor intrínseco es mal conductor pudiendo aumentar la conducción eléctrica del mismo, incrementando la temperatura.

La corriente en los semiconductores es debida al

movimiento de electrones y huecos, por lo tanto existen dos tipos de portadores: el electrón e(-) y el hueco h(+).

Si se introducen impurezas (átomos de elementos pentavalentes ó trivalentes) para formar ya sea, cristales de silicio con exceso de cargas libres, el material será tipo N, cuyos portadores mayoritarios son casi exclusivamente electrones, ó para formar cristales de silicio con exceso de huecos ó cargas positivas libres, el material será tipo P, cuyos portadores mayoritarios son casi exclusivamente huecos.

Los tipos de materiales, intrinseco, N y P se muestran en la fig. 2.2 en donde se tiene como elemento principal el silicio.

De esta manera se puede tener un material de baja conductividad a partir de un semiconductor intrinseco, o bien aumentar ésta, según se desee, introduciendo al material la cantidad adecuada de impurezas, teniendo además la posibilidad de obtener dos tipos de conductividad diferentes: tipo P y tipo N. La fig. 2.3 muestra la unión P-N.

Cuando un semiconductor tipo P está en estrecho contacto con uno de tipo N, se tiene lo que se denomina la unión P-N. En la mayoria de los dispositivos electrónicos con

semiconductores, intervienen una o varias uniones P-N,un semiconductor tipo P realiza la conducción por huecos, que son los portadores mayoritarios, mientras que uno tipo N, lo hace por electrones. Al estar estos dos materiales en contacto íntimo, algunos electrones de la región N se difundirán hacia la parte P. Igualmente lo harán los huecos de la región P hacia la región N, debido a estos desplazamientos de carga, aparecerá en ambas regiones de la unión una zona de átomos ionizados, de alta resistividad.

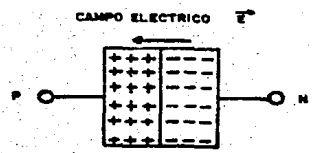


FIGURA 2.3 UNION P-N

Dado que estas zonas, llamadas de carga especial, tienen una carga eléctrica opuesta, se genera un campo eléctrico $\vec{E}$ entre ellas, estableciendo así una barrera de potencial que limitará el número de huecos y electrones que se difunden de uno a otro lado de la unión. La intensidad de este campo eléctrico, depende de la cantidad de impurezas contenidas en las regiones tipo P y N. En la fig. 2.4 se observan los tipos de polarización en la unión P-N, en el sentido directo se inyectan electrones a la región N y huecos

en la región P, reduciendo así el campo eléctrico en la unión. Esto facilitará el flujo de electrones de la región N a la región P y de huecos de la región P a la región N, estableciendose así una corriente electrica a través de la unión y ésta dependerá del voltaje aplicado, dicho de otra manera, la unión P-N polarizada directamente presenta una baja resistencia (impedancia).

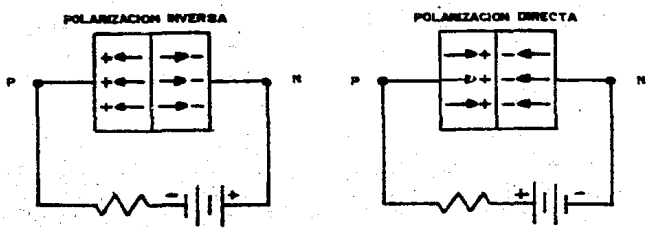


FIGURA 2.4 POLARIZACION

En el caso contrario, cuando polarizamos en sentido inverso, fluyen electrones de la región N a la región P y huecos de la región P a la región N, al hacer esto, la región de carga especial se hará más ancha, provocando que el campo eléctrico entre ellas aumente considerablemente, aumentando así la barrera de potencial, no hay por lo tanto conducción de electrones de la región N a la P, como de huecos de la región P a la N, es decir que la unión P-N presenta una alta impedancia.

El dispositivo semiconductor, más simple que existe, es el diodo, constituido basicamente por una union P-N, con sus respectivos contactos, el simbolo eléctrico está representado en la fig. 2.5 en donde la región P se llama ánodo y la región N se llama cátodo.

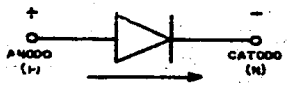


FIGURA 2.5 EL DIODO

El diodo tiene la característica de rectificar, en la fig. 2.6 se muestra la relación de corriente-voltaje de un diodo, en polarización directa la corriente crece en forma exponencial, mientras que en polarización inversa la corriente es nula o muy pequeña (corriente de fuga), si el voltaje inverso sigue aumentando, en un determinado voltaje se produce la ruptura de la unión. Es importante que el diodo tenga una corriente de fuga lo mas pequeña posible y esto dependerá directamente de el proceso de fabricación.

Los diodos pueden fabricarse de germanio o de silicio, aunque en la actualidad son de silicio. En la fig. 2.7 se comparan algunas propiedades de los diodos de germanio y de silicio.

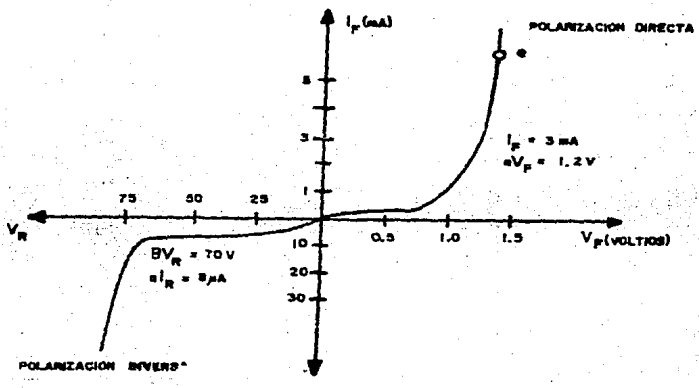


FIGURA 2.6 RELACION I-V

El comportamiento de un diodo, se describe en función de la relación entre el voltaje y la corriente existente en sus terminales. Es importante hacer notar que las características de un diodo se especifican a una temperatura fija, generalmente a la temperatura ambiente, ya que al cambiar la temperatura, también cambiarán las características del diodo.

Otro dispositivo muy importante es el transistor, el cual se forma añadiendo una tercera región impurificada a la estructura del diodo, según como se ordene estas regiones se obtendrán transistores de tipo NPN ó PNP. La fig. 2.8 presenta un transistor PNP,constituido por dos capas de

material P, separadas por una capa N, lo cual da lugar a dos uniones P-N.

	Germanio	Silicio
Voltajes de ruptura	Hasta 700 volts	Hasta 2 000 volts (útiles como rectificadores de alto voltaje)
Voltaje de encendido	Desde 50 mV (muy eficientes como rectificadores de muy bajo voltaje)	700 mV
Temperatura que puede soportar la unión	90 °C	Hasta 200 °C (muy útil en rectificadores de potencia)
Corriente de fuga	Pequeña	Muy pequeña
Comportamiento en altas frecuencias	Muy bueno	Bueno

FIGURA 2.7 DIODOS GERMANIO SILICIO

A la región P, de la izquierda se le denomina Emisor, a la región N, se le llama Base y a la región P, de la derecha se le denomina Colector.

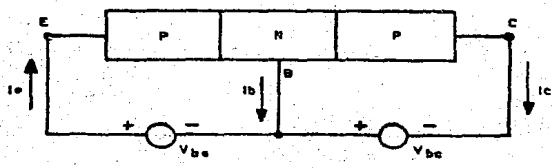


FIGURA 2.8 TRANSISTOR PNP

El otro tipo de transistor, el NPN, funciona por medio de una corriente de electrones a diferencia del transistor PNP, que lo hace con una corriente de huecos.

Los transistores NPN y PNP, constituyen la clase de dispositivos llamados transistores de unión, son también conocidos como transistores bipolares de juntura (TBJ), debido a que su operación se debe a portadores de carga de ambas polaridades. Los símbolos del transistor PNP y NPN, se muestran en la fig. 2.9.

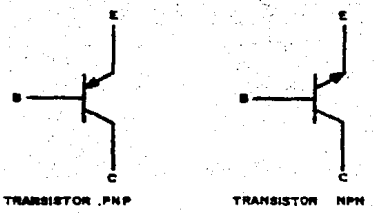


FIGURA 2.9 TRANSISTORES PNP Y NPN

La flecha del emisor muestra el sentido del diodo base-emisor, e indica también el sentido de la corriente a través del emisor.

El transistor en ecencia es un amplificador de corriente, la corriente de entrada es Ib, la corriente de salida es Ic y estará dada por: $Ic = \beta\ Ib$

donde: $\beta > 1$.

REGION DE AMPLIFICACION: Como se ve en la fig. 2.1Ø el diodo BE está polarizado en sentido directo (conducción) y el diodo BC esta polarizado en sentido inverso (no conducción) donde Vc>Vb>ve. En esta región $\beta = Ic/Ib$ toma valores desde 2Ø hasta 999.

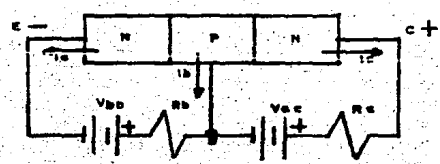


FIGURA 2.1Ø TRANSISTOR EN AMPLIFICACION

REGION DE CORTE: Como se ve en la fig. 2.11 los diodos BE y BC estan en inversa y por lo tanto ninguno de los dos conduce, por lo que no hay circulación de coriente en el TBJ y dice que el transistor está cortado.

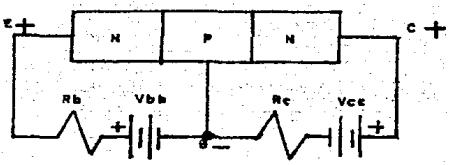


FIGURA 2.11 TRANSISTOR EN CORTE

REGION DE SATURACION: En la fig. 2.12 los dos diodos estan polarizados en sentido directo por lo que conduce y hay un exceso de corriente en el TBJ por lo que Ic depende de las resistencias Rbe y por Rbc por lo que $Ic \neq \beta Ib$.

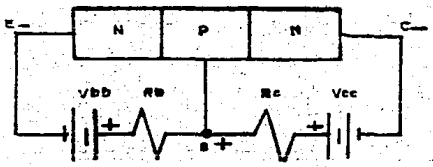


FIGURA 2.12 TRANSISTOR EN SATURACION

La fig. 2.13 muestra la gráfica de las regiones características del transistor bipolar de juntura (TBJ).

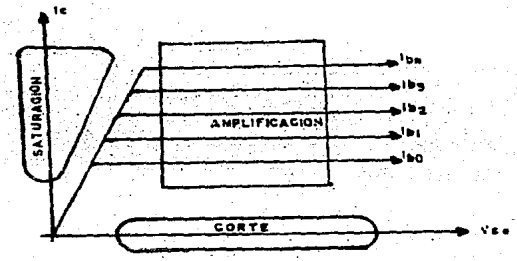


FIGURA 2.13 REGIONES DEL TRANSISTOR

2.2.2 ELECTRONICA DIGITAL.

La automatización de la tecnología se ha logrado con la ayuda de pequeños circuitos electrónicos, los cuales pueden ser analógicos, digitales e híbridos. Los sistemas digitales estan constituidos por un sin número de elementos, entre los cuales hay compuertas lógicas, memorias, contadores, multiplexores, etc., los cuales tienen un número finito de estados, dos en este caso, los cuales se conocen como estados lógicos y se les denomina 1 y Ø ó alto y bajo ó encendido y apagado. La principal razón de que solo tengan 2 estados es la confiabilidad que se tiene al manejarlos.

La característica fundamental de una computadora es la habilidad de representar, almacenar y procesar información, la información se puede representar en

diferentes sistemas numericos, el sistema binario ha probado ser el más eficiente para utilizarlo en las computadoras, pues la informacion se presenta mediante "unos" y "ceros", los cuales pueden ser procesados fácilmente mediante dispositivos digitales.

SISTEMAS NUMERICOS Existen distintos sistemas de numeración, de los cuales el más comunmente usado es el sistema posicional, en este sistema cada número tiene un valor, que depende de su posición.

En el sistema posicional, un número N se representa por la siguiente ecuación:

$$N=d_{n-1}\,xb^{n-1}+\ldots+d_{n-0}xb^{0}+d_{-1}\,xb^{-1}+\ldots+d_{-m}\,xb^{-m}$$

donde los coeficientes: d.- representan los digitos.
b.- la base
n.- el número de digitos enteros.
m.- el número de digitos fraccionales.

Para identificar la base de un número, se pone una letra mayúscula en su extremo derecho: una N para los binarios, una L para los Octales, una H para los Hexadecimales y nada para los Decimales.

En el sistema de numeración cotidiano, el sistema

Decimal empleamos diez dígitos Ø,1,2,...,9. Un número mayor que 9 se representa asignando un significado al lugar o posición ocupado por cualquier digito. Por ejemplo, en virtud de las posiciones ocupadas por los dígitos individuales del número 69Ø3, el significado numèrico se calcula:

$$69Ø3 = 6x1Ø^{3} + 9x1Ø^{2} + Øx1Ø^{1} + 3x1Ø^{0}$$

Notamos como en la ecuación anterior, un número se expresa como la suma de potencias de 1Ø multiplicadas por coeficientes apropiados. En el sistema decimal, el 1Ø es la base o raíz del sistema, hay diez dígitos, de los que el mayor es 9 en un sistema de numeración de base n hay n dígitos y el mayor es n-1.

En relación con los sistemas digitales resulta muy conveniente utilizar el sistema de numeración de base 2, este sistema se denomina Binario y utiliza solamente los dígitos "Ø" y "1" una ventaja de utilizar el sistema binario es que podemos establecer una correspondencia uno-a-uno entre los dos dígitos (numéricos) Ø y 1 y los valores lógicos (no numéricos) "Ø" y "1", encendido y apagado, alto y bajo.

Cuando un número se reprenta en el sistema binario, los dígitos individuales reprentan los coeficientes de las potencias de 2 en lugar de las potenciasde 1Ø como en el decimal. Por ejemplo, el número decimal 19 se escribe en la

representación binaria como 10011, ya que el grupo de digitos binarios tiene el significado:

$$10011 = 1x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 1x2^0$$

$$= 16 + 0 + 0 + 2 + 1$$

$$= 19$$

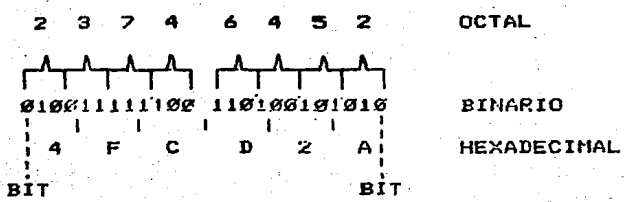
Los sistemas digitales manejan facilmente los números binarios pero nosotros estamos acostumbrados al sistema decimal. Para hacer un manejo más comodo de los bits, los agrupamos de tres en tres o de cuatro en cuatro, para representar estos agrupamientos utilizamos el sistema Octal y Hexadecimal respectivamente.

Los sistemas de numeración Octal y Hexadecimal resultan interesantes, ya que tienen una relación especial con el sistema binario. En el sistema Octal la base es ocho, y los digitos empleados son del 0 al 7. En el sistema Hexadecimal la base es dieciséis, y los diez digitos decimales habituales proporcionan 10 de los digitos requeridos y los otros seis se representan por letras A, B, C, D, E y F.

Las relaciones especiales, de los sistemas Octal y Hexadecimal con el binario, surgen de que tres digitos binarios pueden representar exactamente ocho (2^3) números diferentes y cuatro números binarios pueden representar

exactamente dieciséis (2^4) números distintos.

A continuación se muestra como se lleva a cabo la conversión entre sistemas de numeración Octal binario y Hexadecimal.



más significativo menos significativo

BIT. Se define como un bit a la unidad mínima de información que puede adoptar dos valores o estados "1" ó "0" lógicos.

En el Apéndice 'F' se muestran las diferentes conversiones existentes entre los diferentes sistemas numéricos antes mencionados.

El sistema numérico binario, tiene la desventaja de que las conversiones entre los códigos binarios y decimal son relativamente complicadas, ya que en general, cada dígito binario puede afectar a cada dígito decimal y viceversa. Cuando es importante poner remedio a esta situación, puede utilizarse el sistema de representación decimal codificado en

binario BCD (del ingles Binary Coded Decimal)

DECIMAL CODIFICADO EN BINARIO En el sistema BCD, se emplean cuatro dígitos A,B,C y D para representar los dígitos decimales del Ø al 9, cuando se tiene un número decimal multidígito hay una correspondencia uno a uno entre los dígitos individuales decimales y el grupo binario BCD. Así por ejemplo el número 96Ø3 en BCD aparece como:

9 6 Ø 3

1ØØ1 Ø11Ø ØØØØ ØØ11

Por lo tanto la comversión de decimal a binario necesita examinar solamente un dígito decimal cada vez, en la conversión reciproca es necesario examinar cada vez cuatro dígitos binarios. Una desventaja del código BCD es que de las dieciséis combinaciones posibles con cuatro dígitos sólo se utilizan diez, como consecuencia, un número BCD tiene más dígitos que su número equivalente expresado en binario puro.

Las cadenas de bits, es decir, palabras pueden representar no sólo números sino tambien datos e información, todo lo que se requiere es que haya un entendimiento, es decir, un mismo código conocido por el generador, transmisor ó fuente y por el receptor de datos. Un código ampliamente usado en la transmición de datos ya sean números, letras del alfabeto, caracteres de puntuación e instrucciones, es el

código ASCII (American Standar Code for Information Interchange), el código utiliza 7 bits, asi que 2^7 =128 elementos de información diferentes pueden ser transmitidos. Este código se explica ampliamente en el capitulo III.

El concepto de variable lógica, fue introducido en 1850 por el matemático George Boole, en relación con sus estudios de los procesos del pensamiento, la adaptación del álgebra Booleana a los sistemas digitales, fue presentada en 1938 por Claude Shannon de los laboratorios Bell.

El álgebra de Boole, ha logrado una verdadera importancia en la era de las computadoras. Aunque la mayoria de las computadoras pueden efectuar la gama completa de operaciones aritméticas, sólo la suma se efectúa directamente, mediante circuitos lógicos, las operaciones restantes como la multiplicación, la división y la resta, se realizan combinando adecuadamente los circuitos "sumadores".

La teoria de Boole se basa en dos condiciones ó en los llamados símbolos de Boole: AND (Y) y OR (O).

COMPUERTAS LOGICAS La compuerta lógica es el elemento básico en los sistemas digitales, estos se construyen generalmente usando sólo tres compuertas lógicas basicas, las cuales son: la compuerta AND, la compuerta OR y la compuerta NOT. El

principal factor en la determinación de la rapidez, con que un sistema digital realiza la función a la que se ha diseñado, es la velocidad de operación de sus compuertas, los tiempos que emplean las compuertas varían entre los microsegundos y los nanosegundos.

Las compuertas lógicas se utilizan en los sistemas digitales para decodificar instrucciones, señales de control y direcciones de ciertos procesos internos, asi como de ejecutar operaciones aritméticas dentro de las unidades aritméticas.

COMPUERTA NOT Esta compuerta tambien se le denomina Inversor, pues genera el complemento de lo que se pone a la entrada, la fig. 2.14 muestra esta compuerta.

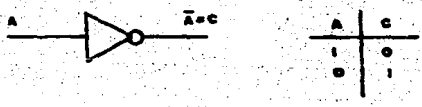


FIGURA 2.14 COMPUERTA NOT

COMPUERTA AND En este tipo de compuerta, sólo se presentará un "1" en la salida, cuando todas las entradas presenten un "1" respectivamente. La compuerta AND (Y) es representada por la expresión booleana siguiente: A•B=C. La expresión anterior se lee como A "Y" B igual a la salida C,

el punto (•) representa la función lógica AND en el Álgebra Booleana, la fig. 2.15 muestra esta compuerta.

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

FIGURA 2.15 COMPUERTA AND

Las leyes del álgebra Booleana que gobiernan las operaciones de la compuerta AND son las siguientes:

$A \cdot \emptyset = \emptyset$

$A \cdot 1 = A$

$A \cdot A = A$

$A \cdot \overline{A} = \emptyset$

COMPUERTA OR La salida en una compuerta OR (O) será un "1", cada vez que aparece un "1" en alguna o todas las entradas, la expresión de una compuerta OR es: A+B=C, se lee como A "O" B igual a la salida C, la fig. 2.16 muestra esta compuerta.

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

FIGURA 2.16 COMPUERTA OR

Las leyes formales para la función OR son:

$A+\emptyset=A$ $\qquad$ $A+A=A$

$A+1=1$ $\qquad$ $A+\overline{A}=1$

Existen otras compuertas basadas en las tres compuertas principales, como son la compuerta NAND, NOR y la OR Exclusiva. La compuerta NAND es equivalente a la AND seguida de una compuerta NOT o inversor, la compuerta NOR tambien es equivalente a una compuerta OR seguida de un inversor y por ultimo la compuerta OR-Exclusiva se comporta como la OR excepto para el caso de entrada 1,1 que se excluye (la salida es "0"). El simbolo de estas compuertas, junto con su tabla de verdad correspondiente se muestra en la fig. 2.17

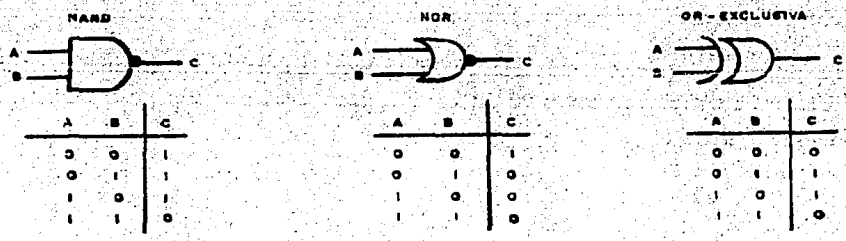


FIGURA 2.17 COMPUERTAS NAND, NOR Y OR-EXCLUSIVA

Las compuertas lógicas más utiles son empaquetadas como circuitos integrados, los cuales por lo general

pertenecen a la familia TTL (Transistor-Transistor Lógica), dentro de esta familia se encuentran circuitos integrados de alta velocidad y alta disipación (serie H), los Schottky (serie S), los Schotty de baja disipación (serie LS), de baja disipación y baja velocidad (serie L) y circuitos integrados estandars. La tabla de la fig. 2.18 muestra una lista de algunos circuitos integrados (TTL) disponibles comercialmente. El apostrofo antes del número de dispositivo sustituye la indicación de la familia y serie, por ejemplo la denominación '02 para los integrados de Texas Instruments, corresponderá a SN54LS/74LS02, etc, los integrados pueden tener dos, tres, cuatro o seis unidades idénticas pero independientes.

ARITMETICA BINARIA La combinación de circuitos lógicos (varias compuertas conectadas) sumaran, restaran, multiplicaran y dividiran según se requiera, basados en los circuitos sumadores y restadores. Es necesario tener presente las siguientes reglas:

N	DISPOSITIVO	DESCRIPCION
	'00	4 NAND de 2 entradas
	'02	4 NOR de 2 entradas
	'03	4 NAND de 2 entradas colector abierto
	'05	6 Inversores con colector abierto
	'08	4 AND de 2 entradas
	'09	4 AND de 2 entradas colector abierto
	'21	2 AND de 4 entradas
	'27	3 NOR de 3 entradas
	'30	1 NAND de 8 entradas
	'32	4 OR de 2 entradas
	'37	4 buffer NAND 2 entradas
	'38	4 buffer NAND 2 entradas colector abierto
	'126	4 buffer de tres estados
	'136	4 OR-Exclusiva 2 entradas

FIGURA 2.18 COMPUERTAS LOGICAS EN INTEGRADOS

ADICION BINARIA

Augendo	+	Aduendo	=	Resultado	Acarreo(Co)
0	+	0	=	0	0
1	+	0	=	1	0
0	+	1	=	1	0
1	+	1	=	0	1

RESTA BINARIA

Minuendo	-	Sustraendo	=	Resultado	Prestamo
1	-	Ø	=	1	Ø
Ø	-	1	=	1	1
1	-	1	=	Ø	Ø

Con base en lo anterior es posible entender lo que es un circuito combinacional, o sea un circuito que a base de compuertas lógicas realice las operaciones binarias más elementales como la suma y la resta.

SEMISUMADOR En la fig. 2.19 se presenta la tabla de verdad correspondiente a un circuito semisumador, su diagrama a bloques donde A y B son las entradas, y las salidas son: S como resultado de la suma y Co es el acarreo de salida. El semisumador se representa como SS, ó HA del ingles Half Adder (medio sumador).

ENTRADAS		SALIDAS	
A	B	S	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

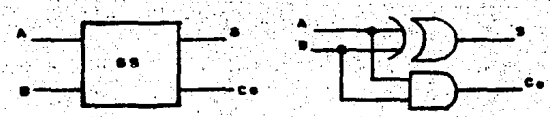


FIGURA 2.19 SEMISUMADOR

SUMADOR COMPLETO Los sumadores únicamente efectúan la adición de dos entradas (A y B), al hacer la suma de más de dos entradas es necesario un nuevo circuito, llamado Sumador Completo SC, ó FA de Full Adder, este circuito tiene tres entradas para la adición, las entradas son A, B y Ci (Acarreo de entrada) Carry in, las salidas del sumador completo son S el resultado de la suma y Co el acarreo de salida. En la fig. 2.20 se muestra la tabla de verdad del sumador completo, la cual es la misma que la del semisumador, pero se agrega una quinta regla. Se presenta el diagrama a bloques del sumador completo, el cual está formado por dos semisumadores y una compuerta OR.

ENTRADAS			SALIDAS	
Ci	A	B	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

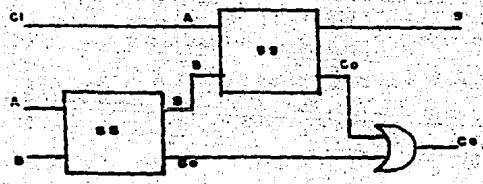


FIGURA 2.20 SUMADOR COMPLETO

SEMIRESTADOR En la fig. 2.21 se presenta la tabla de verdad de un circuito semirestador, su diagrama a bloques,

donde la entrada A es el minuendo y B es el sustraendo. La salida D es el resultado de la diferencia, mientras que la salida P es el prestamo, se representa el semirestador como SR ó HS (Half Substractor).

ENTRADAS (A-B)		SALIDAS	
MINUENDO	SUSTRAENDO	D	P
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

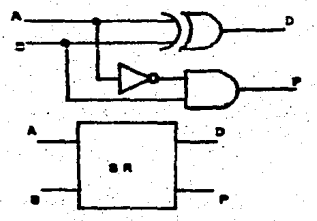


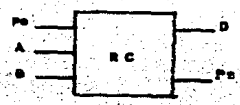
FIGURA 2.21 SEMIRESTADOR

RESTADOR COMPLETO Este circuito esta constituido por dos semirestadores y una compuerta OR, la fig. 2.22 muestra el diagrama a bloques la tabla de verdad y el circuito a base de compuertas. El restador completo RC ó FC (Full Substractor), tiene como entradas a: A minuendo, B sustraendo y Pe prestamo de entrada, y como salidas a: D diferencia y Ps prestamo de salida, las lineas Ps y Pe se conectan de restador a restador para transferir los prestamos.

La suma y resta binaria pueden realizarse por dos diferentes técnicas, pueden usarse sumadores en serie ó en paralelo,un sumador en serie, opera en forma parecida a la adición a mano, sumandose columna por columna más el acarreo anterior en cada una de ellas, la suma en serie toma una

considerable cantidad de tiempo cuando se suman números binarios muy largos, la adición en paralelo sin embargo, es más rápida. En la adición en paralelo todas las palabras binarias que se sumaran, se introducen en las entradas y la operación es casi inmediata.

ENTRADAS			SALIDAS	
MINUENDO	SUBTRAENDO	Pe	D	Ps
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



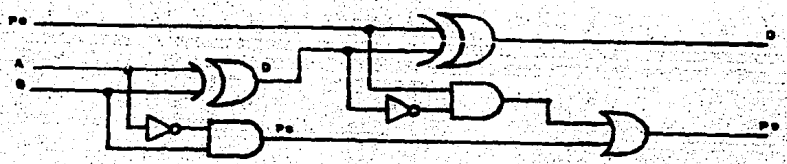


FIGURA 2.22 RESTADOR COMPLETO

Los sumadores en serie son más sencillos pero más lentos, los sumadores en paralelo son más rápidos pero más complicados.

En el aspecto práctico, los sumadores y restadores completos, se encuentran en forma de circuitos integrados en vez de armarse a partir de compuertas lógicas. Por lo

general, una unidad sumadora se representa por un simbolo de bloque como en la fig. 2.23 este es un circuito sumador completo de 4 bits, 7483 comercial, las entradas A1 y B1 son las entradas de los bit menos significativos y las entradas A4 y B4 son las entradas de los bits más significativos.

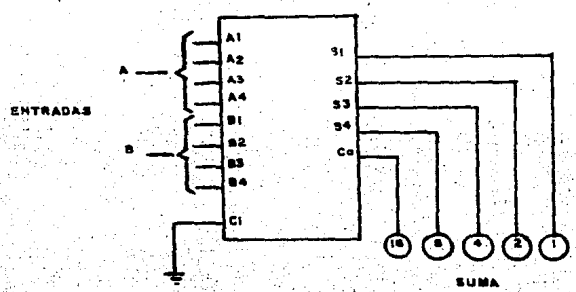


FIGURA 2.23 SUMADOR 7483

Es común conectar el (Ae) acarreo de entrada, a la tierra cuando no se encuentra conectado previamente a un sumador en paralelo.

Cuando la velocidad es importante deberán usarse sumadores en paralelo, donde se requiere simplicidad y menor número de componentes deberán usarse sumadores en serie, tanto los sumadores en serie como en paralelo se emplean en sistemas digitales.

COMPLEMENTO A DOS.

Por lo general, los números binarios se usan en las computadoras, algunas veces, sin embargo, se usa un codigo especial llamado notación de complemento a dos cuando se requieren números con signo, este sistema simplifica los circuitos de las computadoras.

Un registro común o posición de almacenamiento en una computadora se muestra en la fig. 2.24. Este registro cuenta con espacios para datos de 8 bits, el lugar de los bits esta numerado del 7 al Ø, en donde el bit 7 corresponde al bit de signo, el cual indicará si el número es positivo o negativo. Un cero en el lugar de el bit de signo, indicará que el número es positivo y con un 1 este indicará que el número es negativo.

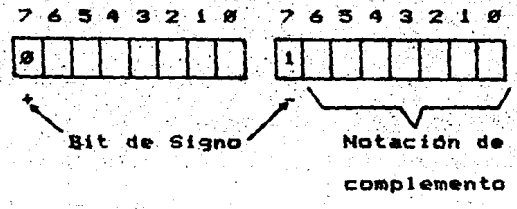


FIGURA 2.24 "COMPLEMENTO A DOS"

Si el número con signo es negativo, el registro

contendrá la forma complementaria a dos de este número, en el siguiente ejemplo tenemos el complemento a dos para el -1, llevando a cabo la siguiente secuencia:

1.- Se escribe en decimal el numero.
2.- Lo convierte a binario.
3.- Completamos cada bit.
4.- Sumamos un 1 al complemento.

```
                1
          0000 0001
          1111 1110
        +       1
        ----------
          1111 1111 = -1
```

La tabla de la fig. 2.25 presenta la notación complementaria a dos, para ambos números negativos y positivos.

Un microprocesador, puede usar números de complemento a dos, por que puede completar, incrementar (sumar +1 a un número) y sumar números binarios, los microprocesadores no tienen circuitos para sustraer, en su lugar usan un sumador y números de complemento a dos para hacer la sustracción como se vió anteriormente en la parte de resta binaria.

Decimal	Representación de Números con signo	
+ 127	0111 1111	
+ 126	0111 1110	
.	.	
.	.	
.	.	
+ 8	0000 1000	
+ 7	0000 0111	
+ 6	0000 0110	
+ 5	0000 0101	
+ 4	0000 0100	Números positivos representados igual que en su forma binaria.
+ 3	0000 0011	
+ 2	0000 0010	
+ 1	0000 0001	
0	0000 0000	
- 1	1111 1111	
- 2	1111 1110	
- 3	1111 1101	
- 4	1111 1100	Números negativos representados en forma de complemento a dos.
- 5	1111 1011	
- 6	1111 1010	
- 7	1111 1001	
- 8	1111 1000	
.	.	
.	.	
.	.	
- 128	1000 0000	

FIGURA 2.25 COMPLEMENTO A DOS

MULTIPLICACION BINARIA

A continuación se muestran las cuatro reglas fundamentales para la multiplicación binaria, correspondiendo esta tabla de verdad a una compuerta AND.

A	B	P
Ø	Ø	Ø
Ø	1	Ø
1	Ø	Ø
1	1	1

La fig. 2.26 muestra un problema de multipicación binaria, en donde en la parte izquierda, está la multiplicación decimal usando el método tradicional, y a la derecha se usa el mismo método para números binarios.

DECIMAL	BINARIO
13	1101 Multiplicando
x10	x 1010 Multiplicador
00	0000
13	1101
130	0000
	1101
	10000010 Producto

FIGURA 2.26 MULTIPLIACION BINARIA

La observación del proceso de la multiplicación binaria anterior, revela tres hechos importantes que son:

1.- Los productos parciales son ØØØØ si el bit multiplicador es Ø, ó igual al multiplicando si el bit multiplicador es un 1.

2.- El producto final es el doble de largo que el multiplicando.

3.- El primer producto parcial se recorre un lugar a la derecha, en relación al segundo producto parcial cuando se suma.

Basandose en estos hechos se desarrolla un multipiicador binario como se ve en la fig. 2.27 la cual muestra un circuito lógico binario para multiplicar dos números, usando el llamado método de ADICION y CORRIMIENTO.

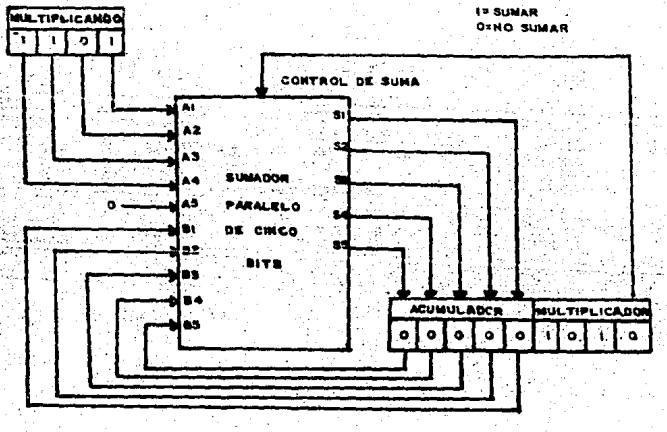


FIGURA 2.27 MULTIPLICADOR

El multiplicador binario consisten un sumador en paralelo de 5 bits, con un control sumar/no sumar, el

registro de corrimiento de 4 bits, arriba a la izquierda es el registro del multiplicando, abajo a la derecha se encuentra un registro acumulador de 5 bits y un registro multiplicador de 4 bits.

El procedimiento de suma y corrimiento, puede programarsele a una computadora ó puede ser diseñado como en la fig. 2.27 el procedimiento es el mismo en ambos casos.

CODIFICADORES Y DECODIFICADORES

La fig. 2.28 muestra un sistema digital, en donde la entrada, que es en forma decimal através del teclado, debe de ser traducida a una forma decimal codificada en binario (BCD). Este proceso se lleva a cabo por un dispositivo digital llamado Codificador, la traducción de decimal a BCD se llama codificación. La salida de la unidad central de proceso, es en una forma BCD, el decodificador traduce el dato de BCD a un código especial de despliegue, generalmente de 7 segmentos, para el usuario, el decodificador está traduciendo de BCD a decimal.

Un codificador puede ser un sólo circuito integrado pero también se puede construir a base de compuertas lógicas individuales, producir este circuito requiere de alrededor de diez a veinte compuertas lógicas.

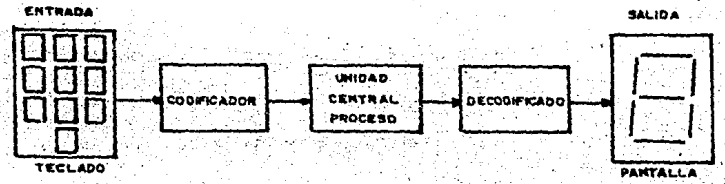
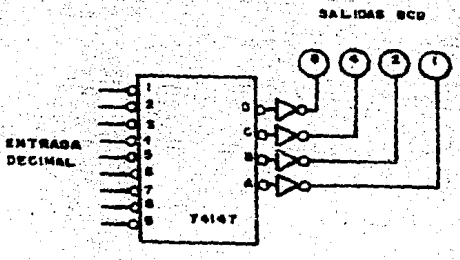


FIGURA 2.28 SISTEMA DIGITAL

El codificador que se muestra en la fig. 2.29 se llama codificador de prioridad de diez a cuatro líneas. Este dispositivo TTL se le denomina codificador 74147, cuya tabla de verdad se muestra en la misma figura, el codificador puede tener sólo una entrada activa que a su vez produce una salida única.



ENTRADAS									SALIDAS			
1	2	3	4	5	6	7	8	9	D	C	B	A
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

FIGURA 2.29 CODIFICADOR DE PRIORIDAD

La primera línea de la tabla de verdad, es para cuando no hay entradas, todas estas flotan en Alto, las salidas flotan en Alto y esto se interpreta como ØØØØ en los

indicadores de salida BCD.

La segunda linea de la tabla, muestra la entrada decimal 9, siendo activada con un nivel bajo ó Ø lo que produce Øiiø en las salidas D,C,B y A; los cuatro inversores invierten Øiiø y los indicadores BCD leen iØØi, que es la forma de representar al 9 decimal en BCD. La segunda línea de la tabla de verdad, muestra las entradas 1 a 8 marcadas con una X, que significa irrelevante, este codificador tiene un mecanismo de prioridad que activa el número mayor que tiene entrada Ø, si se colocara un Ø en las entradas 9 y 5, la salida seria iØØi, correspondiente al 9 decimal; el codificador sólo activa la salida de el número mayor.

Como se observa el circuito decodificador y de despliegue que se muestra en la fig. 2.3Ø corresponde a un decodificador BCD a 7 segmentos, el cual está traduciendo Øiii (BCD), a su equivalente de 7 decimal en el despliegue (Display)de 7 segmentos.

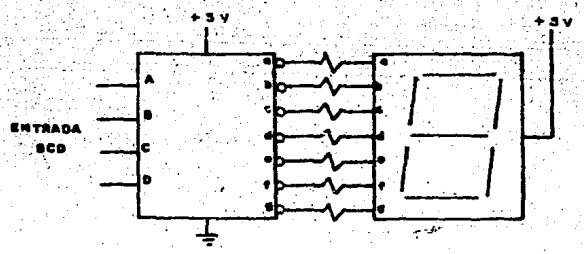


FIGURA 2.3Ø DECODIFICADOR BCD A 7 SEGMENTOS

Frecuentemente la información codificada debe de traducirse a otro código diferente, al circuito lógico que realiza esta traducción se le denomina conversor de código. Un conversor de código puede construirse, conectando en serie un decodificador y un codificador.

MULTIPLEXORES Y DEMULTIPLEXORES

Cuando se tienen varias entradas y una sola salida y se desea seleccionar una sola de las entradas para que pasea la salida, se utiliza un circuito Multiplexor, mientras que cuando se tiene el caso opuesto, ó sea una entrada y varias salidas y se desea activar con la entrada una sola de las salidas, se emplea un Demultiplexor. Debido a estas funciones, a los Multiplexores se les denomina también como Selectores y a los Demultiplexores se les llama como Decodificadores.

En un sistema a base de microprocesadores, todos los dispositivos que se comunican con el microprocesador tienen asignadas direcciones especificas. Los circuitos decodificadores aseguran que el dispospositivo correcto esté en comunicación cuando sea direccionado por el microprocesador. La ventaja que ofrecen, los circuitos decodificadores ó demultiplexores, reside en la capacidad de

poder seleccionar un número amplio de salidas con un número reducido de entradas, los demultiplexores se caracterizan por que el número de salidas esta relacionado con el número de entradas, como se muestra en la siguiente fórmula:

$$Y = A^2$$

donde: Y Es el número de salidas

A Es el número de entradas

El uso adecuado de los decodificadores, permite optimizar en un microprocesador, el uso de las líneas de dirección del mismo.

En un multiplexor, se tienen un número amplio de entradas hacia una sola salida valiendose de las líneas de selección del multiplexor, para poder seleccionar esta salida es necesario conocer las características ó funcionamiento del circuito multiplexor que se esta operando en particular. El diagrama lógico de un multiplexor/selector de datos de ocho entradas se muestra en la fig. 2.31 el multiplexor de ocho entradas, puede ser usado para convertir la entrada paralela de 8 bits, a na cadena en serie con un contador de tres bits conectado a sus entradas de selección de datos por lo cual las entradas son seleccionadas en secuencia (IØ,I1,..,I7). Los selectores ó multiplexores de datos se usan para resolver problemas complicados de lógica binaria.

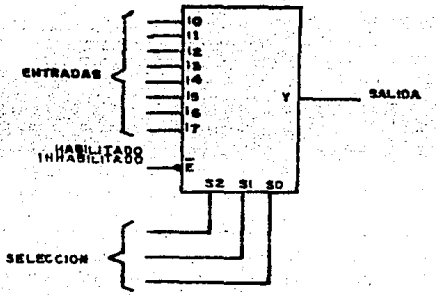


FIGURA 2.31 MULTIPLEXOR DE 8 A 1

La fig. 2.32 muestra un demultiplexor uno a ocho, los datos de entrada en la serie, pueden ser distribuidos a las 8 salidas en orden consecutivo, conectando las entradas selectoras de datos a un contador de tres bits.

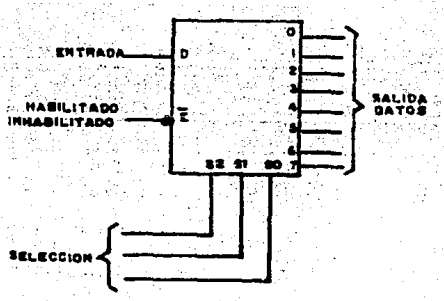
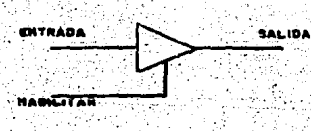


FIGURA 2.32 DEMULTIPLEXOR 1 A 8

LOGICA DEL TERCER ESTADO

En los sistemas digitales se tenia el problema, de interconectar varias unidades entre si, esto daba lugar a un gran número de lineas y circuitos, se penso usar una sola línea, esto dió lugar a la lógica del tercer estado, en donde aparte de los estados lógicos "1" y "Ø" se tiene un tercer estado llamado de alta impedancia, este estado adicional, es equivalente a desconectar las salidas no deseadas en un momento dado.

Los circuitos del tercer estado tienen una entrada, por la cual se les ordena que abandonen el tercer estado, conocida como habilitar ó deshabilitar, la fig. 2.33 muestra el simbolo de los circuitos de tres estados.



ENTRADA	HABILITAR	SALIDA
0	HABILITADO	0
1	HABILITADO	1
X	INHABILITADO	ALTA IMPEDANCIA

FIGURA 2.33 CIRCUITOS DE TRES ESTADOS

Se han desarrollado dispositivos especiales cuyas salidas pueden ser conectadas entre si a una linea compartida, estos dispositivos ó circuitos del tercer estado, se les denomina Acopladores ó Buffers. Un buffer es un circuito que se puede usar para incrementar la capacidad de

manejar las corrientes en una línea de un sistema digital, estos buffers proporcionan un aislamiento eléctrico entre componentes de un sistema, siendo, estos acopladores, de uso primordial en los puertos de entrada ó dispositivos de entrada de datos.

Con el uso de los acopladores de tres estados, se incrementa el número de puertos de entrada que se pueden conectar a las líneas de una microcomputadora, ya que mientras el microprocesador no le indique al puerto de entrada, éste no se "conecta" al sistema, es decir, el puerto de entrada se encuentra en el tercer estado y eléctricamente aislado del sistema.

Los primeros circuitos de tres estados ó TRI-STATE fueron desarrollados por National Semiconductors Corporation, teniendo las siguientes características:

- Serie 54/74 compatible TTL.
- Hasta 128 dispositivos pueden ser conectados a una línea común.
- 12 ns tiempo de propagación.
- Alta capacidad de mandar cargas capacitivas.
- Control independiente de cada acoplador.
- Se mejora la comunicación bidireccional en una línea común puesto que en el estado de alta impedancia, las entradas de los dispositivos de tres estados no presentan la

carga normal al dispositivo que las manda.

MEMORIAS

Todo sistemas computacional, durante su procesamiento de datos, debe continuamente almacenar y rescatar bits en una memoria. La memoria es la parte de las computadoras donde se almacena (codificadas en binario) todas las instrucciones ó datos de un programa, las memorias consisten en arreglos de elementos biestables.

A la memoria por su ubicación en un sistema de computo, se divide en dos áreas: Memoria primaria y Memoria secundaria.

La memoria primaria usualmente se encuentra instalada en una ó varias tablillas de circuitos impresos, la memoria secundaria se encuentra fuera de la computadora, se han desarrollado una gran variedad de tipos de memorias tanto primarias como secundarias. Las primarias pueden ser desde bulbos, transistores, nucleos magnéticos y semiconductores de mediana y alta escala de integración. Entre las memorias secundarias más comunes se tienen las tarjetas perforadas, cintas perforadas, cinta magnética, disco magnético (de cabeza móvil y fija), cassette y diskette ó floppy disk.

Las memorias de semiconductores de alta escala de integración, consisten basicamente en arreglos de "compuertas" que pueden ó no estar conduciendo, como se muestra en la fig. 2.34

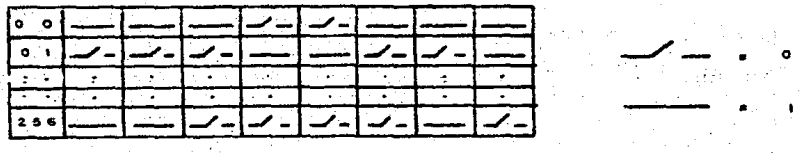


FIGURA 2.34 MEMORIAS DE SEMICONDUCTORES

Entre las características que proporcionan las memorias de semiconductores se pueden mencionar las siguientes:

a) Son más rapidas, más compactas y de bajo consumo de potencia.

b) Se pueden programar, microprogramación(almacenar programas permanentemente en memoria, utilizando un tipo especial de estas llamdas ROM).

c) Su versatilidad, su empaque y su compatibilidad con otros circuitos integrados las han hecho útiles para nuevos propósitos, completamente fuera de la tecnologia de las computadoras, se utilizan para almacenar palabras de control en máquinas de control numérico.

Las memorias usadas en los microprocesadores son del

tipo de lectura "no destructiva", es decir, al obtener la información de una localidad determinada, ésta no se modificará.

Conceptualmente la memoria, consiste en un número de posiciones ó localidades, en donde se puede leer ó introducir información, esta información se maneja convenientemente en grupos de bit, denominados palabras ó "Byte", las longitudes normales ó más comunes de las palabras, son de 8, 12, 16, 18, 24, 32 y 64 bits.

0 0 H	0	0	1	0	1	1	0	1
0 1 H	1	1	0	0	0	0	1	0
0 2 H	0	1	1	1	1	0	1	1
. . .	.	.	.	.	.	.	.	.
. . .	.	.	.	.	.	.	.	.
F F H	0	1	0	1	0	1	0	0

FIGURA 2.35 ESQUEMA DE UNA MEMORIA

Los circuitos de las memorias se diseñan de tal manera, que cada posición ó localidad permite el almacenamiento de una palabra de un formato determinado (4,8,16, etc. bits), una memoria puede representarse según el esquema de la fig. 2.35 en el que cada posición tiene 8 lugares para cada byte de 8 bits. La organización de la memoria es tal que contiene una colocación ordenada de los

bits, desde el bit más significativo MSB al bit menos significativo LSB.

Cada posición ó localidad en memoria, tiene una única Dirección, en la práctica, para los microprocesadores actuales emplean memorias que comprenden desde unos centenares de direcciones hasta 6400 ó más.

Las memorias, según su forma de acceso, se clasifican como:

a) De acceso Directo ó Aleatorio.

En estas memorias se asocia una dirección a cada palabra, y al suministrar a la memoria una, dirección, determina que se suministre ó modifique la información de la palabra asociada a dicha dirección en un tiempo que no dependa del valor de la dirección.

b) De acceso Secuencial.

Estas memorias se caracterizan, por que el tiempo de acceso a una palabra determinada, depende de su posición con respecto a la posición de referencia. El dato es accesible mediante una secuencia temporal.

c) Asociativas.

En estas memorias el acceso a una palabra determinada, se consigna mediante la información contenida en una parte de la propia palabra.

La fig. 2.36 muestra los tipos de acceso en memorias.

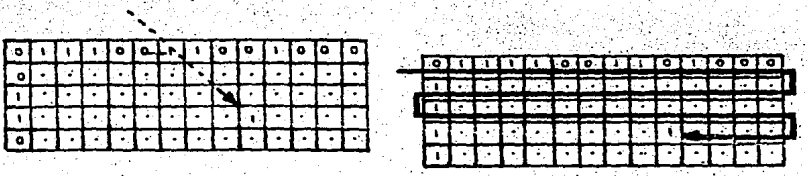


FIGURA 2.36 ACCESO

Según las operaciones que pueden efectuar con la información contenida en sus palabras, las memorias se clasifican en:

a) Vivas Se puede leer y modificar el valor de las palabras.

b) Muertas Se puede leer el valor de la palabra pero no se puede modificar.

Cuando al leer una palabra, se destruye la información que contienen sus celdas, la memoria sera "Destructiva", las memorias Destructivas serán también vivas. Las memorias que conservan los datos, aún después de desconectar la fuente de poder, se conocen como memorias No Volátiles, y a las que pierden todos los datos, en el momento en que se desconecta la fuente de poder se conocen como Volátiles.

MEMORIAS RAM (Random Access Memory)

Las memorias a las cuales se les puede cambiar el contenido de sus localidades con la función de escribir, lo mismo que obtener los contenidos de sus localidades con la función de leer, se llaman "memorias para leer-escribir". Esta clase de memorias se conocen comunmente como memorias de acceso aleatorio ó RAM.

Las memorias RAM pueden constituirse de un tipo de estructuras de flip-flops, a las que se conoce como memorias estáticas, ó pueden ser de estructura capacitiva a las que se conoce como memorias dinámicas. Las memorias RAM estáticas conservan la información, tanto tiempo como la energía este presente, mientras que las memorias RAM dinámicas se les debe refrescar cada pocos milisegundos para regenerar la carga almacenada en cada localidad.

Las memorias dinámicas requieren de un ciclo de refresco que utiliza del uno al cinco por ciento de tiempo, del tiempo total del procesamiento de una microcomputadora, esto puede ser importante en algunas aplicaciones de tiempo real, en las que se distinguen memorias que estan ocupadas y no disponibles de usarse mientras está en proceso un ciclo de refresco. Las memorias dinámicas son por lo tanto más baratas que las estáticas.

Las memorias RAM dinámicas, deben de estar refrescando la memoria, pues la carga y descarga de los capacitores ocacionan una fuga de corriente.

La fig. 2.37 muestra la estructura lógica, de una memoria RAM Estática. La cual es una memoria de 4 palabras de 2 bits cada una, los bits se almacenan en flip-flops, formados por dos inversores cruzados acoplados.

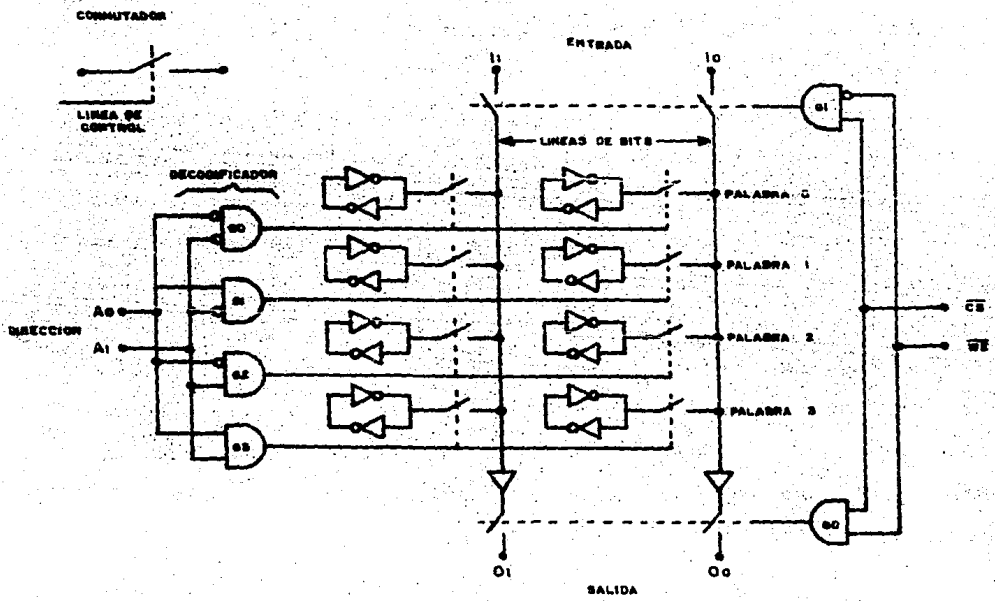


FIGURA 2.37 RAM ESTATICA

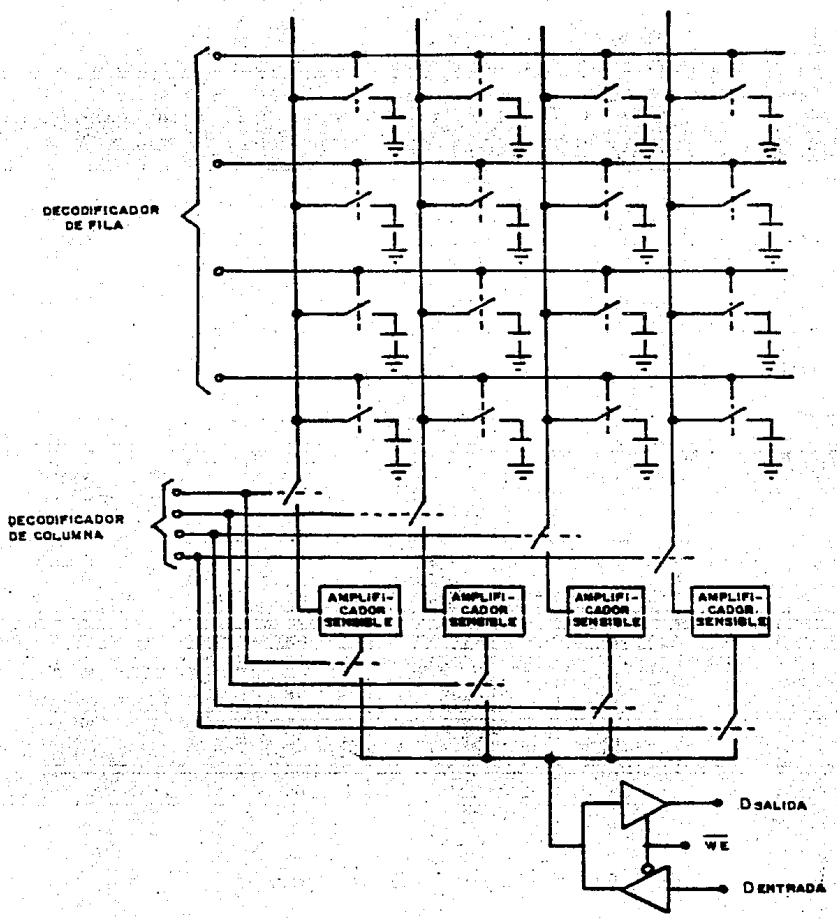


FIGURA 2.38 RAM DINAMICA

La fig. 2.38 presenta la organización de una memoria de 16 palabras de 1 bit, esta es una RAM dinámica, en donde

no es necesario refrescar cada capacitor, sino todos los capacitores de una fila pueden refrescarse simultáneamente, en donde cada fila debe de refrescarse cada 2 ms.

MEMORIAS ROM (Read Only Memory)

Una memoria de sólo lectura, es en la que no se puede escribir, pues el contenido de la memoria es fijo e inalterable, este contenido es establecido en el momento de la fabricación.

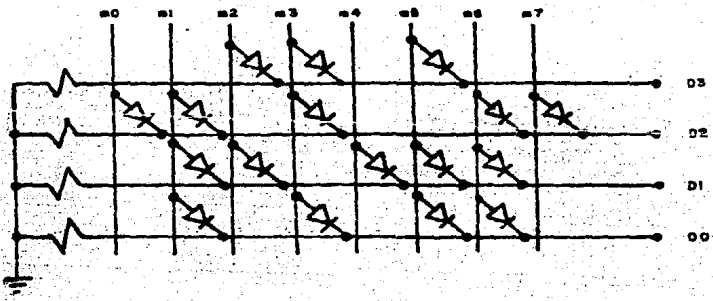


FIGURA 2.39 MEMORIA ROM

Las memorias ROM están formadas por una simple matriz de diodos, como se muestra en la fig. 2.39 y su programación se efectua una sola vez, fundiendo mediante la aplicación de alta tensión los diodos a descartar. La falta ó presencia de conexiones entre las líneas puede servir para indicar un

estado lógico "1" ó "Ø", un diodo ó su ausencia establecen un bit permanente, por lo tanto hay tantos diodos en la malla como 1 en la tabla de datos.

Como la memoria de lectura y escritura (RAM), la memoria de sólo lectura es de acceso aleatorio, por consiguiente es erróneo utilizar la etiqueta de Memoria de Acceso Aleatorio (RAM) para referirse únicamente a las memorias de lectura y escritura y no a las de sólo lectura.

Existen dos variantes en las memorias ROM que permiten más versatilidad al sistema, estas memorias se denominan; memorias de sólo lectura programables PROM y las memorias de sólo lectura programable borrable EPROM.

Las memorias PROM trabajan en forma parecida a las ROM, pero con la variante de que vienen vacías de fabrica, es el usuario quien las carga, con programas ó información de su interés y según sus necesidades, ésta carga se realiza por medios especiales, una vez cargada la memoria se comportará como otra ROM. Las memorias EPROM son semejantes a las memorias PROM, pero con la opción de borrar (generalmente con luz ultravioleta) lo previamente cargado, se pueden cargar y borrar varias veces, lo cual es de gran ventaja.

Una PROM se caracteriza por que en el proceso de

fabricación se forma una intersección de la rejilla de direcciones con las líneas de datos (la conexión consiste en un diodo ó un transistor de cualquier tipo). En serie con cada conexión el fabricante incluye un fusible que puede fundirse y por lo tanto abrirse al pasar una gran cantidad de corriente a su través.

En las EPROM el elemento utilizado en las conexiones, no es un diodo sino un transistor MOS, estos dispositivos suministran ó no conexión dependiendo de que exista ó no carga eléctrica en la compuerta de el transistor. La propiedad importante y distinta de estas EPROM es que la exposición a la fuerte radiación de rayos ultra violeta (alrededor de 30 minutos) permite la pérdida de las cargas de la compuerta, limpiando asi la memoria, despues de esto puede almacenarse nueva información en ella.

Otros sistemas para almacenar datos numéricos, que todabía no se han difundido comercialmente, son las memorias de burbujas magnéticas y las basadas en dispositivos de corrimiento de carga (CCD Charge Couple Devices).

En el caso de las burbujas magnéticas, los datos se memorizan aprovechando las propiedades de ciertos cristales, que bajo la influencia de un campo magnético, almacena lo que se conoce como burbujas dotadas de polaridad magnética. La

polaridad de las burbujas determina si se ha almacenado un 1 ó un Ø.

Las burbujas magnéticas están capacitadas para almacenar un elevado número de datos, los cuales, sin embargo, y para que el dispositivo funcione correctamente, han de ser memorizados en el equivalente de una bobina magnética cerrada sobre sí misma. Por consiguiente es necesario un elemento especial para escribir la información, más ó menos como en el caso de la cabeza de una grabadora, y otro elemento para leer la información. El lector siendo fijo, puede leer la información sólo cuando ésta se vuelve cíclicamente accesible debido al movimiento de el soporte.

Todo esto hace que el tiempo de acceso sea bastante largo, pero las burbujas tienen la ventaja de no ser volátiles, además la densidad de concentración permitida por este sistema, es de el orden de 1Ø veces mayor que una memoria de semiconductores.

Las memorias de este tipo están empezando a ser utilizadas por empresas norteamericanas como la IBM, así como por algunas firmas japonesas.

Las memorias por corrimiento de carga utilizan una técnica basada en los semiconductores, que las combierte en

algo parecido a larguísimos archivos, los datos se almacenan en una larga sucesión y se pueden accesar a ellos como en las memorias de burbujas, sólo cuando pasan frente a cabezas lectoras especiales. Existen varias semejanzas, entre las burbujas y los dispositivos por corrimiento de carga, a ambos se les considera memorias de acceso en serie. En los dos casos los datos almacenados, circulan en tuberías cerradas los dos emplean las cargas en un cristal de almacenamiento para representar los datos, y ambos se frabican en pastillas de semiconductores, por lo tanto, son completamente electrónicos, sin embargo el CCD es más veloz que las burbujas, el CCD es muy compacto y en el futuro puede ser de producción muy barata, pero desafortunadamente el almacenamiento CCD es volátil.

MEMORIAS MASIVAS O SECUNDARIAS

En sistemas digitales de gran tamaño ó de cierta sotisficación se pueden requerir capacidades de muchos millones de bits. Lograr y utilizar con efectividad estas grandes capacidades a base de memorias semiconductoras, aunque no imposible, es generalmente bastante inconveniente.

Los grandes sistemas no requieren de accesos rápidos a toda la información almacenada en memoria. Además es bastante aceptable que los tiempos de acceso de memorias de almacenamiento masivo, vayan de las decenas de milisegundos a

las decenas de segundo y aún a muchos minutos. Los datos e información que se necesitan inmediatamente, se encuentran en memorias de acceso corto.

Los grandes méritos de el almacenamiento masivo, son el abaratar sustancialmente el costo por bit de memoria y el suministrar una memoria no volátil.

Una característica fundamental de la utilización de las memorias masivas, es la forma de acceso, ó sea el método empleado para tomar una información (lectura) ó para grabarla (escritura), el tipo de acceso puede ser secuencial y directo.

CINTA MAGNETICA

Este es el método más popular de almacenamiento para grandes archivos a los que se tenga acceso en forma secuencial, los datos se almacenan como pequeños puntos magnéticos sobre un material de óxido de hierro, que cubre un lado de la cinta plástica. La cinta consiste en un soporte de plástico normalmente, de una anchura de media pulgada y de longitud aproximada de 2400 pies para un carrete de 10.5 pulgadas de diámetro, a lo largo de la cinta se pueden grabar una densidad de 1600 bits/pulgada, por lo tanto se pueden almacenar un total de:

(1600 bits/pulgada)(2400piesX12pul/pie)
= 46 millones de bits

Y si consideramos que la cinta, se encuentra dividida en 9 pistas, tendremos un total de 9 x 46 = 414 millones de bits. Una velocidad tipica para una cinta, es de 45 pulgadas/seg..

El tiempo de latencia, en el peor de los casos, suponiendo que la cinta esta en un extremo, y que la información requerida esta en el otro es de:
(2400 pies x 12 pulg./pie)(45 pulg/seg) = 10 minutos

DISCOS MAGNETICOS

En este sistema la información, se almacena en pistas concéntricas sobre un disco, en un disco cada pista es un circulo cerrado separado de los otros. Estos son unidades de memoria masiva de acceso directo.

El tiempo de acceso tipo, es de unos 30 milisegundos. La densidad de pista en un disco, es de unas 200 pistas/pulgada, y la densidad de bits/pista es de 4000, por lo tanto la capacidad de almacenamiento, en una memoria de disco que tenga 20 superficies esta en el rango de varios cientos de millones de bits.

El intercambio de datos entre las unidades de disco y la computadora, se efectua con velocidaes de transferencia, que va desde varias decenas de millares hasta 1000000-2000000 de caracteres por segundo.

DISCO FLOTANTE (Floppy Disk)

Comercialmente se dispone de memorias conocidas, como sistemas de disco flexible ó más familiarmente conocido Disco Flotante (Floppy Disk), el disco no necesita ser particularmente rigido, y de hecho puede ser delgado para que sea ligeramente flexible.

Los Floppy son dispositivos más sencillos de manejar y de costo inferior a los discos duros. Estos Floppy, son de material plástico flexible y van dentro de una funda, provista de una ventanilla que permite el acceso de las cabezas de lectura/escritura, este pequeño orificio redondo sirve para posicionar el disco en su alojamiento, a fin de obtener la alineación de las cabezas con el punto inicial de las pistas.

Tambien los floppy, estan divididos en pistas, y cada pista subdividida en sectores, la capacidad de un floppy varia en función de los siguientes elementos:

1 Grabación en una sola cara ó ambas

2 Grabación con densidad sencilla ó doble.

Un pequeño indice en el disco y cartucho, permiten el paso de la luz a través, cuando el disco esta en una posición angular determinada, definida para indicar el comienzo de la pista, el disco gira en su cartucho, cuyo interior esta fabricado de un material que minimiza la fricción entre ambos y ayuda a limpiar el disco. La capacidad total de almacenamiento de un disco flotante, es aproximadamente de 30 millones de bits por superficie, los tiempos de acceso están en el rango de cientos de milisegundos.

2.2.3 FUNDAMENTOS DE LAS COMPUTADORAS

Esas cajas negras que no trabajan solas, las computadoras definidas por A. M. Turing como "Una computadora es, ecencialmente un dispositivo que permite recibir, almacenar, manipular y comunicar la información".

Una computadora puede ejecutar operaciones aritméticas, escoger, copiar, mover, comparar y ejecutar otras instrucciones con los diversos simbolos de una forma deseada, siguiendo un mapa intelectual llamado programa, un programa es un detallado conjunto de instrucciones preparado para dirigir a la computadora y que ésta funcione de manera que proporcione el resultado deseado, una computadora es un rápido y exacto sistema de manipulación de simbolos

electrónicos, diseñado y organizado para aceptar resultados de salida bajo la dirección de un programa almacenado de instrucciones detalladas paso a paso.

Existen dos tipos básicos de computadoras, digitales y analógicas, las analógicas trabajan con señales electrónicas continuas, mientras que las computadoras digitales trabajan con señales electrónicas discretas. La principal característica de las computadoras digitales es su velocidad, debido a que estan construidas con componentes electrónicos de alta velocidad y que operan en base a señales de dos estados (binario). Las computadoras operan con circuitos binarios, por lo que para operar a altas velocidades deben ejecutar las operaciones aritméticas y lógicas utilizando el sistema de números binarios, esta característica binaria, hace que se tenga que utilizar algebra Booleana en el diseño de los circuitos necesarios para las computadoras digitales.

A continuación se describen algunos conceptos básicos requeridos para una mejor comprensión de la estructura de la computadora:

- BUS Es un camino por medio del cual se transfiere la información digital, desde una ó varias fuentes a cualquiera de varios destinos. En un tiempo determinado solamente puede tener lugar una de estas transferencias de información.

Mientras que se está produciendo una de estás transferencias, todas las demas fuentes que estan unidas a este bus deben de quedar bloqueadas.

El propósito fundamental de utilizar el concepto de bus es el de reducir el número de lineas de conexión requeridas para la transferencia de información, se utilizan tres tipos de bus para la información procesada.

BUS de DATOS Es en el cual la información digital (dato), es transmitida en forma bidireccional, entre la CPU, la memoria y otros dispositivos externos.

BUS de DIRECCIONES Es un bus unidireccional, mediante el cuel la información digital sirve para identificar ó bien una determinada posición de memoria ó un dispositivo determinado de E/S, en donde han de ser escritos ó leidos datos.

BUS de CONTROL Es el conjunto de líneas, que proporciona las señales que regulan el funcionamiento de un sistema digital, incluyendo la memoria y los dispositivos externos, estas señales pueden provenir de la CPU ó de un dispositivo externo. El bus comprende unas señalez que sincronizan las operaciones de E/S entre la CPU, la memoria y otros dispositivos externos, otras señales que controlan a la CPU, tales como interrupciones, espera y paro. Y otras señales que

controlan el acceso a los buses de datos y de direcciones.

DIRECCION Es un grupo de bits que identifican una posición de memoria ó dispositivo concreto de E/S.

HARDWARE .- El cual es toda la constitución física, de circuitos integrados interconectados sobre tarjetas de circuitos impresos agrupadas en elementos funcionales de características definidas, Memoria, Unidad de control, Entradas, Salidas y Unidad Aritmética Lógica, formando racks ó armarios con sus periféricos, impresoras, discos, consolas de visualización etc.

SOFTWARE .- Es el conjunto de instrucciones básico que define completamente al sistema desde el punto de vista de la programación, el software no se limita al conjunto de instrucciones sino que incluye también programas inteligibles por la computadora, escritos de acuerdo con el software básico y que permite interpretar en lenguaje más evolucionados.

Las computadoras pueden ser de Propósito General ó computadoras de programa almacenado, las cuales realizan practicamente cualquier trabajo, sea un cálculo científico ó el mando de una máquina herramienta, dependiendo precisamente del programa almacenado que realice una u otra función.

Las instrucciónes básicas las entiende la computadora al explorar secuencialmente su memoria, puesto que están escritas en lenguaje de máquina ó binario y el hardware de la computadora es capaz de interpretarlas.

En la actualidad las computadoras varian en su tamaño fisico generalmente entre más grandes, mayores son su velocidad de procesamiento, su capacidad de almacenamiento y su costo, los sitemas más grandes están mejor equipados para manejar un numero mayor de dispositivos de entrada y salida. Los sistemas pequeños como las microcomputadoras ó minicomputadoras, de uso general, pueden ejecutar las mismas operaciones y usar las mismas instrucciones de programa que muchas computadoras grandes.

La arquitectura de las microcomputadoras está formada por las siguientes partes:

a) Unidad Central de Proceso (CPU)

c) Unidad de Entrada

d) Unidad de Salida

La fig.2.40 muestra las partes de una microcomputadora.

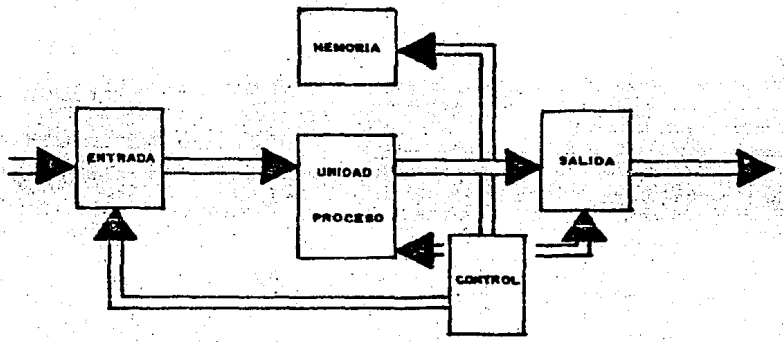


FIGURA 2.40 ESTRUCTURA DE UNA MICROCOMPUTADORA

La unidad central de proceso (CPU) es la parte principal de la microcomputadora y se puede dividir por sus funciones en la Unidad de Control y en la Unidad Aritmética y Lógica.

La Unidad de Control recibe las instrucciones desde la memoria y decide cuando, como y que operación se debe ejecutar para realizar cada instrucción, conoce cuando se termina la ejecución de una instrucción e indica cual es la que se debe ejecutar acontinuación. En ella se encuentra el contador de el programa con su sistema de aritmética de direcciones y su stack, con el puntero correspondiente que genera las informaciones para el bus de direcciones.

La unidad Aritmética y Lógica (ALU), puede recibir datos y efectuar con ellos operaciones aritméticas, lógicas, de comparación y corrimiento, entre otras. La ALU tiene

algunos registros para almacenar los datos sobre los que va a realizar las operaciones, el número exacto de estos registros depende de cada microcomputadora en particular, el registro principal de la ALU se conoce como Acumulador.

La unidad de memoria, contiene almacenadas las instrucciones y los datos que se van a procesar en la CPU. Al llegar a una dirección al registro de direcciones de memoria (MAR) a través del bus de direcciones, se obtiene unos datos en su registro de datos de memoria (MDR) que aparece en el bus de datos en una forma de lectura ó se introducen los datos en este registro a través de el bus de datos, en una operación de escritura en memoria.

Los dispositivos ó unidades de entrada se utilizan para alimentar los datos necesarios para los cálculos, lo mismo que los programas que indican a la microcomputadora las operaciones a desarrollar. La unidad de entrada no es mas que un multiplexor con el que se escoge con el bus de dirección y el de control cúal de las informaciones presentes se desea transladar al bus de entradas y salidas. Las unidades de entrada más comunes son: lectora de tarjetas, teclado, cinta magnéticas, disco, diskette, tubo de rayos catódicos (CRT).

Los resultados de las operaciones de la

microcomputadora se proporcionan al usuario por medio de las unidades de salida.

La unidad de salida recibe por el bus de datos la información a sacar al extremo y por el bus de direcciones la dirección por la cual debe sacar los datos, la unidad de salidad consiste en una serie de registros en los que se deposita esta información accionados por el bus de control.

2.2.4 EL MICROPROCESADOR

La forma más elemental de definir a un microprocesador es como un circuito integrado, capaz de ejecutar un programa y controlar las unidades necesarias para dicha ejecución.

En lo que respecta a la arquitectura de los microprocesadores, es decir, a la organización interna de los registros y las unidades operativas, todos ellos siguen esquemas comunes, en general, todos disponen de una unidad operativa, la ALU, un conjunto de registros de trabajo y un sistema de decodificación y ejecución de instrucciones.

Las principales aplicaciones del microprocesador son:

A.- Realiza la labor de Unidad Central de Proceso (CPU) en un sistema de microcomputadoras.

B.- La sustitución de circuitos lógicos que sólo son utilizables para una única tarea, por circuitos programables capaces de solucionar distintos problemas mediante distintos programas.

Los sistemas de microcomputadoras tendrán las características muy influenciadas por las del microprocesador en que se basan, ya que tanto su potencia como el resto de sus prestaciones estarán condicionadas por las características de su CPU constituida por el microprocesador, las principales características de un microprocesador son:

- Longitud de la palabra procesada.

Las longitudes más comunes de los microprocesadores actuales son de 8 y 16 bits, aunque también existen algunos que trabajan con palabras de 4 ó 32 bits. Cuanto más largas son las palabras tratadas mayor será la precisión de cálculo del microprocesador y su capacidad de direccionamiento.

- Capacidad de memoria.

Esta capacidad está potencialmente relacionada con la longitud de la palabra procesada. La capacidad máxima de memoria accesible, viene marcada por sus posibilidades de direccionamiento, no obstante, microprocesadores de igual

longitud de palabra pueden tener distinta memoria en su configuración inicial.

- Velocidad de ejecución de las instrucciones.

Se denomina ciclo de instrucción al tiempo que invierte el microprocesador en ejecutar completamente una instrucción, con esta característica queda determinada la velocidad de ejecución de un microprocesador. Otra medida es el ciclo de máquina, que refleja el tiempo empleado por el microprocesador en ejecutar una operación elemental de las que componen cualquier instrucción.

- Registros Especiales.

Otra característica importante de los microprocesadores, es el número de registros especiales que contienen. La mayoría disponen de un único acumulador en la unidad aritmética-lógica, no obstante, existen microprocesadores que incluyen dos acumuladores, con lo que se amplia su potencia y velocidad de operación. Así mismo existen dos tendencias en cuanto al resto de los registros internos, una consiste, en utilizar parte de la memoria RAM, como registros propios del microprocesador, la otra opta por incluir varios registros de trabajo dentro del propio microprocesador.

Un registro se define, como un circuito capaz de

almacenar una pequeña cantidad de bits (generalmente 8, ó sea un byte), el registro difiere del resto de la memoria, en que es mucho más fácil de direccionar y de manejar, utilizandose normalmente para operar con los datos durante la operación del microprocesador. Estos circuitos se borran y escriben frecuentemente durante la operación del microprocesador, mientras que la memoria lo hace pocas veces, por lo tanto un registro es una memoria de un byte.

Capacidad de Interrupción.

La ejecución de un programa puede ser interrumpida en algunas circunstancias, una característica básica del microprocesador, es la capacidad de recibir y gestionar un determinado número de interrupciones.

Mediante estas interrupciones se pueden establecer las comunicaciones necesarias, tanto con el usuario como con otras unidades del microprocesador, sin que ello efecte a la correcta ejecución del programa en su curso.

- Familia de Circuitos.

La necesidad de completar la operatividad del microprocesador, exige el empleo de unas series de circuitos integrados, adaptables al mismo. De esta manera surgen distintas familias de circuitos complementarios, por ejemplo la familia 6800 de Motorola ó la de 8060 de Intel.

El microprocesador trabaja directamente en lenguaje de máquina, no obstante si se dispone de los correspondientes traductores pueden utilizar determinados lenguajes de programación más evolucionados.

EL MICROPROCESADOR ZILOG Z-80

Microprocesador lanzado al mercado en 1976, por la compañía ZILOG, creado por un equipo que anteriormente habían desarrolado el 8080 en Intel, por lo tanto el Z80 esta basado en el 8080, siendo totalmente compatibles en software. Y se puede considerar al Z80 como un super 8080.

El juego de componentes del Z80, es superior tanto en hardware como en software comparado con cualquier otro microprocesador de 8 bits existente en el mercado.

Entre las características que presenta el Z80 estan:

a) Desde el punto de vista Hardware, ofrece una mayor facilidad de utilización, por los siguientes aspectos:

- Necesita una sola fuente de alimentación.
- Trabaja con un reloj de una sola fase, que puede ser de hasta 4Mhz.

- Genera una forma completamente decodificada, por patillas concretas de circuito, las señales de control de la memoria y de entrada/salida.

- Dispone de una señal para refresco transparente de memoria de tipo dinámico.

- Operación estática; la frecuencia de reloj puede disminuir hasta Ø.

b) Desde el punto de vista software el Z8Ø ofrece las siguintes ventajas:

- Una mayor riqueza de registros de trabajo.

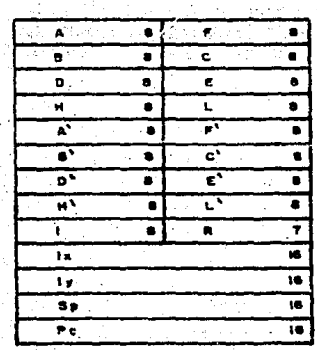
- Un conjunto de instrucciones ampliado.

- Una mayor flexibilidad en el sistema de interrupciones.

En lo que se refiere a los registros de trabajo el Z8Ø, cuenta con 18 registros de 8 bits y 4 registros de 16 bits.

Los registros del Z8Ø son llevados a cabo usando RAM's estáticas. Los registros incluyen dos juegos de 6 registros de propósito general, que pueden ser utilizados individualmente como registros de 8 bits, ó en pares como registros de 16. Cuenta con un registro acumuladore y registros de banderas.

La fig. 2.41 muestra el mapa de registros asi como el uso de cada uno de ellos.



En donde:

A .- Acumulador

F .- Registro de Banderas B,C,D,E,H,L .- Registros de propósito general

I .- Registro de Interrupciones

R .- Refresca memorias y genera números Aleatorios

Ix e Iy .- Registros de índice de acceso a Matriz

Sp .- Apuntador de la pila

Pc .- Apuntador del programa

Todos los registros primos son de respaldo

FIGURA 2.41 REGISTROS DEL Z8Ø

El Z80 puede ejecutar 158 tipos de instrucciones diferentes incluyendo las 78 del 8080. En este amplio conjunto de instrucciones se encuentran:

- Movimientos de registros
- Transferencia de bloques (memoria)
- Busqueda de bloques
- Aritmética de 8 y 16 bits (incluye Suma y Resta BCD)
- Se incluyen multiples modos de direccionamiento (directo, indirecto, relativo e inexado)

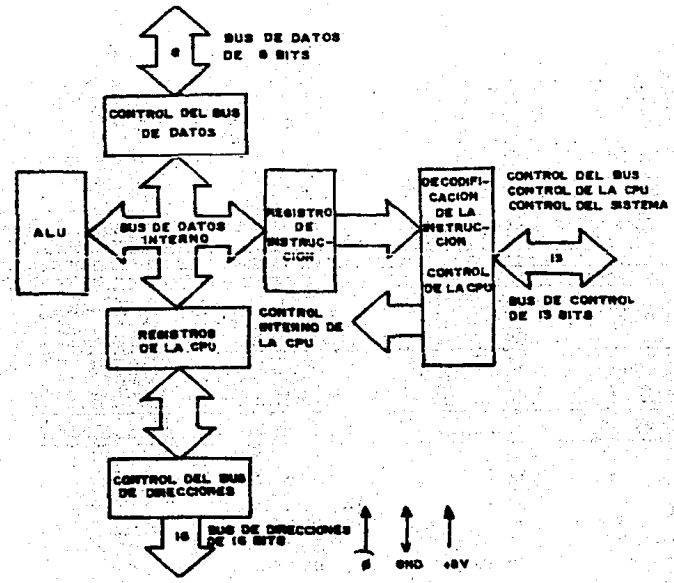


FIGURA 2.42 DIAGRAMA A BLOQUES DEL Z80

La fig. 2.42 muestra el diagrama funcional a bloques de la Unidad Central de Proceso CPU Z80.

Tal como la CPU Z80 ejecuta un programa residente en su memoria asociada, se lee cada instrucción en secuencia desde la memoria, colocando la dirección contenida en el registro contador del programa (PC) en el bus de direcciones, generando las propias señales de control en el bus de control, para activar la memoria, y leyendo entonces el dato en el bus de datos, para situarlo en el registro adecuado dentro del Z80.

Es claro que el tiempo es critico, para asegurar que el contenido de la memoria direccionada está en el bus de datos, cuando la CPU lee el bus de datos.

Las funciones de control del Z80, coordinan estas tareas y aseguran que los códigos de operación de las instrucciones, sean colocados en el registro de instrucción, y decodificados propiamente. Así mismo, esta función controla la ALU, para que realice todas las operaciones aritméticas y lógicas soportadas por el conjunto de instrucciones del Z80.

Estas operaciones incluyen, suma, resta, operaciones lógicas como AND, OR y OR-Exclusiva, comparación,

desplazamiento hacia la izquierda ó derecha y rotación, incremento, decremento, colocar a 1 un bit, colocar a Ø, y hacer una prueba de un bit.

Al realizar estas operaciones la ALU, se comunica mediante el bus de datos interno, con los 22 registros internos, el registro de instrucción, y el controlador del bus de datos. Los controladores de los Buses de datos y direcciones, vigilan todas las actividades relacionadas con el intercambio de datos entre el microprocesador Z8Ø y el mundo exterior, mediante sus buses respectivos.

2.2.5 LENGUAJES DE COMPUTACION

Los lenguajes de computación se clasifican en 4 clases que son:

- Lenguajes de máquina (números binarios)
- Lenguajes simbólicos directos (escritos en mnemónicos, correspondencia uno a uno entre mnemónico y número binario) llamado también ENSAMBLADOR (Assembler)
- Lenguajes de alto nivel funcionales ó algoritmos (escritos con mnemónicos, cada intruccion se convierte en un conjunto de instrucciones de máquina) representativos de estos lenguajes lo serían FORTRAN, ALGOL, PL/1 etc.
- Lenguajes de alto nivel conversacionales ó dialógicos (con una función parecida a los anteriores, pero cambia en

que son interactivos en la ejecución, creación ó modificación de instrucciones, un ejemplo sería el lenguaje BASIC.

El lenguaje que nos va interesar y con el único que se va a trabajar es con el lenguaje de programación del microprocesador. Un microprocesador realiza las operaciones y acciones que le especifica su programa.

El programa está formado por una secuencia de instrucciones, las cuales tienen un significado para la unidad de control del microprocesador, en este sentido puede ser el desencadenamiento de miniprogramas (microprocesadores microprogramables) ó la ejecución de acciones sobre registros ó puertos a través de un circuito cableado.

LENGUAJE DE MAQUINA

El conjunto de instrucciones válidas para un microprocesador, es lo que se denomina lenguaje de máquina ó abreviadamente lenguaje máquina. Programar en lenguaje de máquina supone por lo tanto escribir secuencias de números en binario, que son directamente decodificadas por el circuito de la unidad de control e interpretadas por el microprocesador.

Los códigos de operación son difíciles de recordar en

binario, es necesario usar una tabla de equivalencias, la codificación se hace lenta y tediosa, por los números en binario. Las direcciones en binario de los operandos, de las instrucciones son también difíciles de recordar. Teniendo en cuenta que muy probablemente el programa, no funcione a la primera, resultará difícil seguir las instrucciones de prueba a través de direcciones en binario.

Un programa de máquina sólo puede ser trasladado a otro microprocesador igual al primero. El problema de incompatibilidad no se plantea quizas al equipo que se diseña por primera vez, pero se presentará al querer hacer cambiar de microprocesador, por razones de precio ó nuevas necesidades de diseño.

Un programa en lenguaje de máquina, está tan densamente codificado, que es imposible de entender tanto por su propio autor al cabo de un cierto tiempo, como para los posibles interesados en adaptarlo para un sistema parecido.

La programación en lenguaje de máquina, puede sistematizarse y mejorarse con una metodología adecuada. El análisis previo, La confección de diagramas de flujo, el escribir previamente el programa en un lenguaje de máquina, efectuar tablas de símbolos y el empleo de sistemas octal y hexadecimal, pueden constituir una innegable ayuda. La

automatización de estas ayudas a la programación, se concreta en la utilización de lenguajes ensambladores.

LENGUAJE ENSAMBLADOR

Con el nombre de ensamblador se conocen dos cosas muy distintas. Se llama ensamblador, a un lenguaje simbolico en que se puede escribir programas para un microprocesador. Recibe el mismo nombre el programa traductor encargado de convertir (ensamblar) los programas escritos en lenguaje simbolico en programas objeto en lenguaje de máquina. El ensamblador proporciona tres grandes ayudas, permite utilizar simbolos (mnemónicos) para designar operaciones y nombres para designar direcciones y especificar datos (constantes) en otras formas que binario puro.

Cada ensamblador tiene su lista prefijada de símbolos para las instrucciones, esta lista puede ser fija ó expandible. En el lenguaje ensamblador, se puede asignar un nombre a una dirección utilizando éste nombre como etiqueta, el programa ensamblador hace equivalente los nombres con las direcciónes, el nombre es libremente inventado por el programador y sólo esta limitado en longitud y en lo que se refiere a su primer caracter, para el programador esto representa poder dominar a las direcciones por un nombre relacionado con el significado de su contenido, la

legibilidadd del programa aumenta considerablemente y el ensamblador pasa a manejar automaticamente todas las direcciones relativas.

La inserción ó eliminación de una instrucción no representa problema, por cuanto el ensamblador pasa a manejar todas las instrucciones y recalcula sistematicamente todos los desplazamientos (diferencias de dirección) del programa.

En el manejo de los direccionamientos (absolutos, relativos, indirectos, inmediatos) el ensamblador tambien ayuda al programador que proporciona una serie de símbolos especiales como: *,.,X,, que los representa en el programa fuente.

La tercer ayuda, es la especificación de constantes, proporciona la posibilidad de introducir directamente datos de distintos tipos como decimal, octal, hexadecimal, caracter (ASCII, BCD), y coma flotante. Además proporciona la posibilidad de algún tipo de aritmética (suma, resta y multiplicación como mínimo) sobre las direcciones.

El ensamblador, permite la incorporación de comentarios al texto mismo del programa. Los comentarios son una ayuda a la documentación de los programas, pueden ser frases cortas explicativas ó la versión en algún lenguaje de

alto nivel del mismo programa, conversando en la medida de lo posible.

Este tipo de documentación es muy fácil para hacer nuevas versiones, ya sea en lenguajes de alto nivel ó en otro lenguaje ensamblador.

Muy importante es la carga de los programas, ó sea la operación de llevar el programa ya traducido a memoria, para ser ejecutado ó probado, el ensamblador deja el programa traducido ya directamente en memoria (RAM) ó produce alguna salida sobre algún medio físico (cinta, cassette, floppy). Paralelamente produce un listado del programa con los comentarios, la traducción de las instrucciones y una tabla de equivalencias entre nombres y direcciones del programa.

Dentro del lenguaje ensamblador, se tienen los llamados extensiones del ensamblador, los cuales son: el ensamblador reubicable y el macroensamblador, desde el punto de vista lenguaje fuente no son muy distintos a éste.

El ensamblador reubicable produce un código que no es código objeto puro(lenguaje de máquina ejecutable). Necesita un último proceso, realizado por un programa llamado cargador reubicable, que permite colocar el programa en cualquier lugar de la memoria y no en una posición fija determinada al

escribir el programa fuente.

Se tiene ertonces que una instrucción en lenguaje ensamblador es traducida al lenguaje de máquina ó instrucción de código de máquina.

En el lenguaje macroensamblador, se permite dar un nombre a un conjunto de instrucciones (macroinstrucción). Después de la definición de una de estas instrucciones en el programa, el macroensamblador inserta el conjunto de instrucciones definido, expandiendo una linea del programa fuente a varias del programa objeto, el resultado es algo parecido a una subrutina pero sin los tiempos extras de llamadas y retornos.

LENGUAJE DE ALTO NIVEL

La utilización de un lenguaje de alto nivel reduce los costos de programación, incrementa la fiabilidad de la lógica programada (Software) simplifica el mantenimiento y documentación de los programas si los comparamos con la utilización de lenguajes de máquinas.

La utilización de lenguajes de alto nivel supone la utilización de volúmenes de memoria que son desde un 10 a un 100 por ciento mayores que los que necesitaría un programa equivalente en ensamblador.

Al pasar el programa en lenguaje de alto nivel, pasamos a manejar ya no el microprocesador directamente, con su estructura de registros, acumuladores, pilas y puertos sino un microprocesador de estructura distinta, encomendado no para ser manejado fisicamente, si no para adaptarse a la solución de los problemas planteados.

Las instrucciones contienen directamente expresiones aritméticas y lógicas y los datos pueden estructurarse, por lo cual se necesitan programas traductores bastante complejos llamados "Compiladores", para generar programas en códigos de máquina a partir de sentencias en lenguajes de alto nivel. Las ventajas que presentan los lenguajes de alto nivel son varias, como ejemplo tenemos que los programas escritos en lenguaje de alto nivel son muy compactos, se entiende mucho más fácil lo que se hace en cada sentencia ó grupos de sentencias, rapidez en la detección y correción de errores, aumento en la codificación de los programas.

Una gran ventaja de los lenguajes de alto nivel sobre los ensambladores, es su vida media, mientras el lenguaje ensamblador cambia para cada nueva generación de microprocesadores, el lenguaje de alto nivel es independiente de estos cambios. Desarrollando el compilador adecuado se pueden tener todos los programas escritos en este lenguaje,

adaptarlos a un nuevo microprocesador, la independencia de los programas respecto al microprocesador utilizado para una aplicación, no es sólo algo deseable, sino que es una necesidad por el desarrollo del software. Actualmente, el desarrollo de nueva programación es mucho más costosa y lenta que la adopción de un nuevo microprocesador.

Un lenguaje de alto nivel, puede no proporcionar acceso a ciertas caracteristicas deseadas del microprocesador, en estos casos hay que utilizar forzosamente el lenguaje ensamblador. La no utilización del compilador puede venir dictada tambien por medidas económicas, el problema es el exceso de memoria que para un programa dado ocupa el código generado por el compilador.

Hay un punto para un cierto número de sistemas fabricados, en el que se equilibran ambos factores, dado que el precio de la memoria bajará en los proximos años este punto no cesará de desplazarse a favor de la utilización de lenguajes de alto nivel.

Toda una serie de empresas de software ofrecen los lenguajes de alto nivel (FORTRAN, COBOL, BASIC, ALGOL) con ciertas restricciones para microprocesadores más populares, estos lenguajes tienen el incombeniente de que fueron creados en un ambiente totalmente ajeno a las necesidades del

microprocesador.

Primero Intel y más tarde Motorola presentaron lenguajes más apropiados a las necesidades de sus usuarios, como por ejemplo el PL/M aunque creado inicialmente por Intel para su serie 8000, existe ahora para otros microprocesadores como Motorola 6800.

Motorola presentó en 1975 el MPL, un lenguaje de alto nivel para el 6800, ofreciendo la posibilidad peculiar del fácil manejo a nivel de bit y de introducir sentencias en lenguaje ensamblador. Fairchild y National Semiconductor tienen el PL-MIPROC, una extensión del ALGOL y MITS ha dotado a su Altair 8800 de un super-BASIC y Texas Instruments para su nuevo microprocesador tiene preparados compiladores de COBOL, FORTRAN y BASIC.

ALGUNOS LENGUAJES DE ALTO NIVEL

Los lenguajes de alto nivel más usados son:

FORTRAN .- (FORmula TRANslation) nace en 1954, ideado por Jhon Backus, siendo un lenguaje concebido para uso científico y sus principales características son:

- Necesidad de pocas instrucciones fundamentales.
- Escasa necesidad de tratar con textos.
- Los datos son normalmente números en forma

exponencial.

- La cantidad de datos a elaborar es reducida mientras puede ser notable el número de elaboraciones necesarias.

- Necesidad de disponer de instrumentos de cálculo especiales (Algoritmos matematicos).

La utilización de este lenguaje para fines comerciales ó de gestión no debe excluirse a priori, aunque resulte incomodo y trabajoso, sobre todo por la falta de instrucciones que permite elaborar caracteres alfanuméricos.

Este lenguaje fue el primero ampliamente utilizado. Estructuras elementales de control y poderosas subrutinas para problemas matematicos.

COBOL .- (Common Business Oriented Language) inventado en 1959, es un lenguaje concebido exclusivamente para fines comerciales, procesamiento de datos en lote, en administración de empresas. Sus principales caracteristicas son:

- Auscencia de instrucciones de cálculo (excepto las operaciones principales)

- Posibilidad de una buena gestión de datos en disco.

- Instrucciones destinadas a la impresión de informes de tipo económico.

- Estructuras elementales de control.

BASIC .- (Beginners All Purpose Symbolic Instruction Code) nace en 1964 ideado por Kemeny John y Tommas Kurtz, siendo un lenguaje para principiantes. Su uso es general elemental y científico-numérico, de sintáxis sencilla con estructuras elementales de control y algunos recursos avanzados de proposito general.

Es un lenguaje apto para objetivos científicos y comerciales, ya que además de las caracteristicas del FORTRAN, tiene una notable capacidad para gestionar textos y diversas instrucciones para la expresión de informes económicos. La enorme proliferación del uso del BASIC ha traído como consecuencia lógica, el nacimiento de muchos dialectos pero las principales y más importantes instrucciones se mantienen iguales de una versión a otra.

Otros lenguajes de alto nivel usados son ALGOL, LISP, APL, PL/1, FORTH, PASCAL, C, ADA, MODULA-2, RPG y otros.

CAPITULO III

COMUNICACION DE DATOS

3.1 TEORIA BASICA

El principal atributo de la comunicación es la transferencia de información desde un punto a otro. En sistemas de comunicación de datos generalmente se llama a esta información dato ó mensaje.

Para enviar un mensaje de un punto a otro se requiere de un sistema que contenga los siguientes componentes: una fuente para generar el mensaje, un medio de transmisión para transferir el mensaje, y un elemento receptor fig. 3.1.

Estos elementos son los requerimientos minimos para un proceso de comunicación. Estos elementos pueden presentarse en muchas formas diferentes del sistema de comunicación.

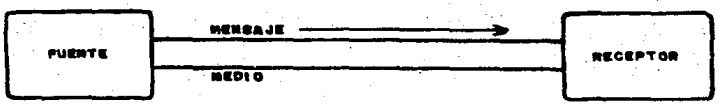


FIGURA 3.1 COMUNICACION

El término red de comunicación se entiende muchas veces como un arreglo de redes fig. 3.2. Una red de comunicación de datos usualmente involucra una computadora con una ó más terminales conectadas a través de una linea de comunicación.

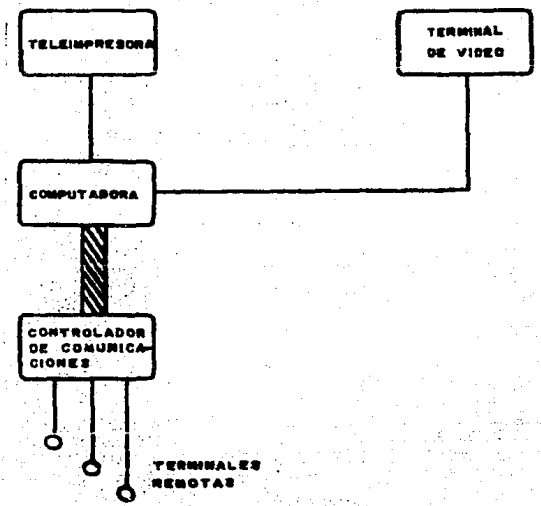


FIGURA 3.2 RED

3.2 TIPOS DE TRANSMISION

3.2.1. TRANSMISION EN UN SENTIDO (ONE-WAY).

Esta linea consiste de uno ó varios canales, en donde el canal es definido bajo un sólo sentido de transmisión. Un canal puede llevar información en cualquier sentido pero

solamente puede llevar una al mismo tiempo. El sentido del flujo de información es determinado por las caracteristicas del mecanismo en cada terminación de canal.

La difusión a través del radio y la televisión, son un ejemplo de transmisión en un sólo sentido. Las señales enviadas por una estación radio difusora pueden ser recibidas mediante un receptor en nuestros hogares. La televisión y la radio reciben información, pero no la pueden enviar ó transmitir posteriormente porque los receptores de nuestros hogares no estan diseñados para transmitir, asi como tampoco está diseñada una estación radiodifusora para recibir información.

Un ejemplo eléctrico de un sistema de comunicación de un sólo sentido es mostrado en la fig. 3.3. Cuando pulsamos el interruptor la lámpara del lado "b" encendera. Esto quiere decir, que cuando activamos el interruptor (lado "A"), estamos enviando información hacia el lado "B".

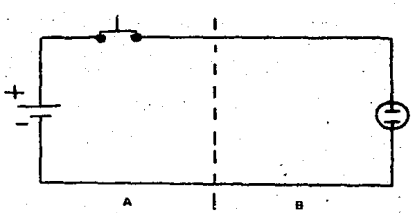


FIGURA 3.3 COMUNICACION UN SOLO SENTIDO

La transmisión en un sólo sentido suele utilizarse poco, porque en casi todas las conexiones ordenador periférico es necesario que las informaciones viajen en ambos sentidos: una impresora, por ejemplo, debe informar al ordenador que ha encontrado un error en el buffer que contiene los caracteres a imprimir.

3.2.2 TRANSMISION EN UN SOLO SENTIDO AL MISMO TIEMPO (HALF-DUPLEX)

Este tipo de transmisión hace que los equipos de terminales sean más rentables, porque puede alternar la dirección del flujo de datos a lo largo de un canal. Es decir, le permite al equipo enviar ó recibir información a través de un canal. Puede inicialmente enviar desde un punto "A" a un punto "B" y al terminar de enviar el mensaje, se puede cambiar el sentido; el punto "B" como si fuera transmisor y "A" como receptor.

Un puente de un sólo carril en una carretera es un sistema Half-Duplex: puede llevar información en cualquier dirección, pero solamente en una dirección al mismo tiempo, por lo tanto, la dirección del flujo de dato es determinado por la entrada/salida que tenga el puente.

Un sistema Half-Duplex tipico de comunicación

eléctrico simple es mostrado en la fig. 3.4. Con el modo de transmisión en el lado "A" y el lado "B"en modo recepción; "A" puede comunicarse con "B" presionando el interruptor del lado "A" (enviandose corriente a través del circuito a la lámpara "B"). Al terminar esta comunicación se puede cambiar de posición los dos interruptores (transmisor/receptor) a la posición contraria y entonces "B" se podrá comunicar con "A" presionando el interruptor del lado "B" (enviandose corriente a través del circuito a la lámpara "A").

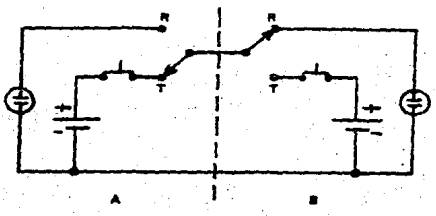


FIGURA 3.4 HALF-DUPLEX

Hay dos puntos que pueden observarse a través de este sistema de comunicación:

a)Se usa solamente dos hilos para conectar a "A" y "B".

b)La acción de invertir el sentido de flujo de datos toma una cantidad finita de tiempo.

3.2.3 TRASMISION EN AMBOS SENTIDOS (FULL-DUPLEX)

Organizando una linea de comunicación con dos canales, podemos tener capacidad para enviar información en ambas direcciones al mismo tiempo. Usualmente un canal lleva información en una dirección, y el otro canal lleva información en dirección contraria.

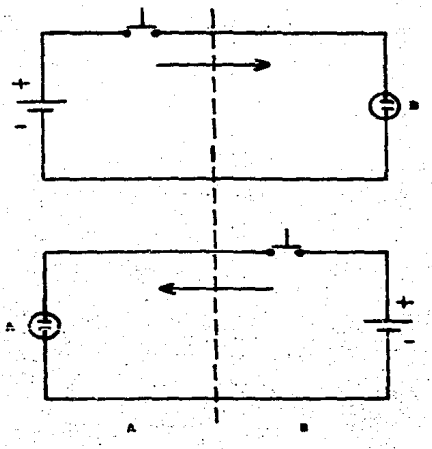


FIGURA 3.5 FULL-DUPLEX

Un ejemplo eléctrico de este sistema es mostrado en la fig. 3.5. Este sistema es capaz de trazar dos flujos de datos simultáneos. Al pulsar el interruptor del lado "B", se envía corriente a través del circuito encendiendose asi la lámpara "A"; al pulzar el interruptor del lado "A", se envia corriente a través del circuito encendiendose entonces la

lámpara "B". En este sistema se tienen cuatro hilos.

Es muy común en comunicaciones de datos, que las terminales Half-Duplex esten conectadas por dos canales.

La popularidad de esta organización se debe a dos razones:

1) Muchas lineas de redes telefónicas tienen dos canales para la transmisión.

2) Porque en esta configuración puede minimizarse el tiempo de regreso al sistema.

3.3. CODIGOS DE TRANSMISION

En los sistemas de comunicación de datos usualmente se desea trasmitir un flujo de caracteres (letras, números ó simbolos especiales) desde un punto a otro.

Los códigos en comunicación de datos estan basados en el sistema binario. En comunicación de datos, el término binario es usado para describir alguna condición de existencia entre dos estados diferentes. Por ejemplo, el interruptor de la luz en un salón puede tener dos estados: encendido ó apagado. Los dos estados usados en comunicación

son el cero (Ø) y el estado uno (1). Por consiguiente, usando este sistema binario de "Ø" ó "1", se puede codificar el mensaje dentro de una cadena de "Ø" y "1" y puede ser trasmitidos a lo largo de una linea de datos decodificandose posteriormente por un receptor.

Un código esta limitado por el número de bits que contenga, el término bit esta contenido en las palabras digito binario. En comunicación de datos, la unidad más pequeña de información es el bit. Puede al mismo tiempo llamarse a este bit, como un elemento de nivel, así la capacidad del caracter manipulado en el código es limitado por el número de elementos de bits ó niveles que contenga el código.

Por ejemplo, un bit de código significa que puede tener dos caracteres, así que puede decodificarse la letra A y la letra B, donde la letra A esta representada por el estado "Ø" y la letra B es representada por el estado "1".

Un código de dos elementos puede manipular hasta cuatro caracteres, puede decodicar por ejemplo, la letra A con la combinación binaria "ØØ", la letra B será la combinación "Ø1", la letra C será la combinación "1Ø" y la letra D será la combinación "11". Se puede tener además un código de 3 bits decodificando 8 caracteres, porque son 8

combinaciónes posibles de estos 3 bits. En general se puede tener un código de N niveles, pudiendose decodificar 2^N caracteres.

3.3.1 CODIGO ASCII (AMERICAN STANDAR CODE FOR INFORMATIUN INTERCHANGE)

Este es un código de ocho niveles ó ocho bits en los cuales siete bits son de información, y un bits para chequeo de paridad.

El código ASCII es el más ampliamente utilizado en códigos de transmisión de datos. Hay un gran número de versiones estandarizadas con diferentes nombres, pero básicamente estas se refieren al mismo código.

Los siete niveles de información nos proporcionan 128 combinaciones, esto nos permite decodificar una alta escala incluyendo caracteres alfanuméricos, gráficos y de control. Un método común de representación de estos caracteres es mostrado en la fig. 3.6.

Esta grafica coloca el caracter poniendolo afuera en 8 columnas y 16 filas. Las columnas estan numeradas del Ø al 7, la representación binaria del número de la columna correspondiente a los tres bits más significativos de el

séptimo bit del caracter patrón. Las filas estan numeradas del Ø al F en Hexadecimal y la representación binaria del número de la fila corresponde a los 4 bits menos significativos de el caracter. Cuando se escribe en código ASCII una combinación en binario, se toma como regla general los bits más significativos del 1 al 7 el último bit menos significativo será el de la derecha.

Bits b7 b6 b5 / b4 b3 b2 b1 / Columns	$0\,0\,0$	$0\,0\,1$	$0\,1\,0$	$0\,1\,1$	$1\,0\,0$	$1\,0\,1$	$1\,1\,0$	$1\,1\,1$
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS ←	CAN	(	8	H	X	h	x
1 0 0 1	9	SKIP HT	EM	)	9	I	Y	i	y
1 0 1 0	A	LF ↓	SUB	*	:	J	Z	j	z
1 0 1 1	B	VT ↑	ESC	+	;	K	[	k	{
1 1 0 0	C	FF →	FS	,	<	L	\	l	|
1 1 0 1	D	CR	GS	-	=	M	]	m	}
1 1 1 0	E	SO	HOME RS	.	>	N	^	n	~
1 1 1 1	F	SI	NEW LINE US	/	?	O	_	o	DEL RUB

FIGURA 3.6 CODIGO ASCII

Hay dos métodos comunes para identificar algún caracter ASCII en la grafica, el primero puede usar la representación binaria del caracter para identificarlo: por

ejemplo, el bit patrón 1001000 corresponde al caracter "H". El otro método es utilizar el número de fila y columnas para identificar el caracter. Por ejemplo, el número 48 en Hexadecimal representa tambien al caracter "H", esta localizado en la columna 4, fila 8.

3.4 MODOS DE TRANSMISION

3.4.1 TRANSMISION EN SERIE Y PARALELO

Los datos pueden ser transferidos en serie mediante una línea simple (más el retorno), ó en paralelo usando varias líneas al mismo tiempo. En ambos casos, las transferencias pueden ser Sincronas, del tal modo que se pueda predecir con exactitud la partida ó llegada de cada bit de información, ó puede ser Asincronas, caso en el cual los datos se transmiten por periodos uniformes.

En la transmisión en paralelo, todos los bits de un caracter decodificado son trasmitidos simultáneamente, esto significa que cada nivel de código tiene un canal exclusivo. Por lo tanto, para caracteres ASCII se necesitarán 8 canales. La fig. 3.7 muestra como todos los bits de un caracter dejan simultáneamente la fuente y como estos llegan al mismo tiempo al receptor.

La transmisión en paralelo es utilizada para transmitir datos en distancias cortas (transmisión entre una computadora y dispositivos periféricos). La principal ventaja que tiene en la transferencia de datos es su rapidez. Debido a que esta transmisión necesita de un número mayor de línea para la tranferencia de datos comparada con la transmisión serie, esto ocasiona que sea su principal desventaja.

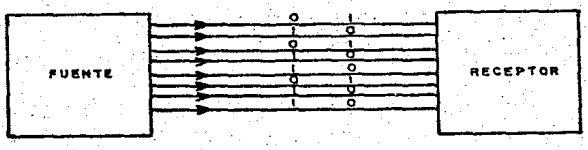


FIGURA 3.7 TRANSMISION PARALELO

La transmisión serie es por ahora el método más comunmente utilizado. El la transmisión serie, los bits del caracter decodificado son enviados uno tras otro a lo largo del canal, como se muestra en la fig. 3.8.

Este tipo de transmisión es utilizado para transmitir datos en distancias grandes. La reducción de lineas a utilizar en la trasmisión de datos, es su principal ventaja,

la lentitud en la transferencia de datos es su principal desventaja. Por todo esto, en la mayoría de las aplicaciones de comunicación de datos, se prefiere la transmisión en serie a la transmisión en paralelo.

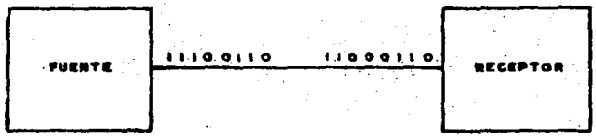


FIGURA 3.8 TRANSMISION SERIE

3.4.2. TRANSMISION ASINCRONA Y SINCRONA

En la tranemisión serie se preveen dos formas distintas de mensaje a enviar: Transmisión Asincrona y Sincrona.

En la modalidad asincrona la distancia (en términos de tiempo) entre un signo a enviar y otro es arbitraria, mientras que la distancia entre los bits de cada signo es fija. En la fig. 3.9. se esquematiza la transmisión de los signos "F" y "H" en modalidades asincrona.

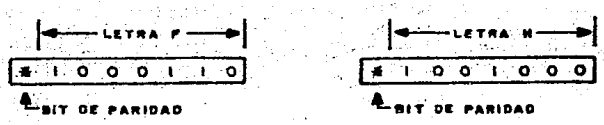
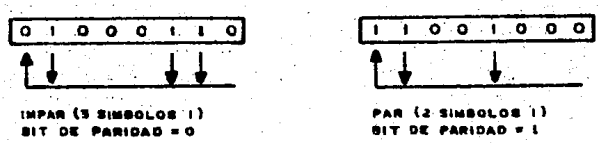


FIGURA 3.9 TRANSMISION ASINCRONA

En la fig. 3.9 el bit de paridad, es un elemento de control sobre la fiabilidad del mensaje recibido. Este control se efectúa igualando a "1" el bit de paridad (bit 8) si en los otros bits hay un número par de símbolos "1"; viceversa, si hay un número impar de símbolos "1", en el signo a enviar, el bit de paridad se iguala a "0". En el ejemplo anterior, el caracter F (1000110) contiene tres "1", el caracter H (1001000) contiene dos "1", y el bit de paridad toma el valor "1". La forma del mensaje será, por lo tanto la siguiente:



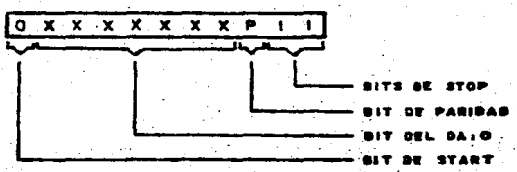
En la práctica nos podemos encontrar con la lógica inversa, es decir:

bit de paridad= 1 en caso impar

bit de paridad= Ø en caso par

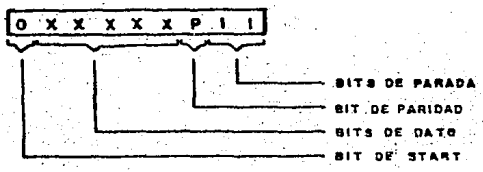
Esta inversión es de índole meramente formal. Además del bit de paridad, en la transmisión asincrona hay que enviar una señal de reconocimiento de comienzo de mensaje y una señal de final de mensaje. La señal de comienzo se denomina bit de START; la de final, bit de STOP. Normalmente los mensajes se concluyen con dos bits de STOP.

El formato completo es, por lo tanto, el siguiente:



El bit de START vale Ø y los bits de STOP valen 1. El estado 1 se llama MARK, y el estado Ø, SPACE.

La transmisión de información se realiza en base a datos de 8 bits, pero puede ser de 5,6,7 y 8 bits. Por ejemplo, si el protocolo estipula que habrán solo 5 bits de datos asincronos en cada palabra asincrona transmitida, entonces el dispositivo receptor sólo recibira 5 bits de datos e interpretará los restantes de cada palabra recibida como sigue:



Entonces en realidad se está transmitiendo una unidad de 9 bits. El bit de paridad está siempre presente ya sea para especificar paridad impar ó paridad par.

Si se tienen 2 bits de parada, entonces cada palabra de 8 bits de dato serie contendrá 12 bits. Si se tiene 1 bit de parada, entonces cada palabra de datos de 8 bits serie contendrá 11 bits. Algunos protocolos de transmisión especifican uno y medio bits de parada.

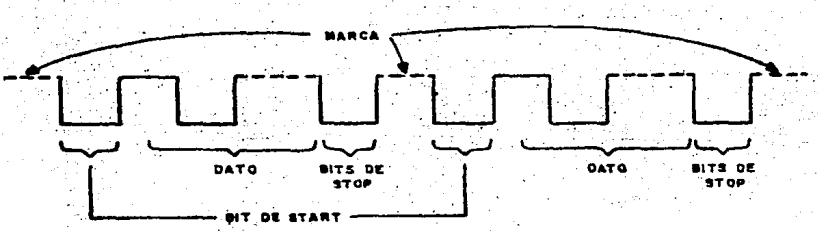


FIGURA 3.10 SEÑAL DE "MARCA"

En la transferencia de datos serie en modo asincrona, el dispositivo transmisor enviará una señal conocida como "marca" (usualmente de nivel alto, fig. 3.10) mientras no

tenga un dato a transmitir.

La caracteristica principal de la transferencia de datos serie sincrona, es que los datos se conforman exactamente a una señal de reloj.

En el protocolo sincrono se debe definir la longitud de las unidades de datos y se debe proporcionar al dispositivo receptor con alguna forma de sincronizar los extremos de la unidad de datos. El caracter SYNC es una palabra fija previamente definida. Cada flujo de datos sincronos comienza ya sea con 1 ó 2 caracteres SYNC.

ØIIØIØØI	ØIIØIØØI	IØIIØI
SYNC 1	SYNC 2	Primer caracter de la corriente de datos.

La unidad de datos en un flujo de datos serie sincronos usualmente consiste de bits de datos sin paridad; pero puede tener 1 bit de paridad.

La transmisión de datos sincrona requiere que el dispositivo transmisor mande los datos continuamente. Si el dispositivo transmisor, no tiene datos listos para mandar debe meter relleno con caracteres SYNC hasta que el próximo caracter real este listo para transmitir.

Para ilustrar este concepto consideremos a un operador metiendo datos en un teclado el mensaje siguiente:

LA$COMUNICACION$ES

Donde el caracter $ representa el caracter de espacio. Puesto que el operador estará metiendo datos a velocidad variable, la transmisión de datos serie de teclado insertará caracteres SYNC, entonces el mensaje que se transmite podría ser el siguiente:

LA$$$COM$$U$$NI$CACI$$ON$$ES$$

Donde el caracter $ representa a los caracteres SYNC. Cuando el dispositivo receptor decodifica el caracter SYNC en medio de un mensaje, ignorara el caracter, pero permanecerá en sincronización con el flujo de datos en serie, listo para interpretar el próximo caracter.

3.5 TIPOS DE LINEAS DE COMUNICACION DE DATOS

3.5.1 LINEA DE PUNTO A PUNTO

La línea de punto a punto mostrada en la fig. 3.11 es una componente fundamental de una red de comunicación.

FIGURA 3.11 LINEA PUNTO A PUNTO

La longitud de la línea puede ser desde 3 metros hasta 10 Km. La línea puede ser simplex, Half-duplex ó Full-duplex y puede operar en modo sincrono ó asincrono.

La configuración punto a punto es sencilla de realizar y tiene la ventaja de tener simpre la disponibilidad de línea; en cambio, tiene la desventaja de que es necesario emplear tantas líneas como periféricos tengan que conectarse, con el siguiente aumento de los costos, especialmente en el caso de comunicaciones de larga distancia.

3.5.2 LINEA MULTIPUNTO

En la linea multipunto se tienen dos ó más terminales conectadas a una línea de comunicación. La fig. 3.12 ilustra una configuracion utilizando líneas multipunto.

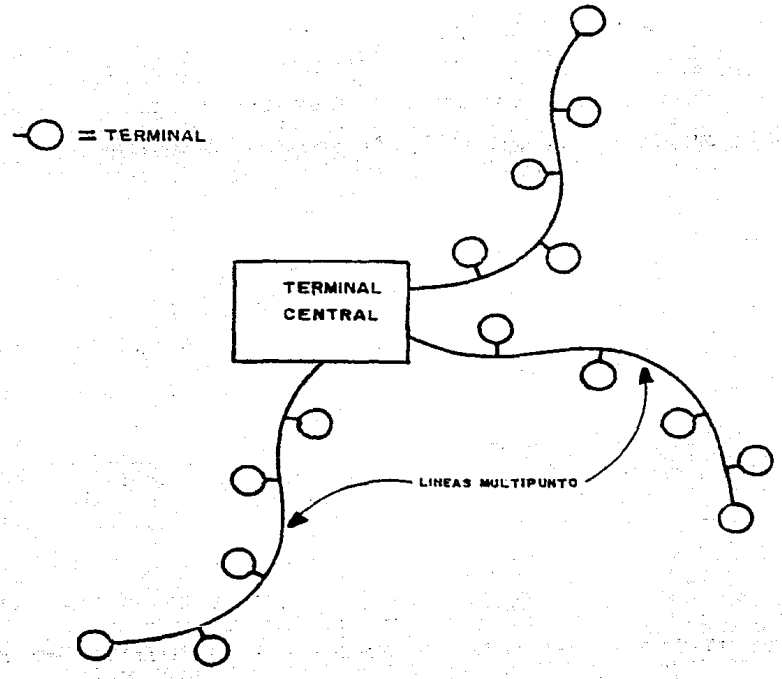


FIGURA 3.12 LINEA MULTIPUNTO

Las líneas multipunto son particularmente rentables para aplicaciones en donde no se necesite utilizar constantemente la línea. Además este tipo de líneas estan limitadas a un número de terminales de acuerdo a los siguientes factores:

- La capacidad de HARDWARE y SOFTWARE que involucre.
- Al congestionamiento de tráfico generado por las terminales (la longitud del mensaje y el tiempo en el cual sean generados los mensajes).
- La velocidad de la línea.

3.6 MULTIPLEXORES Y CONCENTRADORES

Otro método para incrementar la utilización efectiva de expansión de líneas de comunicación es el uso de concentradores y multiplexores.

3.6.1 MULTIPLEXORES

Un multiplexor es un dispositivo transparente que divide la capacidad de una línea de comunicación entre un número de terminales.

Los dos aprovechamientos básicos de los multiplexores son:

a) Multiplexado por división de tiempo
b) multiplexado por división de frecuencia.

MULTIPLEXADO POR DIVISION DE TIEMPO (MDT)

El MDT es un sistema que concede un intervalo fijo de tiempo a cada terminal que debe de transmitir y reproduce con el mismo tiempo las informaciones al otro extremo de la línea. Describimos los principios del MDT a través del siguiente ejemplo. Supongase que una organización tiene una computadora en una ciudad y cuatro oficinas sucursales en otra ciudad. En la organización se desea instalar una terminal con 300 bits por segundo, y en la que cada una de las cuatro sucursales de estas terminales puedan ser conectadas a la computadora mediante una línea de punto a punto, como se muestra en la fig. 3.13. Esta puede dar a cada terminal acceso sin restringir a la computadora.

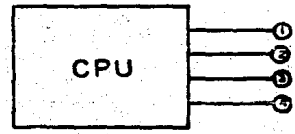


FIGURA 3.13 MDT

La fig. 3.14 muestra las cuatro terminales conectadas a la computadora por una línea de comunicación de alta velocidad y multiplexado. Este diagrama muestra el flujo de caracter de baja velocidad, donde cada terminal va dentro del

multiplexor B, el cual toma los caracteres de cada linea intercalandolos y transmitiendo estos a lo largo de un canal de alta velocidad a el multiplexor A, por lo cual entonces demultiplexa el flujo del caracter intercalado y reconstituye el flujo de datos de baja velocidad original.

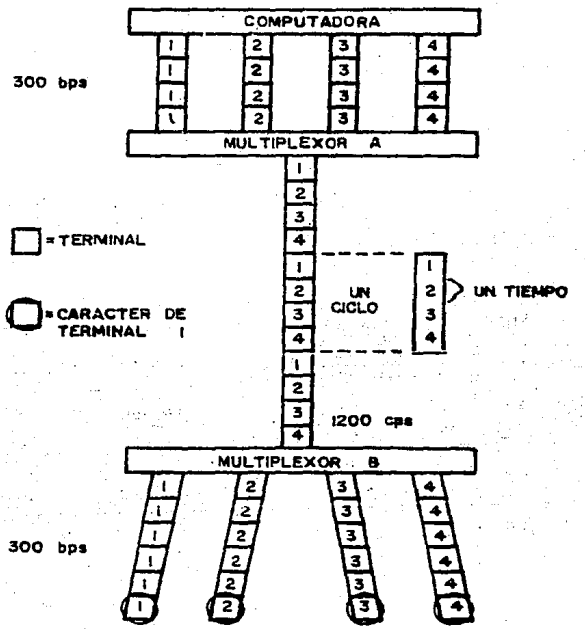


FIGURA 3.14 LINEA DE ALTA VELOCIDAD

Esta figura en particular muestra los caracteres que estan intercalados en la linea de alta velocidad, aunque en

realidad muchos multiplexores intercalan bits antes que los caracteres.

Los multiplexores disponibles comercialmente proporcionan un amplio rango de líneas de velocidades en el lado de baja velocidad, manipulando diferentes códigos y pueden muchas veces intervenir transmisiones sincronas y asincronas. En estos casos el flujo del bit en el circuito de alta velocidad puede ser totalmente complejo, este problema no es único, porque es demultiplexado el otro al final y reconstruido dentro del flujo de dato original.

MULTIPLEXADO POR DIVISION DE FRECUENCIA (MDF)

Con el MDF cada canal de datos correspondiente a cada terminal trabaja en una determinada banda de frecuencias. El principio es similar a la transmisión de radio: hay varias emisoras que transmiten simultáneamente en frecuencias diferentes, y un aparato de radio receptor solo capta la emisora que sintoniza, ver fig. 3.15.

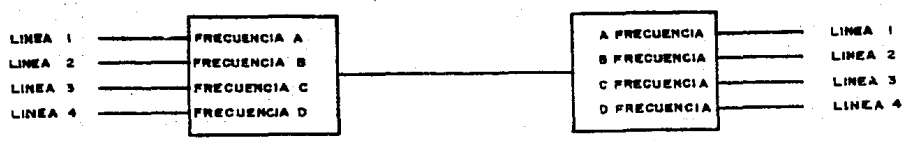


FIGURA 3.15 MDF

El MDT es el más común de el multiplexado en redes de computadoras. Es generalmente más eficiente que el MDF porque puede hacer mejor uso de la capacidad disponible de la línea de comunicación. La MDF requiere de "bandas de resguardo" entre los rangos de frecuencia que han sido asignado para cada terminal, y esto reduce la eficiencia de transmisión porque las "bandas de resguardo" utilizan un rango de frecuencia de la capacidad de la línea.

3.6.2 CONCENTRADORES

Un concentrador ó procesador de comunicaciones es una computadora basada en un dispositivo que muchas veces tiene alguna forma de almacenar cantidades juntas. Funcionalmente, el concentrador es similar a el multiplexor debido a que estos combinan los datos desde un número de terminales en una línea de alta velocidad para transmisión a una computadora patrón (Host).

Esto es, sin embargo, un dispositivo sotisficado porque puede alterar ó modificar la forma del flujo de datos anteriores para mezclarlo dentro de una línea de alta velocidad. La fig. 3.16 muestra como un concentrador puede interfasar una red de comunicación a una computadora. El concentrador mismo es usualmente conectado a la computadora host a través de una línea sincrona de alta velocidad, y la

red de la terminal es interfasada directamente dentro del concentrador. En la figura 3.16, un concentrador es interfasado a un rango de variedades de redes desde terminales asincronas de baja velocidad hasta medias velocidades sincronas, y lineas asincronas a alta velocidad de estaciones de entrada (job) remotas.

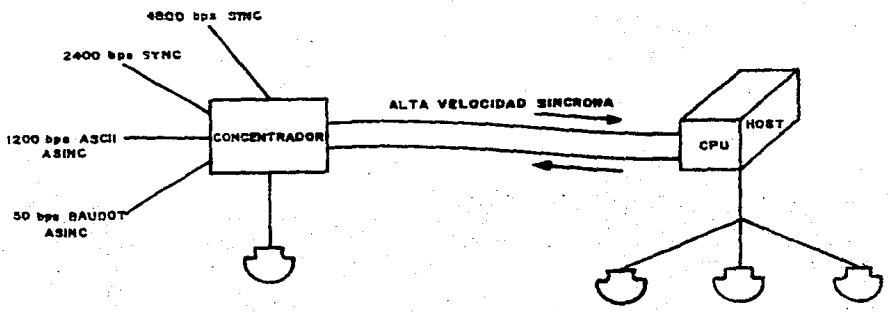


FIGURA 3.16 CONCENTRADOR

El concentrador puede ser equipado con bastante almacenamiento ó con una memoria principal, representando un almacenaje y un dispositivo avanzado en el que puede ensamblar un mensaje completo ó bloques de mensajes desde la red de las terminales, almacenando esta en la memoria y avanzando hasta el host.

En algunos casos, el concentrador es usado como núcleo recaudador de datos que acumula datos desde una red y

posteriormente avanza entre bloques de mensajes hacia el host.

Un caso especial del concentrador es el procesador frond-end. Una computadora frond-end manipula la red de comunicación en nombre (intercede por) el host y transmite los mensajes completos a el host. El enlace entre el host y el frond-end puede ser la línea de dato serie de alta velocidad ó puede ser a una computadora en paralelo (para interfase). Además el procesador frond-end puede trabajar con ó sin almacenaje auxiliar de memoria. Con el almacenaje auxiliar, una máquina puede recolectar mensajes, datos y guardarlos hasta que el host este preparado para procesar un grupo de dato. Similarmente puede recibir un grupo de datos de salida desde el host y distribuirlo dentro de las terminales a su propia velocidad y a su propio tiempo.

El enlace entre el procesador host y el concentrado ó frond-end es usualmente una línea punto a punto.

3.7 MODEMS

Un módem toma pulsos binarios que recibe de una computadora ó terminal y los convierte en una señal analogica continua que puede transmitirse por una línea de transmision de comunicaciones. Uno de esos métodos de transmisión conoce

como codificación por corrimiento de frecuencia (FSK) debido a que se reconoce un cambio en el valor binario transmitido mediante un cambio en la frecuencia. Los ceros y unos de la computadora se convierten a señales de tono de dos frecuencias distintas. Por ejemplo, si la señal portadora en la linea telefónica es un tono continuo de 1700 Hz y se requiere transmitir un 1, la portadora se corre arriba hasta 2200 Hz; para transmitir un Ø la portadora se corre hacia abajo hasta 1200 Hz. El módem en el extremo receptor detecta el corrimiento en la frecuencia y produce un Ø ó un 1 como resultado.

Los circuitos de comunicaciones se clasifican según la velocidad a la que pueden transmitir datos a través de ellos. Por ejemplo, por lo general los circuitos telefónicos de grado de voz se utilizan aún conjunto de velocidades fijas de transmisión que varian desde 1200 bps hasta 9600 bps. De hecho, la velocidad de transmisión de los datos en estos circuitos está determinada por la velocidad de modulación de módem. Para transmitir a 1200 bps, el módem debe convertir las señales binarias a señales analógicas a razón de 1200 bps; para transmitir a 9 600 bps, la mayoría de los módems utiliza formas especiales de codificación, como los dibits.

Un módem puede tener otras caracteristicas además de modular y desmodular datos. Los módems equipados con una

unidad auxiliar pueden realizar discado automático para llamar a terminales remotas. Se pueden configurar para que estén en estado de alerta continuo de manera que puedan ser llamados desde una terminal remota en cualquier momento; a esta característica se le denomina respuesta automática. Algunos modems se pueden utilizar para transmitir datos ó voz en forma alterna. Otros también permiten la transmisión simultánea de voz, lo que es útil para localizar y reparar fallas entre una computadora central y una instalación de terminal remota ó cuando se trata de sincronizar varias operaciones entre una computadora central y una estación de entrada remota de trabajo (RJE). Algunos módems también pueden operar como "canal en reversa", en la cual se puede lograr una forma limitada de transmisión dúplex utilizando circuitos de dos alambres. En esta configuración, mientras que el módem transmite datos en una dirección (usando la multiplexión de frecuencias), el caracter de respuesta que reconoce la recepción de un mensaje sin errores se envía simultáneamente en la dirección opuesta sobre la misma línea de transmisión, con lo que se elimina el tiempo de retorno para transmitir respuestas desde la estación receptora.

Los módems que operan a menos de 1800 bps por lo general se clasifican como de baja velocidad. Utilizan principalmente la técnica FSK. Los módems que operan desde 1800 hasta 9600 bps y más, generalmente se denomina de alta

velocidad. Por lo común emplean un tipo de modulación de fase y una metodología de transmisión que usa dibits. Los módems de velocidades altas generalmente se utilizan en terminales remotas de video y en estaciones de entrada remota de trabajo con muchos dispositivos de entrada/salida (teleimpresora, pantalla visual, lectura de tarjetas e impresora).

Un tipo especial de módem y de uso ocasional, es el que se emplea en la transmisión en paralelo. Este tipo de módem tiene tantos canales como números de bits tiene la clave del caracter que se va a transmitir. El módem en paralelo transmite un caracter a la vez y se utiliza para comunicación de datos entre dos CPUs. Si la velocidad de transmisión de un módem común fuera de 1200 bps y se cambiará esta disposición a un módem en paralelo, la velocidad aumentará hasta 1200 caracteres por segundo, lo que incrementaría sustancialmente la cantidad de datos transmitidos entre dos CPUs. La transmisión en paralelo no se utiliza por grandes distancias porque los bits tienen a derivar hacia adelante y hacia atrás en su relación mutua y pueden interferir entre sí, borrando algunos de los bits del caracter anterior ó del siguiente.

Otro tipo especial de módem es el acoplador acústico, el cual utiliza la codificación por corrimiento de

frecuencia. Es utilizado a menudo en la red telefónica pública debido a su bajo costo y conveniencia. Este tipo de módem se acopla acusticamente a la línea en lugar de hacerlo eléctricamente como lo hacen otros módems, es decir, las señales digitales de convierten a tonos acústicos que se producen frente a un micrófono de un teléfono ordinario. En la otra terminal un micrófono capta los tonos del auricular y los convierte a forma digital.

3.8 CONFIGURACIONES DE REDES

Todos los elementos para la transmisión de datos, como interfaces, líneas, modalidad de transmisión y protocolos, encuentran su síntesis en las redes de comunicación.

Una red esta compuesta en general por un elemento Hardware y por un elemento Sofware; el primero está constituido por los aparatos que permiten las comunicaciones, y el segundo organiza y gestiona la red.

Existen 4 configuraciones de redes básicas que pueden ser utilizadas:

- red estrella
- red anillo

- red malla
- red jerárquica

La red estrella, mostrada en la figura 3.17 es probablemente la más común de estas configuraciones. En esta red, cada terminal esta conectada a una central a través de una línea punto a punto. Las líneas multipunto pueden también usarse a lo largo con líneas punto a punto en configuración estrella.

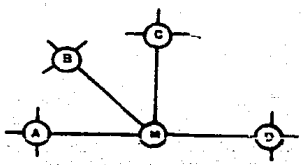


FIGURA 3.17 RED ESTRELLA

La fig. 3.18 ilustra una red anillo, esta configuración consiste de un número de computadoras conectadas todas en un lazo cerrado ó anillo. En este caso hay dos trayectorias que pueden ser establecidas entre cualquiera de las dos computadoras en la red, si por alguna razón una de las trayectorias falla, puede ser usada la otra ruta de reserva.

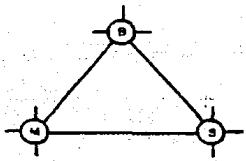


FIGURA 3.18 RED ANILLO

Si hay una necesidad de manipular largos volúmenes de tráfico desde muchas terminales en varias ciudades, puede ser ventajoso el empleo de una red de malla, como se muestra en la fig. 3.19.

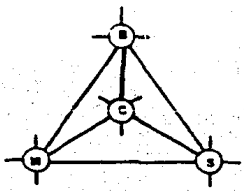


FIGURA 3.19 RED DE MALLA

La decisión para usar una red estrella, anillo ó malla, se basa principalmente sobre el costo de las líneas y la geografía de la red. Por ejemplo: en un pais como Austria, hay pocas ciudades grandes y estan separadas por miles de kilometros, la red más común es la configuración estrella. En paises que tengan varias ciudades grandes la

red más empleada son las de anillo ó de malla.

La red jerárquica es mostrada en la fig. 3.20, y puede ser expandida por sí misma. En estos tipos de configuración varios niveles de computadoras pueden ser interconectadas, como si fuera una corporación en desarrollo.

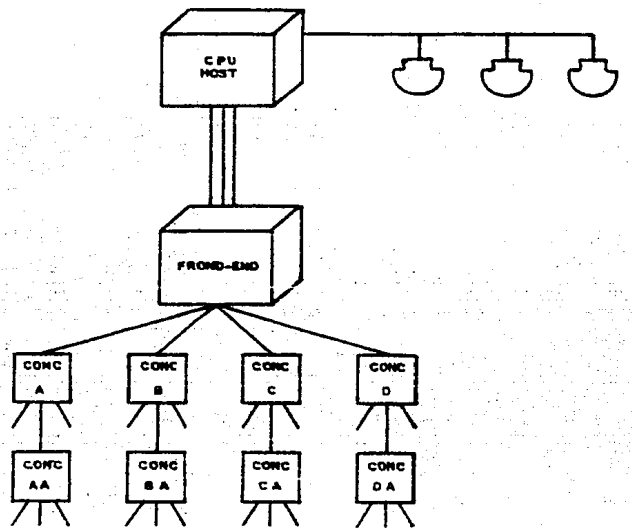


FIGURA 3.20 RED JERARQUICA

3.9 CLASES DE TERMINALES

Hay un amplio rango de terminales, el orden para clasificarlas son:

- Terminales simples
- Terminales sofisticadas
- Terminales inteligentes

La línea de división entre esta es muy confusa, realmente una terminal en particular puede pertenecer a dos clasificaciones dependiendo de la configuración empleada.

Las terminales varían desde un mecanismo simple, hasta llegar a uno complejo. La principal ventaja de las terminales simples es su costo. Sin embargo, las técnicas electrónicas modernas estan sustancialmente reduciendo la manufacturación de los costos de terminales. La tendencia en la industria parece ir hacia arriba produciendo más terminales inteligentes, las cuales hacen posible hacer más procesamiento.

Las características generales de las terminales simples, sofisticadas e inteligentes son resumidas en la siguiente tabla:

CARACTERISTICA GENERAL DE LAS TERMINALES

1) TERMINALES SIMPLES

+ económica
+ no amortiguadas
+ conducción libre
+ asincrona
+ baja velocidad
+ poca (ó sin) capacidad de detección de error
+ poco (ó sin) almacenaje (cinta de papel)

2) TERMINALES SOFISTICADAS

+ más expansivas
+ no amortiguadas
+ sincrona/asincrona
+ alta velocidad
+ alto nivel en el procedimiento de lineas de control
+ detección y corrección automático de error
+ almacenaje (cassette)
+ operación de agrupamiento
+ operación de multipunto
+ seguridad (distingue lectura)
+ mecanismo auxiliares (impresoras, cassettes)..

3) TERMINALES INTELIGENTES

Tiene las mismas caracteristicas de una terminal sofisticada, pero además incluye:

+ capacidad de computación
+ almacenaje en línea
+ editor de datos
+ compresión de datos
+ almacenaje local de "formas y mensaje listo para operar.

CAPITULO IV

TECNICAS DE DETECCION DE ERROR

4.1 ORIGEN DE LOS ERRORES

Todos hemos tenido una conversación telefónica en la que nuestra voz ha tenido que competir con chasquidos y chisporroteos, ó aún con otra voz cuando se cruzan las líneas. Afortunadamente el cerebro humano es una muy buena computadora que puede adaptarse a las condiciones de la línea, minimizando los efectos de ruido. Por ejemplo, si un chasquido borra una palabra de la conversación, generalmente nosotros podemos remplazar la palabra sin necesidad de pedir a la otra persona que la repita. Si las condiciones de la línea empeoran, podemos pedir a la otra persona que hable más despacio ó más claro. Cada una de las anteriores acciones es un ejemplo de transmisión y recepción con modificaciones que tienen como objeto adaptarse a las condiciones imperantes en la línea telefónica.

En comunicación de datos el ruido de las líneas puede destruir los bits, convertir un bit "1" en un bit "0" y viceversa. Además del fondo constante de ruido térmico existen algunos impulsos ocasionales de ruido de gran magnitud que se deben, a equipo de conexión mal protegido,

inducción de los cables eléctricos, estática atmosférica y una gran variedad de diversos factores.

4.1.1. FENOMENOS ADVERSOS DE LAS LINEAS DE TRANSMISION

Los fenómenos adversos ó imperfecciones de las líneas de transmisión son aquellos que por razones intrínsecas ó extrínsecas a la propia línea de transmisión, nos alejan del objetivo ideal: reproducir en el extremo receptor con absoluta fidelidad la señal de origen. A continuación se detallan cada uno de estos fénomenos.

(1) PERDIDAS DE INSERCION O ATENUACION

Las señales eléctricas al propagarse por una línea, sufren pérdidas de potencia que se evalúa mediante la siguiente relación logarítmica:

$$N(dB) = 10 \log_{10} \frac{\text{Potencia enviada}}{\text{Potencia recibida}}$$

Para medir la pérdida de inserción de una determinada línea se elige una frecuencia llamada de referencia, que normalmente es de 800 Hz. Cuando las líneas están constituídas por conductores físicos, la atenuación que se tendrá será la misma en ambos sentidos, lo cual no es

verdadero cuando intervienen elementos amplificadores ó sistemas multiplex en general, en cuyo caso se hace necesario hacer ciertos ajustes para lograr igual atenuación en ambos sentidos.

(2) DISTORSION DE ATENUACION

Según la serie de Fourier: toda señal compleja es la superposición de una serie en teoría infinita, de frecuencias puras.

A causa de su propia constitución interna, las líneas de transmisión presentan una atenuación distinta a cada frecuencia, lo que da como consecuencia que la señal reproducida en recepción no corresponda exactamente con la original, por haberse alterado no solo los valores absolutos sino también los relativos de sus frecuencias puras que la componen.

La distorsión de atenuación es el grado en que la señal recibida deja de ser similar a la señal enviada, y se caracteriza mediante la respuesta atenuación/frecuencia. Esto es, representando en un plano, los valores de la atenuación, que presenta la línea a una serie de frecuencias dentro de la gama a transmitir, ó lo que es lo mismo, la diferencia de estos valores respecto al obtenido a 800 Hz.

(3) DISTORSION DE RETARDO DE GRUPO

Las líneas de transmisión tienen un tiempo de propagación que varia dependiendo de la frecuencia, de manera semejante a lo que sucede con la atenuación. Las distintas frecuencias puras que componen una señal compleja, se propagan a velocidades distintas. Lo anterior da resultado una distorsión en la señal recibida que se conoce como distorsión de retardo de grupo ó distorsión de fase, y se cuantifica por la respuesta retardo de grupo/frecuencia tomando como referencia (retardo cero) la frecuencia que se propaga a mayor velocidad, cuyo valor depende de la propia constitución de la línea.

En una conversación telefónica, este fenómeno carece de importancia, sin embargo, en la transmisión de datos su efecto es notorio sobre todo cuando se transmite a altas velocidades.

(4) RUIDO ALEATORIO O BLANCO

El ruido aleatorio ó ruido blanco que siempre está presente en toda línea de transmisión, es un factor determinante de la velocidd máxima que puede alcanzarse en un determinado circuito destinado a transmisión de datos, por lo

que siempre se debe tomar en cuenta al fijar la calidad del circuito.

La cuantificación de un nivel de ruido se realiza expresandolo en dBmØp, que representa la relación, expresada en decibelios, entre la potencia del ruido y la de una señal de prueba, en un punto particular del circuito conocido como nivel relativo cero (1 mW normalmente). De lo anterior se puede deducir el valor absoluto de la potencia de ruido. Con frecuencia se utiliza el concepto de relación señal/ruido, indicando el número de decibelios que separan a ambas potencias en un punto dado.

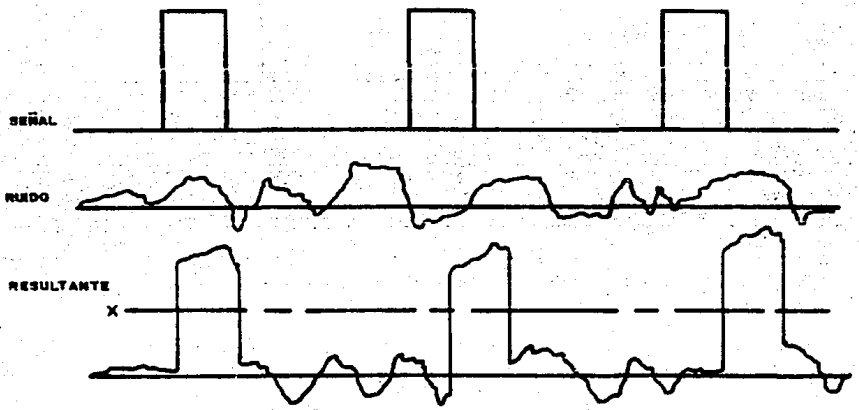


FIGURA 4.1 EFECTO DEL RUIDO BLANCO

(5) RUIDO IMPULSIVO

El ruido impulsivo se puede definir como picos de ruido de muy corta duración y elevado nivel, el cual tiene una incidencia fundamental en la transmisión de datos, ya que contribuye en forma sustancial a configurar la frecuencia y distribución de errores en la linea (éste es un dato bàsico para el diseño de un sistema completo de transmision de datos).

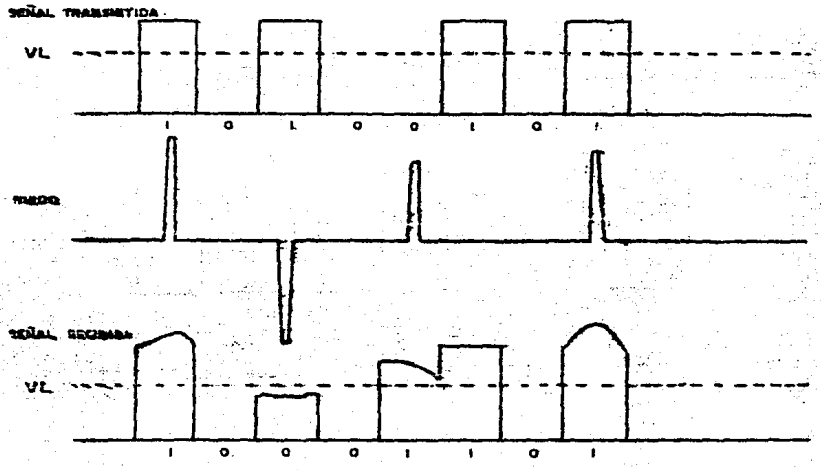


FIGURA 4.2 EFECTO DEL RUIDO IMPULSIVO

Este ruido se origina básicamente en las fuertes inducciones que se producen sobre una linea, consecuencia de conmutaciones electromecanicas de cualquier tipo que se

realice en sus inmediaciones.

La medida se realiza contando el número de veces, en un determinado espacio de tiempo, que los picos sobrepasan un nivel prefijado que se conoce como umbral.

(6) RUIDO DE CUANTIFICACION

Se presenta únicamente cuando en algún tramo de la línea esta constituído por sistemas MDT Se origina en el error que inevitablemente se comete tanto en el muestreo como en la cuantificación de la señal, que se realizan en tales sistemas.

Los efectos de ese error se presentan en la misma forma que el ruido de intermodulación, de desadaptación compresor-expansor, fluctuación de fase, etc. confundiendose con ellos y dificultando su medida.

(7) DESVIACION DE FRECUENCIA

Cuando en un circuito intervienen canales de sistemas multiplex tipo MDF, se producen modulaciones y demodulaciones en las que si los osciladores que generan las portadoras no son rigurosamente idénticas, dá lugar a pequeñas diferencias entre la señal recibida y la transmitida. El CCITT (Comite

Consultivo Internacional de Telefonía y Telegrafía) recomienda que esta desviación no supere los 2 Hz.

(8) FLUCTUACION DE FASE

La fluctuación de fase consiste en el desplazamiento del paso por cero de una señal, con respecto a los instántes previstos. Son variaciones lentas y casi permanentes de la fase de la señal, que se aprecian en el osciloscopio como un temblor de la imagen que por la persistencia de la misma en la pantalla se convierte en un trazo muy grueso.

Las principales causas de este fenómeno son:

Rizado de alimentación; Por efecto de inducciones en general, la corriente de red (60 Hz) y sus primeros armónicos (120, 180 Hz) modifican la fase de los osciladores y producen en consecuencia fluctuaciones.

- Insertabilidad en la frecuencia de la red.
- Interferencia de corrientes de llamada.
- Variación de corrientes de llamada.
- Variación de carga de los osciladores.

(9) ECO

El eco se puede definir como una señal de las mismas características que la original. El efecto nocivo del eco es mayor cuando la señal perturbadora es menor atenuada y más retardada.

Para que las señales reflejadas se reciban con retraso apreciable, deben de recorrer grandes distancias, por lo que este fenómeno sólo se presenta en comunicaciones intercontinentales vía satélite, etc. En estos casos, los circuitos telefónicos van dotados de elementos supresores de eco que impiden la transmisión simultánea en ambos sentidos, elementos que se inhiben cuando el circuito se usa para transmisión de datos.

(10) DIAFONIA

La diafonia, ocurre cuando una línea toma parte de la señal que va por otra línea. Este fenómeno se presenta entre pares de líneas que llevan señales separadas, en líneas multiplex que llevan muchas señales discretas, en enlaces de microondas donde una antena recoje una pequeña porción de la señal reflejada de otra antena en la misma torre y en cualquiera de los circuitos telefónicos de alambre fijo que corren paralelos unos a otros y que esten demasiado próximos

entre sí y no estén eléctricamente balanceados. La Diafonía aumenta con:

- mayor distancia en la comunicación
- mayor próximidad de las líneas físicas.
- mayor intensidad de señal
- señales de frecuencia más alta.

Un tipo especial de diafonía es llamado ruido de intermodulación. Este fenómeno se presenta cuando dos líneas independientes se intermodulan y forman un producto que cae dentro de una banda de frecuencias que difiere de ambas entradas. La frecuencia resultante puede caer dentro de una banda de frecuencias reservada para otra señal. Este tipo de ruido es semejante a los armónicos de la música. En una línea multiplex, muchas señales distintas se amplifican juntas y ligeras variaciones en el ajuste del equipo puede provocar ruido de intermodulación. Un módem mal ajustado puede transmitir un tono de frecuencia intenso cuando no está transmitiendo datos, produciendo así este tipo de ruido.

(ii) AUMENTOS Y DISMINUCIONES EN EL NIVEL DE LA SEÑAL

Cuando se está empleando un enlace a través del circuito de microondas son más ó menos frecuentes (dependiendo de los lugares geográficos por donde se

encuentre la ruta de estos circuitos) los cambios también más ó menos bruscos de los niveles de señal entre repetidoras. Esto se reflejará indudablemente en cambio de nivel de señal del usuario. Tales cambios se deben fundamentalmente a apariciones de cortinas de lluvia ó nieve (ó su desaparición), entre estaciones de microondas, lo que produce fuertes cambios en las condiciones del medio de propagación. Aunque existen grupos de especialistas que vigilán estos cambios para corregirlos variando la ganancia de los equipos e incluso existen correctores automáticos los cuales no tienen una reacción instantánea. El efecto se refleja en el receptor del usuario, aunque sea en un periodo corto y así se producen pérdidas, en este caso de datos, lo que origina retransmisiones.

Para ampliar este concepto, supongamos que un módem está ajustado para recibir su señal a -20 db y de momento se inicia una tormenta entre dos repetidoras por donde pasa el enlace utilizado y la señal baja a -40 db. Mientras se modifican las ganancias, manual ó automáticamente, la señal se pierde, el módem no detecta ésta y provoca la acción correctiva.

Ahora en el caso contrario, cuando los equipos estan ajustados para operar durante una nevada y en un momento ésta desaparece, si la señal que se estaba recibiendo era de -20 db

y con el cambio pasa a ser Ø db ó cerca de cero. Esto saturará al módem y lo descontrolará, con el mismo resultado que en el caso anterior.

Lo anterior nos permite entender por qué, dependiendo de la localización geográfica, este fenómeno se presentará con más o menos frecuencia. También se entenderá que todos estos fenómenos, ruido, distorsiones, etc., son características del medio. Aunque es necesario convivir con ellos, tomandolos en consideración en los diseños de las redes y establecer procedimientos de diagnóstico durante la administración de las redes para vigilar su comportamiento y que no salgan de los rangos previstos.

4.2 RAZON DE OCURRENCIA DE ERRORES

La proporción con que se presentan los errores en un sistema de transmisión varía con la velocidad de transmisión. Esto se debe a que el intervalo que dura un impulso de ruido puede afectar más ó menos bits dependiendo de la velocidad a la que se este transmitiendo. Por ejemplo, si se esta transmitiendo a razón de 5Ø bps, un impulso de ruido que dura 2 milesegundos sólo estaría presente durante una décima parte de la transmisión de un bit con lo cual no adulterará ningún bit. Si se está transmitiendo a 1ØØØ bps, el mismo impulso de ruido corrompería dos bits. Por último, si se esta

transmitiendo a razón de 10 000 bps el impulso de ruido afectaría a 20 bits.

Es difícil obtener la proporción de errores de las líneas de comunicación ya que esta nada tiene que ver con los impulsos de ruido instantaneos, y aleatorios, mientras se permanece usando su red para la comunicación de datos. La tabla 4.1 presenta una muestra de la proporción de errores que podemos esperar si seleccionamos al azar una línea telefónica promedio, y transmitimos datos a través de ella a diferentes velocidades.

Tabla 4.1 Estadisticas de error tipicas para una línea telefónica seleccionada al azar.

Velocidad de Transmisión	Proporción de bits de error
1200 bps	1 en 200 000
2400 bps	1 en 100 000
9600 bps	1 en 1000 hasta 1 en 10 000

Proporciones de error superior a uno en 100 000 ó uno en 200 000 son generalmente inaceptables en la transmisión de datos. Sin embargo se puede transmitir por líneas con una

alta proporción de errores si su velocidad se limita a 1200 bps.

Por lo general, los errores aparecen en ráfagas. En un error en ráfaga, se cambia más de un bit de datos debido a la condición que provoca el error, esto trae como consecuencia que los errores de un bit no estén distribuidos uniformemente. Sin embargo las compañías de comunicación generalmente listan sus tasas de error como el número de bits erróneos dividido entre el número de bits transmitidos, sin hacer referenia a su distribución no uniforme.

Existen ventajas y desventajas del hecho de que los errores tiendan a aglomerarse en ráfagas en lugar de estar dispersos uniformemente. Si los errores estuvieran distribuidos uniformemente durante el dia, con una tasa de error de 1 bit en 500 000 sería raro que ocurrierán dos bits erróneos en el mismo caracter, por lo que sería sencillo idear un plan de comprobación de caracteres. Sin embargo, los errores casi siempre se presentan en ráfagas, durante períodos que pueden llegar a alterar 50, 100 ó más bits. El aspecto positivo es que, entre ráfagas puede haber periodos relativamente largos de transmisión sin errores.

4.3 ATENUACION DE LOS FENOMENOS ADVERSOS DE LAS LINEAS DE TRANSMISION

4.3.1 ACONDICIONAMIENTO DE LA LINEA

El acondicionamiento es un servicio que sólo se encuentra disponible en líneas arrendadas privadas y consiste en un balanceo eléctrico especial del circuito para asegurar transmisiones lo más libre de errores posible. Cuando se renta una línea que tiene que estar permanentemente conectada a nuestras terminales y computadoras, es posible para la compañía de comunicaciones (ó portadora común), eludir el equipo de conmutación en la central telefónica, con lo que se reduce subtancialmente la cantidad de ruido en la línea. Es decir, la compañía de comunicaciones puede medir el funcionamiento de la línea seleccionada y sumar componentes eléctricas que alteren sus caracteristicas.

El acondicionamiento nos permite usar la línea a velocidades tan altas como 9600 bps, con una aceptable cantidad de errores. El proceso de acondicionamiento sólo puede ser aplicado en líneas arrendadas, ya que en el proceso de selección aleatorio, que lleva implicito el uso de la red telefónica pública, nunca se conoce cuales componentes de la red en particular serán enlazadas al establecer una trayectoria de comunicación, de tal forma que nunca será

posible aplicar el acondicionamiento en una red telefónica pública.

4.3.2 COMPENSACION

Actualmente se pueden encontrar módems que son capaces de monitorear la línea e introducir señales que automáticamente contrarrestan ciertas alteraciones del comportamiento normal. Este proceso es conocido como compenzación y en la actualidad es un proceso automático. La compensación no es capaz de producir los mismos resultados que el acondicionamiento, pero puede lograr una transmisión de 4800 bps, que es poco factible en una red telefónica conmutada.

La compensanción es frecuentemente usada en conjunto con el acondicionamiento de línea, en lineas multipunto. Los módems receptores en una línea multipunto, ven las condiciones de la línea en forma diferente dependiendo de cual de los módems le está enviando información. El proceso de acondicionamiento de la línea en una línea multipunto puede elevar la calidad de la misma sobre un standard particular; de esta manera el proceso de compenzación adaptivo en el módem receptor puede cambiar de acuerdo a las condiciones de las líneas transmisoras.

4.4 DETECCION DE ERRORES

El sistema telegráfico y el télex generalmente dependen de la capacidad del cerebro humano para interpretar la idea de una palabra cuando se encuentra mutilada. Generalmente una cantidad considerable de números es a menudo repetida al final del telegrama, a causa de que la información numérica no puede ser interpretada intuitivamente.

La detección de errores en un sistema de comunicación de datos, lleva consigo el uso de redundancia. Es decir, la suma de información extra a la requerida para transmitir el texto.

Se puede desarrollar una metodología de transmisión de datos que proporcione un alto rendimiento de detección y corrección de errores. La única manera de detectar y corregir, es enviar datos adicionales con el mensaje. A mayor cantidad de datos adicionales puede lograrse mayor protección contra los errores, pero al elevar la protección, se reduce el procesamiento total de datos útiles. Esto nos lleva a la conclución de que la eficiencia del procesamiento total de datos varía inversamente con la mayor detección y corrección de errores. Incluso, al usar transmisión síncrona, los errores afectan la longitud del bloque de datos

que desea transmitirse. Con bloques de mensaje pequeños, hay menos probabilidad de retransmitir el bloque, pero a la vez existe una menor eficiencia en la metodologia de transmisión por lo que respecta al procesamiento total. Por otro lado, con bloques de mensajes largos, se tiene una mayor proporción de errores, los cuales tendrán que ser reenviados.

La tasa de errores en las transmisiones por la red telefónica pública, sufre una considerable variación de una a otra hora del día, lo normal es que la tasa de errores sea más alta durante los periodos de mayor tránsito. En algunos casos, la única alternativa que tiene el usuario de estas instalaciones, es transmitir los datos a menor velocidad, porque como se explicó anteriormente, a velocidades de transmisión altas se tiene mayor tendencia a los errores.

4.4.1 TECNICA DEL ECO

La técnica del eco es una forma simple de detección de errores, a menudo usada en situaciones interactivas, como sucede cuando se esta introduciendo información dentro de una computadora, usando una teleimpresora ó una terminal con pantalla.

En este tipo de técnica, el proceso que se sigue es el siguiente: El operador piensa en el caracter que desea

introducir y lo teclea en la terminal, la terminal transmite el caracter a la computadora, donde es recibido y almacenado en una memoria en masa, la computadora transmite (ó refleja) el caracter hacia la terminal. Cuando el caracter es recibido por la terminal, es desplegado y entonces el operador puede ver si el caracter que tecleó es el mismo que el desplegado en la pantalla. Si es incorrecto, se puede transmitir el caracter por segunda ocasión. Este método desperdicia la capacidad de transmisión porque cada mensaje (fraccionado) se transmite al menos dos veces y no hay garantia de que algunos mensajes sean transmitidos tres ó cuatro veces. Además, puede suceder que parte de esta retransmisión por segunda ó tercera vez podria ser innecesaria ya que el error pudo haber ocurrido en el viaje de retorno del caracter. Un caso que es poco probable que suceda, es que un caracter que se ha introducido corrompido a la computadora, sea reflejado y por efecto del mismo ruido en la línea, sea desplegado como el caracter que el operador tecleó.

La prueba del eco generalmente se emplea en líneas cortas, alambradas permanentemente con terminales de baja velocidad. Este tipo de técnica da cierto grado de protección, pero no tiene la eficiencia de otros métodos.

4.4.2 TECNICAS DE DETECCION DE ERROR AUTOMATICAS

En los sistemas de computadoras modernos es deseable hacer la detección y coorrección de errores tan automáticos como sea posible. Esto minimiza la intervención de operador y perfecciona el funcionamiento del sistema, al eliminar los tiempos de reacción relativamente largos inherentes a la intervención humana.

Hay un enfoque general para la detección automática de errores que es la base de dos métodos comunmente usados. En la fig. 4.3 se muestra este enfoque general. En este enfoque, ó proposición, los datos son puestos a través de un proceso matemático (ó algoritmo), que produce una secuencia de cuadros de comprobación FCS (frame check sequence). En el transmisor los datos son enviados seguidos por su FCS y en el receptor los datos son sometidos al mismo algoritmo al que fueron sometidos en el transmisor para producir una FCS calculada. Cuando llega la FCS que mandó el transmisor se compara con la FCS calculada en el receptor. En el caso de que ambas FCS sean iguales, el dato es declarado válido, en caso contrario se le considera erróneo y una acción correctiva es tomada.

En esta técnica el problema consiste en adoptar un algoritmo de modo que sea muy poco probable que un dato

mutilado pueda pasar la prueba de la FCS. A continuación se describirán dos proposiciones que se derivan del enfoque general: comprobación paridad doble coordinada y la comprobación de redundancia ciclica.

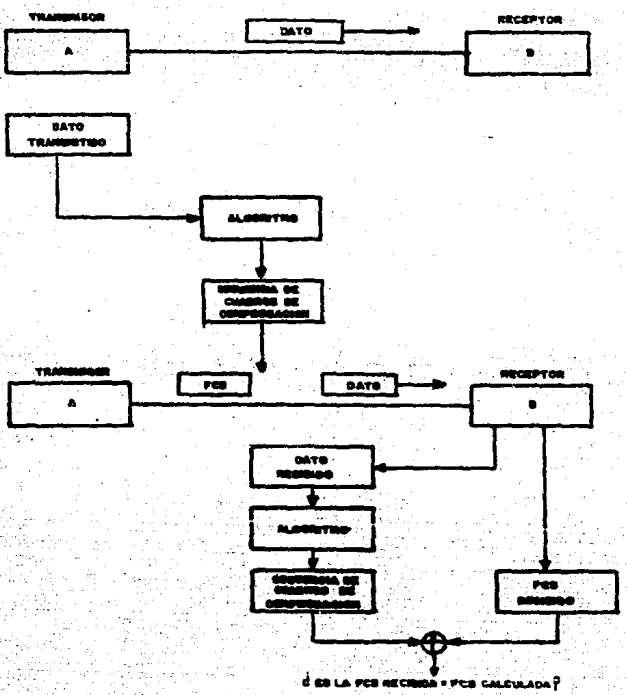
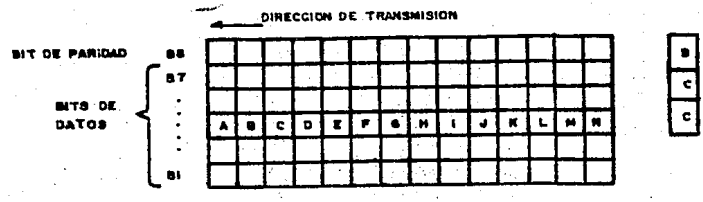


FIGURA 4.3 DETECCION AUTOMATICA

(1) Comprobación de paridad doble coordinada. Cuando transmitimos datos en bloques desde una computadora ó desde

una terminal con amortiguamiento, podemos aumentar la fuerza del bit de paridad agregando un caracter comprobador de bloque, ó BCC (block check character), al final del bloque de datos. En la fig. 4.4 se muestra un bloque de datos que esta siendo transmitido con un bit de paridad por cada caracter al final del bloque es anexado el caracter comprobador de bloque.



(a) Forma general de un bloque de datos con su BCC.

P	7	6	5	4	3	2	1	CARACTER
1	1	0	0	0	0	0	1	1
1	1	0	0	0	0	1	0	2
0	1	0	0	0	0	1	1	3
1	1	0	0	0	1	0	0	4
0	1	0	0	0	1	0	1	5
0	1	0	0	0	1	1	0	6
1	1	0	0	0	1	1	1	7
1	1	0	0	1	0	0	0	8
0	1	0	0	1	0	0	1	9
0	1	0	0	1	0	1	0	10
1	1	0	0	1	0	1	1	11
1	1	0	0	1	1	0	0	12
0	1	0	0	1	1	0	1	13
1	1	0	0	1	1	1	0	14
1	1	0	0	1	1	1	1	15
1	0	0	0	1	1	1	1	BCC

(b) Ejemplo de un bloque de datos con su BCC.

FIGURA 4.4 COMPROBACION PARIDAD DOBLE COORDINADA

En la fig. 4.4 se tiene un bloque de 14 caracteres, cada uno con un bit de paridad y al final del bloque el caracter comprobador de bloque. El bit 8 del caracter comprobador de bloque corresponde al bit de paridad de cada uno de los 14 caracteres: el bit 7 del caracter comprobador de bloque corresponde al bit 7 de cada uno de los anteriores caracteres y asi sucesivamente. De esta manera cada uno de los bits del bloque tiene dos bits de paridad; uno en dirección horizontal, que es obtenido del bit de paridad, y otro en dirección vertical, proporcionado por el caracter checado de bloque.

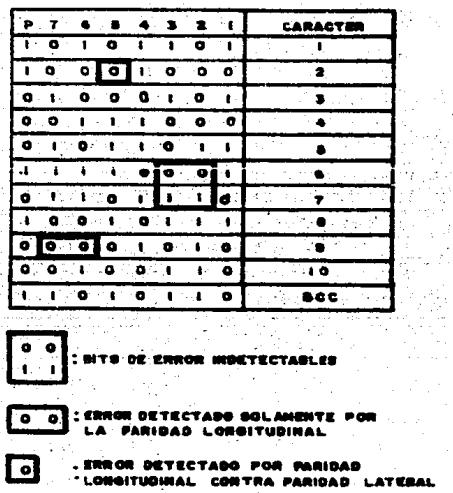


FIGURA 4.5 PARIDAD DOBLE COORDINADA

El bit de paridad de cada caracter es llamado a menudo bit de paridad horizontal, bit de paridad transversal, bit de paridad lateral ó bit de paridad de fila. Los bits de paridad del caracter comprobador de bloque, se les conoce como comprobador de paridad longitudinal, comprobación de paridad de columna ó comprobación de paridad vertical. Algunos autores emplean el termino caracter comprobador de redundancia logitudinal (LRCC) en lugar de caracter comprobador de bloque.

Al mandar el mensaje a la línea, el dispositivo transmisor agrega el caracter comprobador de bloque al final del flujo de datos. En el receptor se calcula nuevamente el caracter checador de bloque de acuerdo a los datos recibidos y se compara con el que fue enviado por el transmisor; en caso de ser iguales, el bloque es declarado válido. En la fig. 4.5 se muestra como si hay solamente un bit erróneo en el bloque, es posible determinar el bit incorrecto ya que tanto la paridad horizontal como la vertical serán incorrectas. Si hay dos errores en un mismo caracter, el bit de paridad horizontal no indicará ningún error, pero habrá dos bits de paridad vertical incorrectos, de tal manera que se detectará un error en el bloque pero no se sabrá cual caracter es el que contiene el error. De igual modo, si hay un error en la misma posición en dos caracteres, en los bits

de paridad horizontal se indicará el error y en el bit de paridad vertical el error no será detectado.

En este caso si es posible determinar cuales son los caracteres que contienen el bit erróneo. Si el error se presenta en dos bits que ocupen la misma posición en dos caracteres de un mismo bloque, no será posible que se detecte el error sin embargo la probabilidad de que esto suceda es muy pequeña. El cálculo de paridad horizontal y vertical actúa como una comprobación complementaria para incrementar la capacidad de detección de errores del sistema.

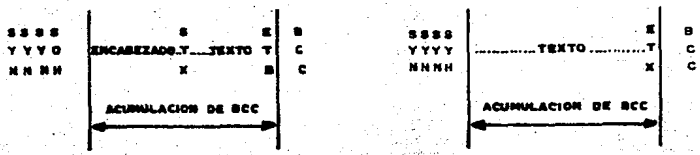


FIGURA 4.5 ACUMULACION DEL BCC

La comprobación de paridad doble coordinada es fácil de implementar tanto en hardware como en software. En la actualidad es más común encontrarla en hardware. El BCC consiste en la ejecución de una OR EXCLUSIVA en todos los caracteres precedentes. Las reglas para la acumulación del caracter de chequeo de bloque se muestran en la fig. 4.6, que muestra dos formatos de mensaje tipico empleados en los sistemas de transmición sincrona.

El transmisor genera un caracter comprobador de bloque de la siguiente manera: el cálculo del BCC se inicia cuando aparece SOH ó STX, este primer caracter no es incluido en el cálculo del BCC. En todos los caracteres que le siguen el sistema ejecuta una OR-EXCLUSIVA incluyendo ETB ó ETX, después de las cuales se transmite el BCC. En la terminal que recibe el receptor busca entre los datos para localizar lo que aparezca primero, ya sea SOH ó STX. Tan pronto como se recibe el caracter de arranque, el receptor comienza el cálculo de un BCC por medio de la ejecución de una OR-EXCLUSIVA en todos los caracteres que le siguen, incluyendo EETX ó ETB. Al recibir el caracter de fin de texto, el receptor tiene acumulado un BCC, y el siguiente caracter que recibe es el BCC calculado en el transmisor. En el caso de que ambos BCC sean iguales el dato es declarado válido, en caso contrario se indicará mediante una bandera de error.

Los caracteres de sincronia, SYN, pueden ser introducidos en el flujo de datos después de que se ha acumulado el BCC. En algunos sistemas los caracteres SYN se insertan como un tiempo de espera cuando esta incapacitado para tomar todos los caracteres que se encuentren en la línea con suficiente rapidez para mantener la sincronización entre los mismos. La mayoría de los sistemas elimina los

caracteres SYN del flujo de datos evitandoles el paso a través del programa del usuario.

(2) Comprobación de redundancia ciclica.- Existe un reciente concepto en la comprobación de la transmisión de datos, que se emplea cuando se tiene una comunicación sicrona. La comprobación de redundancia ciclica tiene mayor ventaja sobre la comprobación doble coordinada a velocidades altas, debido a que no requiere paridad en cada carácter. La comprobación de redundancia ciclica es básicamente una secuencia de dos caracteres transmitidos al final de cada bloque.

La comprobación doble coordinada es útil en los sistemas de transmisión de caracteres, pero involucra una gran cantidad de encabezados en forma de bits extra por caracter, que disminuyen las posibilidades de la linea. La tendencia actual es la transmisión binaria pura en la cual no necesariamente se rompe el flujo de datos en caracteres individuales.

La comprobación de redundancia ciclica se ha aplicado con más frecuencia gracias a los avances en materia de hardware. Consiste en tomar los datos que van a ser transmitidos como un gran número binario (sin importar si se trata de una fila de carácteres ó un flujo de bits binarios

puros), que puede ser dividido por otro número binario llamado constante. La división es de módulo dos no una división aritmética normal. Del proceso de dividir el número binario se va a obtener un cociente y un residuo. Como se muestra en la fig. 4.7, en la transmisión primero se manda el dato seguido por el residuo. En el receptor se hace nuevamente la división, y el residuo recibido se compara con el calculado. Cuando un error es detectado se puede tomar una acción que lo corrija.

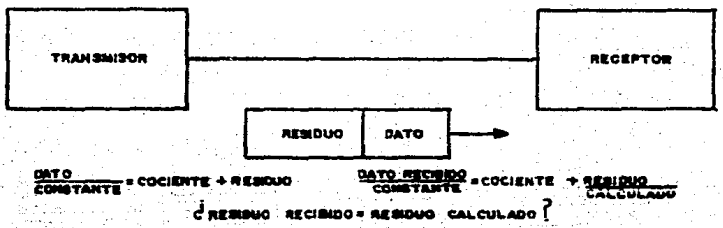


FIGURA 4.7 COMPROBACION DE REDUNDANCIA CICLICA

4.5 EFICIENCIA Y REDUNDANCIA

Una red de comunicación de datos tiene como objetivo lograr el máximo volumen de información exacta a través del sistema. A mayor volumen de información resulta mayor

eficiencia del sistema y menor costo. A la eficiencia del sistema le afectan características de la línea como la distorsión, velocidad de transmisión y tiempo de retorno en las líneas de HALFDUPLEX, el plan de codificación empleado, la velocidad del equipo transmisor y receptor, los métodos de detección y corrección de errores, así como los modos de transmisión. En esta sección se centra el interés en los dos últimos factores.

Se entenderá como eficiencia de transmisión la fracción de la capacidad teórica total del sistema que físicamente transfiere la información. Por tanto, la eficiencia se mide por medio de:

Eficiencia= Información/ tiempo unitario
Capacidad/ tiempo unitario.

Una clave ASCII de 7 bits, que utiliza el octavo bit para la paridad tiene

E= 7/8 = 0.875 x 100 = 87.5%

de eficiencia,que únicamente se refiere al plan de codificación. Dependiendo del método de transmisión, la eficiencia global caerá todavía más.

Un concepto complementario de la eficiencia es la redundancia. En general el término se refiere a los bits que no son parte de la información original. Los bits de parida,

bits de corrección de errores hacia adelante, y los bits de comrobación de redundancia ciclica sobre el bloque de datos son algunos ejemplos de redundancia. La redundancia se puede calcular usando la siguiente relación:

Redundancia= 1- Eficiencia

por ejemplo la clave ASCII de 7 bits tiene una redundancia de Ø.125

El diseñador del sistema tiene mayor interés en la eficiencia de todo el sistema, incluyendo la eficiencia combinada de la codificación y transmisión (que nos da la velocidad efectiva). En un sistema de transmisión, las pérdidas de la eficiencia se deben al plan de codificación y al método empleado para transmitir los datos. Por ejemplo, durante la transmisión sincrona la pérdida de eficiencia se debe al bit de paridad y una pérdida pequeña es consecuencia de los caracteres de sincronización transmitidos antes y después de cada bloque de datos. En la transmisión asincrona hay una pérdida de eficiencia por los bits de sincronización de arrancada y parada agregados a cada caracter antes de su transmisión. Por ejemplo, considere un plan de transmisión asincrona que envía un bits precedido y seguido por un bit de arrancada y uno de parada respectivamente. Entonces la eficiencia del plan de transmisión es de Ø.8, puesto que envía 1Ø bits para transmitir 8 bits de datos. El teletipo, que emplea un bit de arrancada y dos de parada tiene

una eficiecia: E= 8/11 =Ø.727.

Para determinar la eficiencia de los elementos combinados del sistema, el diseñador del sistema debe multiplicar la eficiencia del equipo transmisor:

Eficiencia (sistema total)= Eficiencia (plan de codificación) X Eficiencia (equipo de transmisión)

Utilizando el ejemplo anterior para la codificación ASCII, la eficiencia total del sistema será:

E= Ø.875 x Ø.8Ø = Ø.7Ø

Con la anterior eficiencia si se transmiten datos a 4 8ØØ bps, se tendrá una velocidad efectiva de 3,36Ø bps

Ve =4800x0.07=3360bps.

Este es un factor importante en el diseño de un sistema, ya que en ocasiones se tiene que lograr una determinada velocidad de transferencia de información mínima. Dependiendo de la eficiencia de procesamiento total de todo el sistema, el diseñador puede utilizar distintas estructuras de codificación para mejorar la eficiencia del plan de codificación y distintos modos de transmisión para mejorar la eficiencia del equipo transmisor.

En la práctica se requiere de muchos caracteres para formar un mensaje y la pérdida de uno puede destruir el significado de un mensaje transmitido. Además, también es importante incluir datos de la fuente y su destino en el mensaje para que el receptor pueda saber que no recibe un mensaje mal dirigido. Esta información extra también cuesta tiempo de transmisión. Por último, casi todo el equipo de transmisión necesita de determinados caracteres de control para señalar partes significativas de los mensajes transmitidos.

4.5.1 FACTORES DE BLOCAJE

Una consideración adicional debe plantearse cuando se transmite de una sola vez un bloque completo de datos, en transmisión síncrona. El intercambio de velocidad efectiva contra longitud de bloque, puede constituirse en un problema para el diseñador del sistema. Una longitud corta provoca una relación adversa de tiempo de transmisión al tiempo de retorno debido a las paradas excesivas cuando el dispositivo emisor transmite un bloque corto, el receptor reconoce la recepción correcta ó incorrecta de ese bloque y el transmisor envía otro bloque corto. Una longitud de bloque larga aumenta la probabilidad de error (ya que los errores ocurren en ráfagas) y puede requerir un tiempo excesivo de retransmisión. Por lo tanto, la longitud de bloque larga

aumenta de probabilidad de los errores y a la vez aumenta el tiempo de retransmisión.

Para comprender la eficiencia de la transmisión en bloques, calcularemos el tiempo que se necesita para enviar un bloque y para que el receptor lo reconozca utilizando una línea Halfdúplex. Supondremos que el bloque contiene N caracteres de datos y C caracteres de control, esto es, que la longitud del bloque es N+C. Si el tiempo de transmisión para un caracter es Tc, entonces el tiempo para transmitir todos los caracteres será Tc(N + C). Para reconcer la recepción del bloque se necesita un retraso de la terminal, TT y un tiempo de retorno de línea TL. Después del reconocimiento será necesario que transcurra otra vez TT y TL para prepararse para la nueva transmisión. Por lo tanto, el tiempo total por bloque es:

$$Tc(N + C) + 2(TT + TL)$$

Para calcular la velocidad de transmisión de la información tendremos qué: si hay b bits por caracter, entonces la velocidad de transmisión de información RI:

$$RI = \frac{B \cdot N}{Tc(N + C) + 2(TT + TL)} \text{ (bits/tiempo unitario)}$$

Entonces podemos obtener la eficiencia dividiendo la velocidad de transmisión de información entre la velocidad

teórica de transmisión, RT (en bits por tiempo unitario).

$$E = \frac{RI}{RT} = \frac{B \cdot N}{RT(Tc(N + C) + 2(TT + TL)}$$

Por ejemplo, supongamos que

B= 8 bits/ caracter

N= 80 caracteres de longitud de bloque

C= 20 caracteres de control

RT= 4,800 bps

Tc= 1.7 milisegundos (8 bits/caracter ÷ 4 800 bps)

TT= 60 milisegundos

TL= 150 milisegundos

$$E = \frac{8x80}{4,800(0.0017(80+20)+2(0.060 + 0.150))}$$

E=0.226

Al aumentar el tamaño del bloque duplicando N se aumenta la eficiencia hasta 0.367. Duplicándolo de nuevo (hasta 320) se obtiene 0.534 de eficiencia. La fórmula anterior sólo se aplica a líneas Half-dúplex porque incluyen los tiempos de retorno de línea que son innecesarios en la transmisión Full-dúplex ó si se desconectan los supresores de ecos.

Tratándose de líneas Fulldúplex, el tiempo de retorno de la línea TL de la ecuación, se remplaza por una cantidad nueva más pequeña, TS, que corresponde al tiempo necesario para que el módem se sincronice con la transmisión de entrada. Dependiendo de la velocidad de la línea y del tipo de módem, TS varía desde 5 hasta 50 milisegundos.

Supongamos que se desea conectar una impresora remota a una línea dúplex de 9,600 bps. Qué velocidad de impresión puede lograrse usando registros de información de 120 caracteres? Teniendo en cuenta que los otros factores son:

B= 8 bits por caracter

N= 120 caracteres

C= 12 caracteres

TC= 0.83 ms (8 bits/caracter $\frac{8}{=}$ 9,600 bps)

TT= 60 ms

TS= 20 ms

Tiempo total por bloque=TC(N+C) + 2(TT + TS)

=0.83 (120 + 12) + 2(60 + 20)

=110 + 160 = 270 ms/bloque

lo cual representa $\frac{1.000}{0.270}$ = 3.70 bloques (líneas)/seg.

ó sea 60 x 3.70 = 222 líneas impresas por minuto.

Todas las fórmulas anteriores omiten un factor

sumamente importante: los errores. Si el 1% de todos los bloques tienen al menos un error, entonces la velocidad de información cae en $1\%+(1\%)^2 + (1\%)^3 + ..+1(\%)^n$ (los términos adicionales representan los errores que ocurren cuando se retransmite, re-retransmite, etc.). Qué probabilidad hay de que el 1% de los bloques tengan un error?. Supongamos que la tasa de errores de caracteres es de 1 en 100,00; entonces con caracteres de 8 bits, la tasa de errores es de 1 en 12,500 (100000/8 = 12,5000), Si x es longitud del bloque, (N + C) entonces a la tasa de 1% de errores:

x = 0.01X tasa de errores de caracteres

x = 0.01 x 12 500

x = 125 = N + C para la tasa de error de bloque de 1%

Al aumentar la longitud de los bloques la tasa de error hace que se empeore el caso en proporción creciente debido a la mayor probabilidad de las retransmisiones que se repetiran: también puede provocar problemas de colas. El diseñador del sistema debe de estar alerta a estos problemas potenciales y debe controlar su presencia aplicando el conocimiento y habilidad analítica a su solución.

4.6 CORRECCION DE ERRORES

Las principales medidas que podemos tomar cuando se presenta un error de transmisión son:

substitución de caracter

corrección de error hacia adelante

retransmisión

4.6.1 SUBSTITUCION DE CARACTERES

En muchos casos cuando el caracter no puede ser entendido por el operador, puede usar un chequeo de paridad simple; si se detecta un error de paridad en un caracter se puede substituir el caracter erroneo por algún otro caracter, previamente determinado. En los caracteres ASCII se tiene el caracter de control de comunicación SUB, el cual al ser imprimido ó desplegado envia un signo de interrogación (?) ó una secuencia de tres lineas vertivales (///). Si este caracter aparece en el mensaje, el desplegado se detiene y el operador puede corregir el error.

4.6.2 CORRECION DE ERRORES HACIA ADELANTE

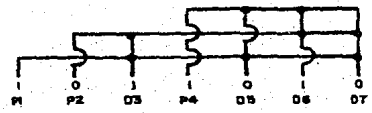
En este tipo de corrección se involucra el uso de códigos especiales de transmisión que contienen suficiente información redundante para que cualquier error detectado se corrija automáticamente al final de la recepción.

La redundancia, ó bits adicionales requeridos para la corrección varia con los distintos planes que existen. Va

desde una pequeña proporción de bits adicionales hasta una redundancia del 100%, lo cual significa que el número de bits para la detección de errores es aproximadamente igual al número de bits de datos. Una de las características de muchas claves de corrección de errores es que debe haber una cantidad mínima de bits correctos entre ráfagas de errores. Por ejemplo, una es estas claves (conocida como clave ó código de Hagebarger), corrige errores hasta de seis bits consecutivos condicionado a que el grupo de seis bits erróneos vaya seguido, por al menos 19 bits válidos antes de encontrarse nuevos errores. Los ingenieros de Bell telephone desarrollaron una clave de corrección de errores que utiliza 12 bits de comprobación por cada 48 bits de datos, es decir una redundancia de 25%. Otra clave de este tipo es la clave de Bose- Chaudhuri, que en una de sus formas puede corregir errores dobles y puede detectar hasta cuatro errores.

Para dar una idea de lo que representa la información redundante cuando se tiene un sistema de detección de errores con corrección automática se considerará la clave de Hamming. Esta clave asocia bits de paridad par, con combinación única de bits de datos. Utilizando una clave de 4 bits de datos como ejemplo, se puede representar un caracter mediante la configuración de bits de datos 1010. Se agregan tres bits de paridad P1, P2, y P4 para producir una clave de 7 bits, como se muestra en la fig. 4.8. En esta figura se puede observar

que los bits de datos (D3, D5, D6, D7), son 1Ø1Ø y que los bits de paridad (P1,P2, P4) son 1Ø1.



a) Relación de prueba de paridad (P) y bits de Datos (D)

0 = PARIDAD CORRECTA 1 = PARIDAD INCORRECTA			DETERMINA EN QUE BIT OCURRIO EL ERROR
P4	P2	P1	
0	0	0	NO HAY ERROR
0	0	1	P1
0	1	0	P2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

b) Interpretación de patrones de bits de paridad

FIGURA 4.8. CLAVE DE HAMMING

Como se muestra en la porción superior de la fig. 4.8, el bit de paridad P1 se aplica a los bits de datos D3, D5, y D7 en tanto que el bit de paridad P2 se aplica a los bits de datos D3, D6 y D7, el bit de paridad P4 se aplica a los bits de datos D5, D6 y D7. Para el ejemplo, en que D3, D5, D6, D7 = 1Ø1Ø, P1 debe de ser igual a 1 porque sólo hay

un 1 entre D3, D5 y D7 y la paridad debe de ser par. En forma análoga, P2 debe de ser Ø ya que D3 y D6 son 1. La paridad P4 debe de ser 1, ya que D6 es el único 1 entre D5, D6, y D7.

Ahora supongamos que durante la transmisión el bit de dato D7 cambio de Ø a 1 a causa del ruido de la línea. Debido a que P1, P2 y P4 prueban este bit de dato, los tres bits de paridad muestran paridad impar en vez de paridad par, que es la correcta (D7 es el único bit de dato probado por los tres bits de paridad, por lo que siempre que D7 esté mutilado los tres bits de paridad mostraran paridad incorrecta). De esta manera, el equipo receptor puede determinar qué bit fue el erróneo e invertir su estado, corriguiendo así el error sin retransmisión.

En la parte inferior de la fig. 4.8 se muestra una tabla que determina la ubicación del bit erróneo. Un 1 en la tabla significa que el bit de paridad correspondiente indica un error de paridad; un cero, significa que la prueba de paridad es correcta. Estos Ø "ceros" y 1 "unos" forman un número binario que indica la ubicación numérica del bit erróneo. En el ejemplo anterior las pruebas de P1, P2, y P4 fueron incorrectas produciendo un 111, que corresponde al 7 decimal, el subindice del bit erróneo.

Existe un número considerable de códigos, ó-claves, que permiten la detección de errores con corrección automática, pero no son usadas con frecuencia en aplicaciones comerciales ya que la cantidad de información redundante que tiene que transmitirse es muy alta en comparación con la cantidad de errores que se pueden presentar. En todo caso, resulta más fácil la retransmisión cuando se ha detectado algún error.

4.6.3 RETRANSMISION

Los dispositivos transmisores y receptores de datos, computadoras delanteras y (frontend) módems tienen incorporados planes de detección de errores y retransmisión, que incluyen la detección de un error y su retransmisión inmediata, detección del error y retransmisión posterior ó detección de un error y retransmisión en, por ejemplo, tres intentos y después retransmisión diferida. La detección del error y la retransmisión es la manera más sencilla, y más efectiva, si se maneja apropiadamente, y menos costosa de reducir los errores en la transmisión de datos. Requiere de lógica más sencilla, relativamente poca memoria, la comprenden mejor los operadores de las terminales y es de uso bastante común. La retransmisión del mensaje erróneo es directa. Por lo general se lleva a cabo cuando el transmisor no recibe un reconocimiento positivo de un plazo

predeterminado. La retransmisión implica que exista un mayor dialogo entre la fuente y el receptor. Esta interacción entre las dos terminales es controlado mediante los diferentes procesamientos de control de linea.

CAPITULO V
"TEORIA GENERAL DE LAS INTERFACES"

5.1 QUE ES UNA INTERFACE

Una interface se define como la unión de miembros de un grupo (tal como gente, instrumentos, etc.) de tal forma que sean capaces de funcionar en una forma compatible y coordinada. Por una "Forma compatible y coordinada", significa sincronizada. A continuación se dan algunas de las definiciones y términos que se utilizan con frecuencia.

SINCRONO.- En sincronismo, o en fase, esto significa el control de la ejecución de una secuencia de operaciones por señales de reloj ó de impulsos.

SINC.- Abreviatura de síncro, sincronizar, sincronización etc.

SINCRONIZAR.- Unir un elemento de un sistema en sincronismo con otro.

IMPULSOS DE SINCRONIZACION.- Impulsos creados por el equipo de transmisión e introducidos en el equipo receptor para mantener los equipos de los dos lugares funcionando en sincronismo.

Podemos ahora definir el interfazar una computadora como:

La sincronización de la Transmisión digital de datos entre la computadora y los dispositivos externos o periféricos, incluyendo la memoria y los dispositivos de Entrada/Salida. La fig. 5.1 muestra un diagrama de lo anterior.

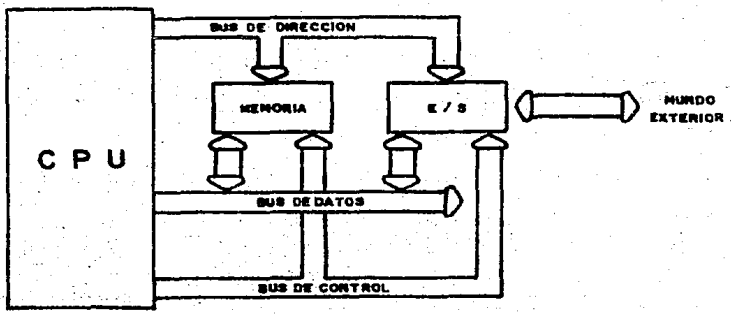


FIGURA 5.1 DIAGRAMA DE FLUJO

Los objetivos básicos de los interfaces se resumen tal como sigue:

ENTRADA: La transferencia de datos desde un dispositivo externo al microprocesador.

SALIDA: La transferencia de datos desde el microprocesador a un dispositivo externo.

CREACION DE IMPULSOS DE SINCRONIZACION: Generar los impulsos de sincronización apropiados para las entradas y las salidas en la transferencia de datos, llamados impulsos de selección del dispositivo, para coordinar las acciones del

dispositivo externo y el microprocesador. Como se muestra en la fig. 5.2.

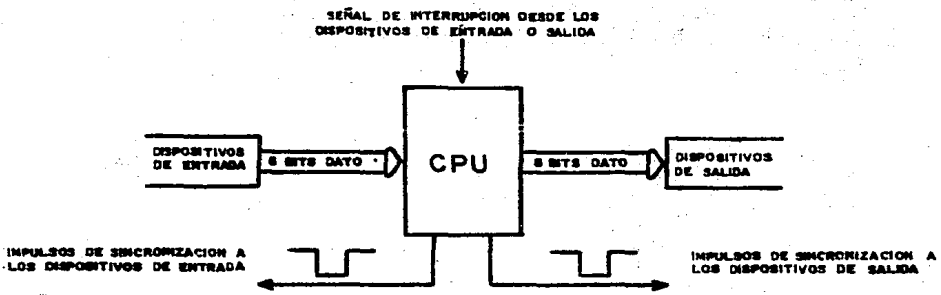


FIGURA 5.2 IMPULSOS DE SINCRONIZACION

MANEJO DE INTERRUPCIONES: Detectar y atender las señales de interrupción que llegan al microprocesador desde los dispositivos externos.

Normalmente los dispositivos de E/S son más lentos que las computadoras. De aquí que una computadora tiene que esperar a que el dispositivo se encuentre listo para recibir o transmitir un nuevo dato. Los buses de dirección, de datos y de control se deben conectar a todos los dispositivos periféricos. La fig. 5.3 ilustra los buses de una microcomputadora con diferentes dispositivos periféricos. Este modo de operación se conoce con el nombre de linea

compartida, en donde cada dispositivo conectado a los buses del sistema se debe comportar como si fuera el único dispositivo conectado al sistema.

Esta condición se logra con el uso de las interfaces, las cuales deben de cumplir con los siguientes requisitos:

a) Decodificar el código de selección del dispositivo que envía la computadora y responde sólo si el código es identico al de él.

b) Decodificar los códigos de los comandos que recibe de la computadora y generar las señales de control para efectuar las operaciones ordenadas.

c) Enviar a la computadora la información que describa el estado del dispositivo periférico.

d) Efectuar la transferencia de datos entre la computadora y el dispositivo periférico.

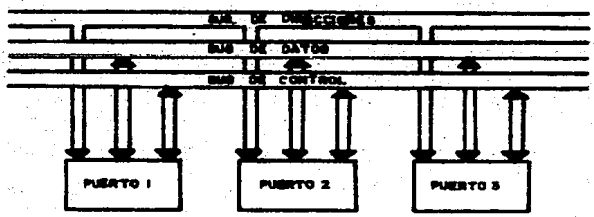


FIGURA 5.3 LINEA COMPARTIDA

5.1.1 SELECCION DEL DISPOSITIVO

Un pulso de selección de dispositivo es un pulso de sincronización generado por la interfaz para sincronizar la transferencia de datos entre la computadora y un dispositivo de entrada o de salida especifico. El término de selección de dispositivo, se asocia con los términos de "seleccion de integrado" o de "habilitar integrado" que se utilizan en los circuitos integrados de memoria. Cada interfaz, debe tener un selector de código de selección, que le permita generar el pulso de selección de dispositivo cada vez que la CPU envia por el bus de dirección su código de selección. Al generarse el pulso de selección del dispositivo, la interfaz queda habilitada para recibir las señales de control de la CPU para el periférico.

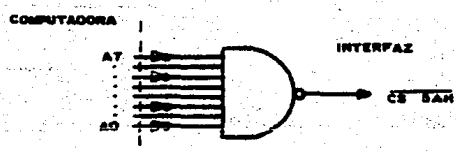


FIGURA 5.4 SELECTOR DE CODIGO

La fig. 5.4 muestra un selector con código de selección 5AH. Cuando el valor en el bus de direcciones es de 5AH la salida de la compuerta NAND pasa a nivel bajo,

indicando a la interfaz que la computadora se va a comunicar con ella. Este tipo de circuito selector debe ser parte de la la interfaz del periférico.

Una forma más simple de generar los códigos de selección es usando un circuito decodificador tales como el 8205 y el 74154, los cuales son decodificadores 3 a 8 y 4 a 16 respectivamente. La fig. 5.5 ilustra un circuito para seleccionar 8 dispositivos. Este circuito decodificador puede ser parte de la microcomputadora y las interfaces deben conectarse a una de estas salidas.

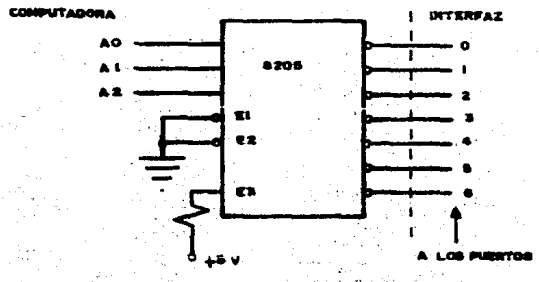


FIGURA 5.5 DECODIFICADOR 3 A 8

5.1.2 DECODIFICADOR DE COMANDOS Y DE CONTROL

Las lineas de control se conectan a todos los dispositivos periféricos. Estas lineas de control pasan a la interfaz a través de compuertas por medio de la señal de

salida del selector de código. De esta forma sólo un dispositivo procesa las señales de control que envía la CPU. La interfaz debe tener un circuito para decodificar las señales de control y posteriormente indicar al dispositivo periférico el comando que debe ejecutar. El circuito decodificador depende de las funciones que puede ejecutar el periférico. Algunos periféricos realizan funciones sencillas de entrada y salida por lo que requieren decodificadores uno a uno que permitan únicamente el paso de las señales de control.

La fig. 5.6 muestra un circuito que permite el paso de las señales de control I/OR, I/OW e INTA para un periférico de E/S código de selección 75H.

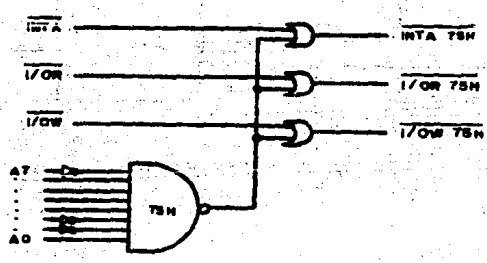


FIGURA 5.6 DECODIFICADOR DE CONTROL

Otros periféricos requieren de más información para realizar adecuadamente sus funciones por lo que las señales de control de la CPU no son suficientes. Por ejemplo, algunos

periféricos necesitan información como la siguiente: velocidad de transmisión, cantidad de bits, número de bits de parada, tipo de paridad de transmisión, etc. Para resolver este problema se utilizan las palabras de control, la interfaz del dispositivo debe tener un registro para recibir la palabra de control y alimentarla al decodificador. Las salidas del decodificador controlan las diferentes partes del dispositivo periférico.

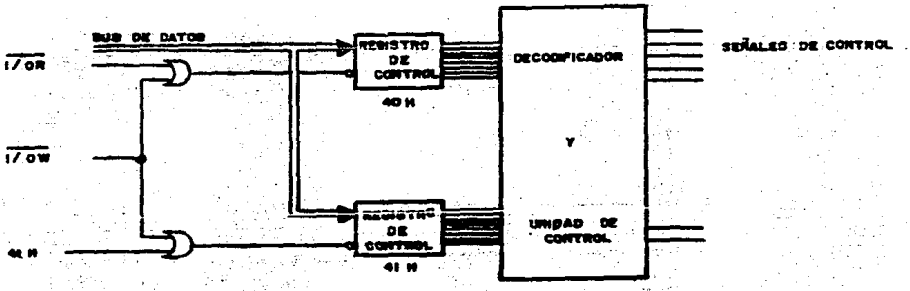


FIGURA 5.7 INTERFAZ

La interfaz puede tener varios registros de control para alimentar al decodificador/unidad de control, dependiendo de la cantidad de información que requieran para generar las señales necesarias del comando ordenado. Para alimentar adecuadamente a los registros estos deben de tener una dirección propia. El circuito de la fig. 5.7 muestra una

interfaz que tiene dos registros de control con códigos de selección 40H y 41H.

5.1.3 BANDERAS DE ESTADO

Como ya se mencionó, las computadoras son más rápidas que los dispositivos E/S, por lo que las computadoras tienen que esperar muchas veces a que el dispositivo periférico se encuentre listo para realizar la función deseada. Por ejemplo, la computadora no debe enviar datos a un periférico de salida sin antes conocer si el dispositivo se encuentra listo para recibir los datos, por lo tanto, el periférico debe contar con un medio para indicarle a la computadora cuando se encuentra listo para recibir el siguiente dato. En forma semejante se requiere de un medio por el cuál un periférico de entrada le indique a la computadora cuando tiene un dato para entrada.

Los estados en que se encuentra un periférico se encuentran en registros de un bit conocidos comunmente como "banderas de estado".

Entre los estados más comunes se pueden mencionar los siguientes: listo para recibir datos, listo para transmitir datos, error de paridad, error de formato del dato, etc. Cuando un estado en particular se encuentra presente, la

bandera correspondiente toma el valor de 1 y cuando no se encuentra presente toma el valor Ø. Al conjunto de registros de banderas se conoce como "palabra de estado" del periférico. Dependiendo de las funciones que pueda desarrollar un periférico su interfaz tendrá más o menos banderas de estado. La unidad de control es la que pone o limpia a cada una de las banderas en función a las operaciones que se encuentre realizando el periférico.

Antes de transmitir o recibir un dato, la computadora debe sensar el estado correspondiente a la función a realizar. Para efectuar esta prueba, la computadora puede leer la palabra de estado del periférico en el registro Acumulador y posteriormente checar el valor de la bandera de interés.

5.1.4 REGISTRO DE DATOS

La interfaz de un periférico, debe contar con un Registro de Datos, que le permita recibir el dato de salida de la computadora, para después enviarlo al periférico de salida y de un registro de datos que le permita recibir un dato de entrada desde el periférico de entrada el cual posteriormente queda disponible a la computadora. La transferencia de datos se realiza entre el acumulador y los registros de datos.

La fig. 5.8 muestra un circuito con las líneas y componentes de una interfaz para la entrada de datos del periférico 80H. Para leer en el Acumulador el contenido del registro de datos la computadora debe ejecutar la instrucción IN 80H. Esta instrucción ordena que se habilite la salida IN 80H permitiendo el paso del contenido del registro de datos al bus de datos a través de las compuertas AND, de donde son leídos en el acumulador cuando la unidad de control de la CPU genera la señal de tiempo correspondiente. La señal IN 80H habilita al buffer de salida de tres estados.

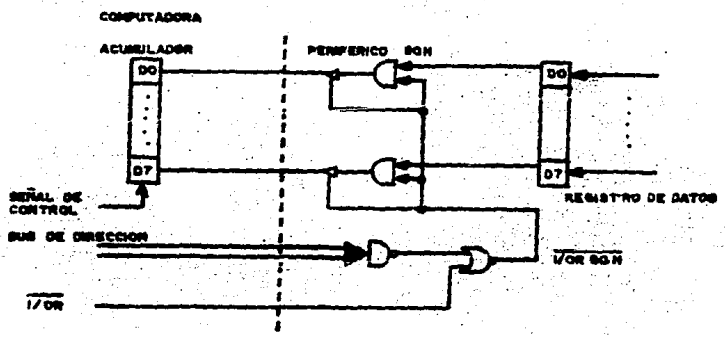


FIGURA 5.8 ENTRADA PERIFERICO 80H

La fig. 5.9 ilustra un circuito con las líneas y componentes de una interfaz para la salida de datos al

periférico 80H. Para escribir el contenido del acumulador, en el registro de datos, la computadora debe ejecutar la instrucción OUT 80H. Esta instrucción envia el contenido del acumulador al bus de datos y habilita a la salida OUT 80H para permitir el paso del dato al registro de datos.

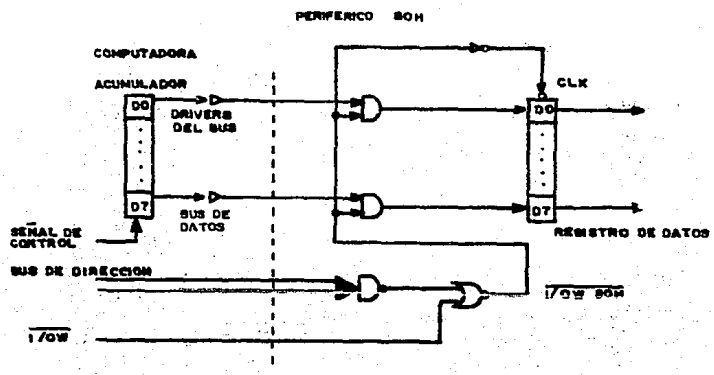


FIGURA 5.9 SALIDA PERIFERICO 80H

5.2 INTERFAZ EN UN INTEGRADO

El microprocesador Z80, cuenta con las instrucciones IN y OUT para efectuar la transferencia de datos entre el acumulador y las interfaces. Se necesitan más de una dirección para especificar el registro en el que va a recibir o enviar el dato y eliminar confusiones en la interpretación de los datos.

Los integrados que realizan las funciones de interfaz, tienen dos entradas para evitar este tipo de confusiones. Estas entradas toman los nombres de $\overline{CS}$ (selección del integrado) y C/$\overline{D}$ (control/dato). La entrada $\overline{CS}$ permite seleccionar el integrado y la entrada C/$\overline{D}$ permite indicarle si es una palabra de control, palabra de estado ó un dato el que esta transmitiendo. Con estas dos entradas se generan las siguientes funciones:

$\overline{CS}$	C/$\overline{D}$	Dato transmitido
Ø	1	Palabra de control ó estado
Ø	Ø	Dato numérico
1	Ø	Nada
1	1	Nada

5.3 INTERFACE RS-232

La RS-232 es una interface estándar que define la interconexión de un equipo de terminal de datos (DTE-Data Terminal Equipment) con un equipo de comunicación de datos (DCE-Data Comunicación Equipment). El flujo de datos entre estos dos equipos es seriado. La interface estándar serial RS-232 es la más ampliamente utilizada, sin embargo su utilización esta limitada para una longitud de cableado muy corto (15 metros) y su respuesta de datos esta arriba de 20 Kbits/segundo. La interface es extensamente utilizada para

conectar computadoras con sus periféricos (terminales de video, impresoras, gráficadoras, etc.). En estos casos la computadora usualmente realiza la función de el DCE, y el periférico realiza la función de el DTE.

La interface RS-232 emplea 4 tiempos de señales de línea:

- Señales de datos
- Señales de control
- Señales de tiempo
- Señales de tierra y retornos.

Las señales de datos tienen un nivel de referencia de "1" lógico para un voltaje negativo, y un espaciamiento de "Ø" lógico cuando es positivo.

Las señales de control y de tiempo se consideran en operación de trabajo cuando tienen un voltage negativo.

Cuando se utiliza la frase "local DCE" significa que el DCE es conectado con el DTE por medio de la interface RS-232.

5.3.1. SEÑALES DE DATOS.

BA: La señal de datos transmitida es generada por el

equipo de terminal de datos. No hay requerimientos específicos de la naturaleza de esta señal, sólo las dos siguientes restricciones son válidas:

1.- La DTE puede manejar la línea de esta señal con la condición señalada entre caracteres ó palabras y también cuando no hay dato transmitido.

2.- La DTE no puede transmitir ningún dato al menos que las siguientes 4 señales esten en posición de encendido (ON):

- Listo para enviar
- Listo para poner dato
- Terminal de datos lista
- Solicitud para enviar.

La señal de dato transmitido es comunmente referida como TxD.

BB: El dato recibido es originario del DCE. Esta señal es generada por la conversión de la señal recibida en el DCE. El dato recibido es manejado con la condición señalada cuando la señal de línea recibida detectada esta en la condición de apagado (OFF). Esta señal es comunmente abreviada como RxD.

SBA: La señal de dato transmitido secundario es equivalente a la señal de dato transmitido pero es usado para transmitir dato por la via de un canal secundario (usualmente en un canal de baja velocidad).

SBB: La señal de dato recibido secundario es equivalente al de la señal de dato recibido pero sobre el canal secundario.

5.3.2 SEÑALES DE CONTROL

CA: La señal de solicitud para enviar es generada por el DTE. Cuando está en la condición de "ON" le permite transmitir el DCE. En un sistema Half-Duplex, la condición "ON" inhibe el modo de recibir; en la condición de tiempo "OFF" manteniendo el modo de recepción.

Cuando se hace la transición de "OFF-ON", el DCE responde con la puesta de "ON" y queda libre la señal para enviar.

CB: Listo para enviar es originada por el DCE. Cuando esta señal esta en "ON", indica que el DCE está listo para transmitir el dato. El dato transmitido puede no ser enviado por el DCE cuando esta señal esté en "OFF". La señal listo para enviar es comunmente abreviada como CTS.

CC: Listo para colocar dato, es generado por el DCE para indicar el estado de dato colocado. Esta instrucción se abrevia como DSR.

CD: Señal de terminal de dato lista es aceptada ó negada por el DTE. la condición establecido entre el DCE local y el DCE remoto. Esta señal es abreviada como DTR.

CE: La señal indicadora de tono se origina del DCE. Una condición "ON" indica que la señal de tono es recibida por el DCE.

CF: La señal de detector de línea recibida es generada por el DCE. Cuando esta en la condición de "ON", el DCE esta recibiendo una señal conocida. Esta señal de línea tambien es abreviada como DCE.

CG: La señal de detector de calidad es suministrada por el DCE. Una condición de "OFF" indica aqui una alta probabilidad de error en el dato recibido. Normalmente una condición "ON" puede estar presente sobre estas lineas.

CH: Esta señal de selector de respuesta de dato es generada por el DTE.

CI: La otra señal de selector de respuesta de dato es suministrada por el DCE. Solamente una de estas dos señales puede estar presente en una interconexión.

SCA: Esta señal de solicitud secundaria para enviar, es equivalente a la solicitud para enviar, excepto que ésta se refiere al canal secundario.

SCB: Esta señal es semejante a SCA (Secundario listo para enviar) es equivalente a la señal primaria de listo para enviar, excepto que esta se encuentra disponible para el canal secundario,

SCF: La señal de detector de linea recibido secundario, indica que la recepción propia de la señal de linea del canal secundario opera de un modo semejante a la señal de detector de linea recibido.

5.3.3. SEÑAL DE TIEMPO

DA: La señal de elemento de tiempo transmisor (fuente DTE) es puesta por el DTE.

DB: La señal de elemento de tiempo transmisor (fuente DCE) es puesta por el DCE.

DD: La señal del elemento de tiempo receptor (Fuente DCE) es generada por el DCE.

PATA	CIRCUITO	DESCRIPCION
1	AA	Tierra de Protección.
2	BA	Dato transmitido.
3	BB	Dato recibido.
4	CA	Solicitud para enviar.
5	CB	Listo para enviar.
6	CC	Listo mensaje ó dato.
7	AB	Señal de tierra (retorno común).
8	CF	Señal detector de línea recibida.
9	--	Reservado.
10	--	Reservado.
11		Sin asignación.
12	SCF	Señal detector de línea secundaria.
13	SCB	Señal listo para enviar secundaria.
14	SBA	Señal dato transmitido secundaria
15	DB	Señal de elemento de tiempo de transmición
16	SBB	Señal de dato recibido secundaria.
17	DD	Señal de elemento de tiempo de recepción.
18		Sin asignación.
19	SCA	Señal de solicitud para enviar secundaria.
20	CD	Terminal de dato listo.
21	CG	Señal detector de calidad.
22	CE	Indicador de tono.
23	CH/CI	Señal de selector de respuesta de dato.
24	DA	Señal de elemento de tiempo transmisor.
25		Sin asignación.

FIGURA 5.10 DESCRIPCION INTERFACE RS-232

SEÑALES DE TIERRA Y RETORNOS

AA: La señal de tierra ó retorno común, provee un potencial de referencias para las señales RS-232 (excepto la tierra de protección).

Una mínima interface puede consistir de tres señales: AA, AB y cualquiera de: BA ó BB. Una interface tipica para una terminal de datos tiene incluidas 8 señales: AA, AB, BA,

BB, CA, CB y CD.

La fig. 5.10 muestra la descripción general de cada una de las patas de la interfaz RS-232.

La fig. 5.11 ilustra las 7 lineas mínimas necesarias para que los datos sean confiables. Estas líneas se especifican a continuación:

1) Solicitud de Envio -RTS: Esta señal va desde el DTE hacia el DCE, este nombre implica que solicita permiso desde el DTE para transmitir dato.

2) Listo para Enviar -RFS (clear to send -CTS): Esta señal desde el DCE informa al DTE que puede empezar a transmitir dato.

3) Dato Transmitido -TxD: Esta linea ó lineas van desde el DCE hacia el DCE conteniendo todos los datos de transmición.

4) Dato Recibido -RxD: Esta linea ó lineas van desde el DCE hasta el DTE, es la que contiene todos los datos de recepción.

5) Tiempo de Transmición TxD.

6) Tiempo de Recepción RxT.

7) Detector de Datos DCD.

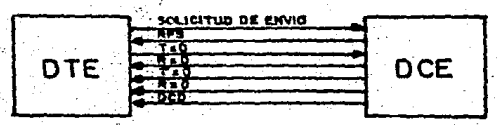


FIGURA 5.11 LINEAS DE UNA INTERFACE

5.4 ESPECIFICACIONES ELECTRICAS RS-232

Un circuito equivalente, para una interconexión a base de una interface RS-232, es mostrada en la fig. 5.11. Este es un circuito tipico para todas las lineas de señales (dato, control, y de tiempo).

La convención tomada por el nivel de señal para la RS-232 es de un voltaje positivo (mayor de +3 volts) para indicar su encendido (ON) y para las condiciones de las señales de control, tiempo y una condición de espacio para dato.

Un nivel de voltaje negativo (más negativo que -3 volts) indica una condición de apagado (OFF) y para una condición de señalización.

El rango de nivel de señal de -3 volts a +3 volts, es lo que se conoce como región de transición. Las condiciones de encendido y apagado (MARK/SPACE) de las señales, no estan definidas en la región de transición.

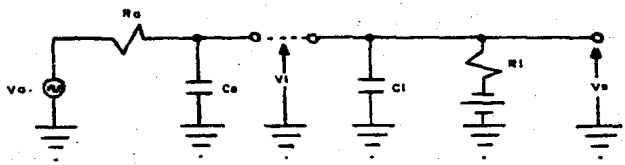


FIGURA 5.12 CIRCUITO EQUIVALENTE RS-232

Los valores de los parametros de la fig. 5.12 se indican a continuación:

a) Voltaje de salida Vo con una RL (3 a 7)Kohms es de:
   Nivel Alto es de +5 a +15 Volts.
   Nivel Bajo es de -5 a -15 Volts.
b) Voltaje de salida Vo a circuito abierto, sin RL a la salida es de:
   Nivel alto es de +3 a +25 Volts.
c) La Ro es mayor 300 Ohms.
d) Io corriente de corto circuito menor que 0.5 Amperes.
e) Impedancia de entrada al receptor RL es de 3 a 7 Kohms.

f) Voltaje en la interface de +5 a +15 Volts.

g) El voltaje a la salida Vs después de corto circuito Vs=Marca de -5 a -15 Volts.

h) Receptor +3 Volts Espacio.

i' Receptor -3 Volts Marca.

j) Capacitancia del Transmisor: no se especifica.

k) Capacitancia del receptor: CL menor ó igual a 25 nf.

l) Voltaje receptor en línea VL menor ó igual +2 Volts.

m) Todas las señales pueden pasar a través de la región de transición en una manera monótona.

n) Para circuitos de control, el tiempo en la región de transición puede ser menos de 1.0 ms. Para datos y señales de tiempo, el tiempo en la región de transición puede ser menos de 1.0 ms. ó 4.0% de la duración de tiempo en el elemento de la señal.

o) El rango máximo de cambio para alguna señal puede ser menor de 30 V/ µs.

p) El componente reactivo para alguna impedancia del receptor puede no ser inductiva.

CAPITULO VI

INTERFAZ PROGRAMABLE DE E/S EN PARALELO (PPI)

6.1 LA 8255 (PPI)

La 8255 PPI fig. 6.1, es un dispositivo de E/S de propósito general programable, diseñado para usarse con los microprocesadores 8080, aunque se puede usar casi en cualquier microprocesador

La 8255 PPI tiene 24 lineas de E/S que se pueden programar individualmente en dos grupos de 12 fig. 6.2, y usarse en tres modos principales de operación (0, 1, 2). En el Modo 0, cada puerto puede ser usado como un simple puerto de entrada ó un simple puerto de salida, fig. 6.3.

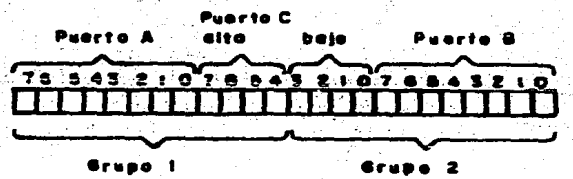
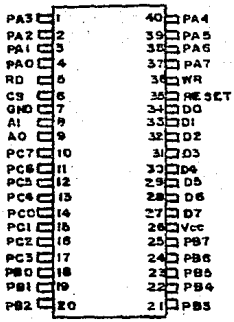


FIGURA 6.2 LINEAS DE E/S

Observe que el puerto C tiene parte alta y parte baja

y se puede usar este puerto separadamente como entrada y/o salida.

CONFIGURACION



NOMBRE DE LAS PATAS

D7-D0	BUS DE DATOS BIDIRECCIONAL
RESET	ENTRADA DE LIMPIAR
CS	SELECTOR DE INTEGRADO
RD	ENTRADA DE LECTURA
WR	ENTRADA DE ESCRIBIR
A0-A1	DIRECCION DEL PUERTO
PA7-PA0	PUERTO A
PB7-PB0	PUERTO B
PC7-PC0	PUERTO C
Vcc	+5 VOLTS
GND	0 VOLTS

DIAGRAMA DE BLOQUES

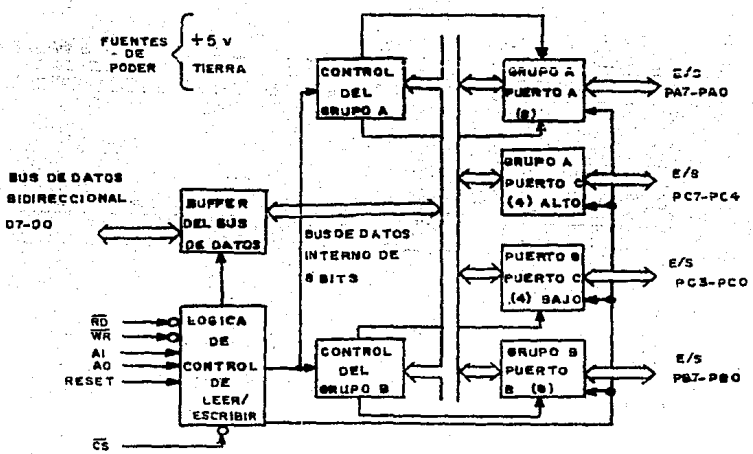


FIGURA 6.1 LA 8255 (PPI)

Cada una de estas 4 unidades, puede asignarles un dato de entrada ó de salida, solo uno a la vez.

FIGURA 6.3 MODO Ø

En el Modo 1, cada grupo se puede programar para tener 8 lineas de E/S, fig. 6.4, de las 4 lineas restantes, 3 se usan para señales de "protocolo" y de control de interrupción.

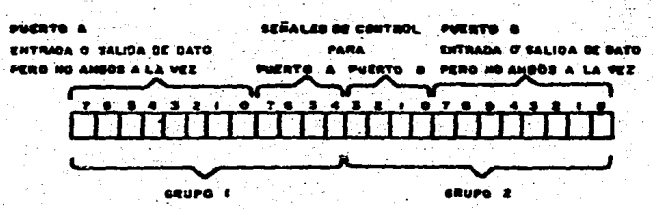


FIGURA 6.4 MODO 1

El Modo 2, es un modo de bus bidireccional en el cual se utilizan las 8 lineas para un bus bidireccional y 5 lineas (pidiendo prestada una al otro grupo) para protocolo.

Cualquiera de las 8 líneas de los puertos B y C de E/S pueden alimentar 1 mA de corriente a 1.5 volts. Esto permite el manejo directo de transistores Darlintong en aplicaciones tales como impresores y displays de alto voltaje.

Las patas y señales de la 8255 estan ilustradas en la fig. 6. 1, tambien se ilustra el diagrama lógico de la 8255.

A continuación se dará una descripción funcional de la 8255.

6.1.1 LINEAS DE ENTRADA Y SALIDA

Las líneas D0-D7 representan el bus de datos bidireccional que comunica con el microprocesador.

La línea PA0-PA7, PB0-PB7 y PC0-PC7 representan los buses de los puertos A, B y C de E/S que conectan con los periféricos.

6.1.2 BUFFER DEL BUS DE DATOS

Este buffer de 3 estados, bidireccional, de 8 bits se usa para interfazar la 8255 con el bus de datos del

microprocesador. El dato se trasmite ó se recibe por el buffer durante la ejecución de las instrucciones de entrada ó de salida por el microprocesador. Palabras de control e información de estados también se transfieren a través del buffer del bus de datos.

6.1.3 LOGICA DE CONTROL Y DE LEER/ESCRIBIR

La función de este bloque es manejar todas las transferencias internas y externas de datos, controles y palabras de estados. Acepta entradas desde los buses de dirección y de control del microprocesador para enviar en seguida comandos a los dos grupos de control (1 y 2)

$\overline{CS}$.

Selector del integrado: Un nivel bajo en esta entrada habilita la comunicación entre la 8255 y el microprocesador.

$\overline{RD}$

Leer: Un nivel bajo en esta entrada habilita a la 8255 para enviar datos o informacion de estados a el microprocesador por el bus de datos. Escencialmente, permite "leer" un dato a el microprocesador de la 8255.

$\overline{WR}$

Escribir: Un nivel bajo en esta entrada habilita a

el microprocesador para escribir datos ó palabras de control en la 8255.

A0 y A1

Selección de los puertos: Estas señales de entrada, en combinación con las entradas $\overline{RD}$ y $\overline{WR}$, controlan la selección de uno de los tres puertos ó del "registro de palabras de control". Ellos estan normarmente conectados a los bits menos significativos del bus de dirección (A0 y A1).

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	OPERACION DE ENTRADA (LEER)
0	0	0	1	0	PUERTO A→BUS DE DATOS
0	1	0	1	0	PUERTO B→BUS DE DATOS
1	0	0	1	0	PUERTO C→BUS DE DATOS
					OPERACION DE SALIDA (ESCRIBIR)
0	0	1	0	0	BUS DE DATOS→PUERTO A
0	1	1	0	0	BUS DE DATOS →PUERTO B
1	0	1	0	0	BUS DE DATOS →PUERTO C
1	1	1	0	0	BUS DE DATOS →REGISTRO DE CTL
					FUNCION DESHABILITADA
X	X	X		1	BUS DE DATOS →TRES ESTADOS
1	1	0	1	0	FUNCION ILEGAL

TABLA 1 OPERACION BASICA

La tabla 1, muestra la tabla de verdad de la transferencia de datos entre el acumulador y los cuatro registros (A, B, C y de control).

LIMPIA (RESET)

Limpiar: Un nivel alto en esta entrada limpia todos los registros internos incluyendo el "registro de control" y todos los puertos (A, B, C) se ponen al modo de entrada.

6.2 CONTROLES DE GRUPO 1 Y GRUPO 2

La configuración funcional de cada puerto se comanda por programación . Escencialmente el microprocesador envia una palabra de control de 8255 que contiene información tal como modo de operación, cuales puertos son de entrada y cuales de salida, fig. 6.3, proporcionando la configuración funcional inicial de la 8255. Cada uno de los grupos de control (1 y 2 fig. 6.5) aceptan comandos de la lógica de control de Leer/Escribir, recibir parte de la palabra de control del bus de datos interno y enviar comandos a sus puertos asociados.

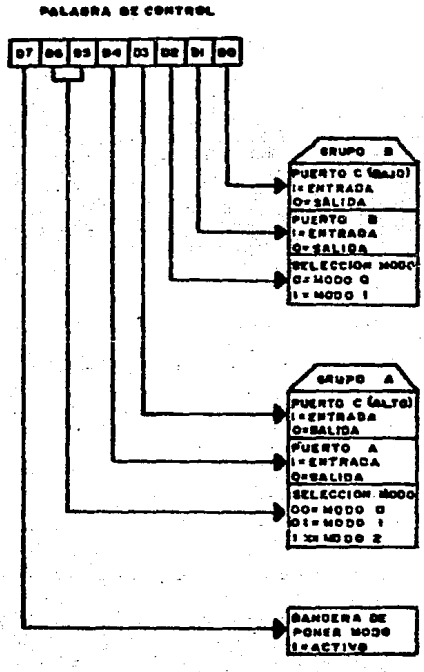


FIGURA 6.5 PALABRA DE CONTROL

Grupo de control 1---Puerto A y puerto C Superior (C7 - C4)

Grupo de control 2---Puerto B puerto C Inferior (C3 - C0)

El registro de control puede solamete ser escrito, no permite la operación de leer su contenido. Los tres puertos de la 8255 tienen la siguiete configuración:

PUERTO A: Un latch/buffer de salida de datos de 8 bits y un latch de entrada de datos de 8 bits.

PUERTO B: Un latch/buffer de entrada/salida de datos de 8 bits y un buffer de entrada de datos de 8 bits.

PUERTO C: Un latch/buffer de salida de datos de 8 bits y un buffer de entrada de datos de 8 bits (sin latch para entrada). Este puerto se divide en dos puertos de 4 bits cada uno bajo el control del modo de operación.

6.3 MODOS DE OPERACION

Un interfaz 8255 trabaja en tres modos principales de operación.

Modo Ø Entrada/salida de básica.

Modo 1 Entrada/salida muestreada (utilizando el pulso strobe).

Modo 2 Bus bidireccional

Cuando la entrada RESET pasa a nivel alto todos los puertos de la 8255 pasan al estado de alta impedancia. Después de quitar el nivel alto de la entrada RESET la 8255

permanece en el modo de entrada. Durante la ejecución de un programa se puede seleccionar cualquiera de los tres modos de operación simplemente cargando el registro de control con la palabra de control adecuada. La fig. 6. 5, ilustra como definir la configuracion de los tres puertos. Para definir la configuración el bit 7 de la palabra de control debe tener el valor 1.

6.3.1 MODO DE OPERACION Ø

El Modo Ø de operacion proporciona operaciones simples de entrada y salida para cada uno de los tres puertos. No se requiere de un protocolo previo, los datos simplemente se leen ó se escriben de un puerto especifico. La 8255 tiene las siguientes características en este modo:

- Tres puertos de 8 bits.
- Cualquier puerto puede ser de entrada ó de salida.
- Las salidas se almacenan en latches.
- Se pueden realizar 16 cónfiguraciones diferentes de E/S

.- Las entradas no se almacenan en latches.

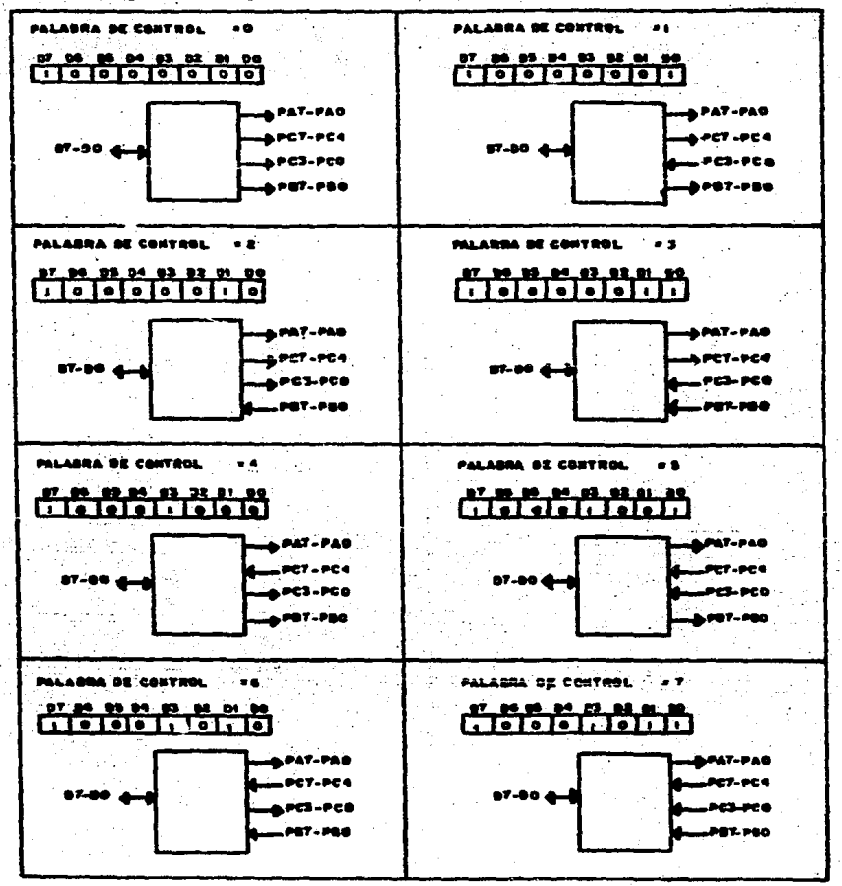
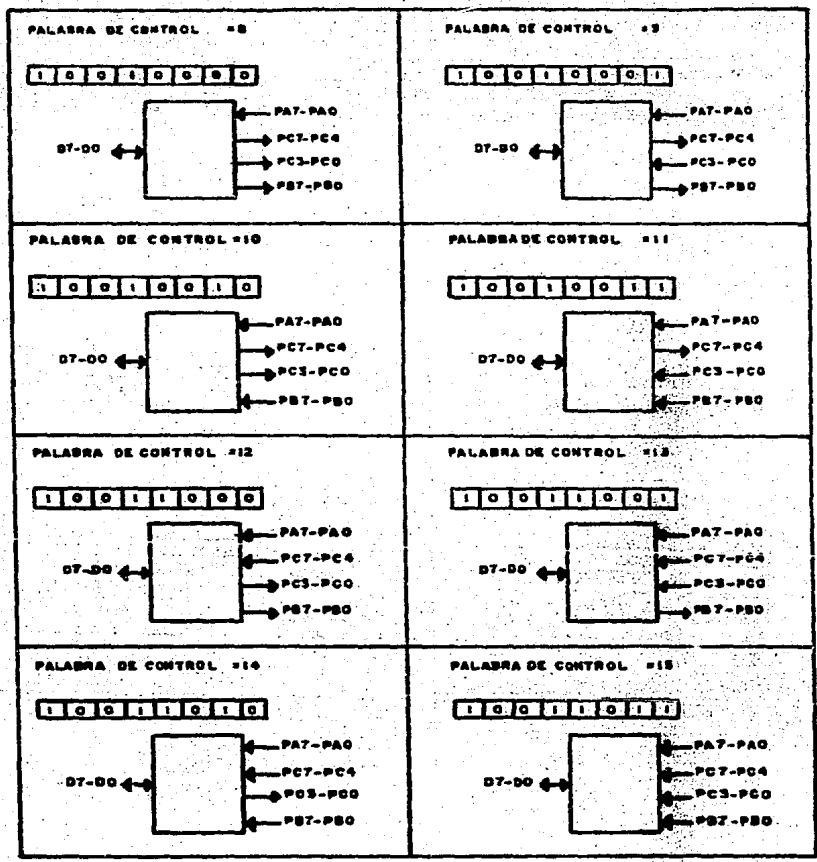


TABLA 2 CONFIGURACIONES DEL MODO Ø



CONTINUA TABLA 2

La tabla 2, ilustra las 16 posibilidades combinaciones y las palabras de control para ordenar cualquiera de ellas y la fig. 6.6, ilustra los tiempos de entrada y salida.

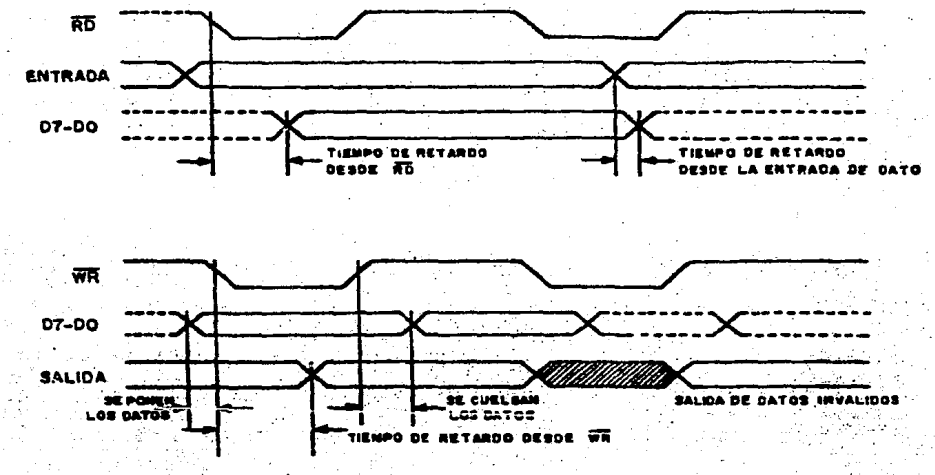


FIGURA 6.6 TIEMPOS DE E/S

6.3.2 MODO 1- MUESTREAR E/S

Esta configuración funcional proporciona una forma de transferencia de datos de E/S con un puerto utilizando señales de muestreo (strobe) ó de protocolo. En este modo los puertos C se utiliza para proporcionar las señales de protocolo. La 8255 programada en el Modo 1 tiene las siguientes características:

- Dos grupos (A y B)

- Cada grupo contiene un puerto de datos de 8 bits y un puerto de control (puerto C) de 4 bits.

- Los puertos pueden ser de entrada ó salida. Tanto las entradas como las salidas se almacenan en latches.

- El puerto de 4 bits de cada grupo se usa para las señales de control y de estados del puerto 8 bits.

En este modo de operación algunas líneas del puerto C se programan para utilizarse de acuerdo a la configuración de los puertos A y B, pero las líneas restantes se pueden utilizar como E/S. El Modo 1, puede ilustrarse como la fig. 6.7. En el modo de entradas solo las líneas PC6 y PC7 se pueden programar para E/S y en el modo de salida las líneas PC4 y PC5 (ver fig. 6.7 y 6.8). Observar que en las palabras de control de las fig. 6.10 y 6.12, no se indica el bit Ø porque las líneas PCØ, PC1, PC2 y PC3 (que son inherentes a este bit en el Modo 1) obtienen la designacion de sus funciones al progrmarse las configuraciones de los puertos A y B. Este bit puede ser Ø ó 1 indiferentemente.

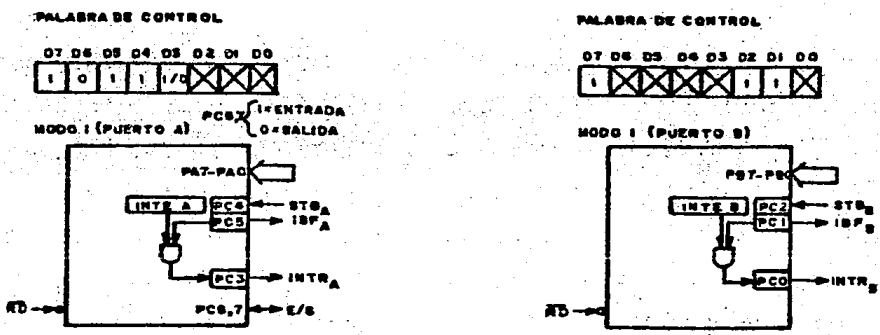


FIGURA 6.7 MODO ENTRADAS

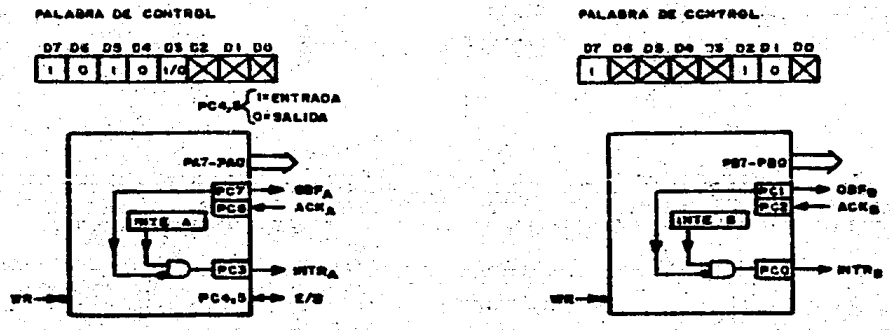


FIGURA 6.8 MODO SALIDAS

6.4 PONER/LIMPIAR EL FLIP-FLOP INTE

Cuando la 8255 se programa para operar el Modo 1 ó Modo 2 las entradas $\overline{STB}$ y $\overline{ACK}$ se pueden utilizar para solicitar una interrupción a el microprocesador.

Estas entradas se combinan a través de una función AND con la salida de un F/F programable interno de la 8255. Si el F/F correspondiente tiene valor alto y las entradas $\overline{STB}$ ó $\overline{ACK}$ se habilitan, la salida INTR del puerto también se habilita. Si esta salida se encuentra conectada a la entrada INT de la Z-80 se realizaá una solicitud de interrupción. Las señales de solicitud de interrupción, generadas desde el puerto C, se pueden habilitar ó inhibir poniendo ó limpiando el flip-flop INTE asociado usando la función del bit SET/RESET del puerto C. La fig. 6.9, ilustra como se forma la palabra de control para poner ó limpiar un F/F INTE. Observar que el bit 7 en este caso es Ø mientras que cuando se generan las configuraciones de los puertos es 1.

Esta función le da la facilidad al programador de impedir ó que un dispositivo de E/S especifico interrumpa a la CPU sin afectar a otros dispositivos en la estructura de interrupción. La definición del F/F INTE es:

(BIT-SET) - INTE es SET - habilitar interrupción
(Bit-RESET) - INTE es RESET -inhabilita interrupción.

Todos los flip-flops de máscara se limpia automaticamente durante la selección del modo y cuando se limpia (RESET = Ø) la 8255.

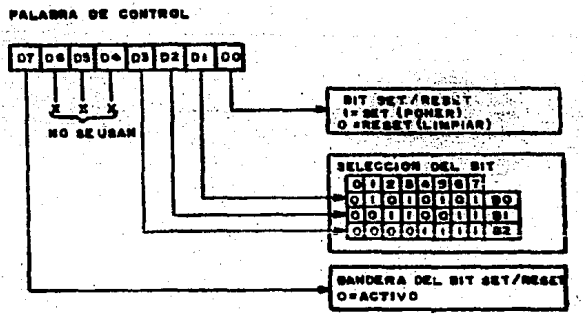


FIGURA 6.9 PALABRA DE CONTROL LIMPIAR F/F INTE

6.4.1 MODO DE ENTRADA

La fig. 6.7 ilustra las palabras de control y las configuraciones para los puertos A y B en Modo 1 como puertos de entrada y la fig. 6.10, ilustra la palabra de control para que los puertos A y B sean puertos de entrada al mismo tiempo.

Los puertos A y B programados como puertos de entrada en Modo 1 permiten a los periféricos indicarle a la CPU que han depositado un dato en el puerto. Los grupos 1 y 2 programados en el Modo 1 como entradas tienen 3 señales de control: $\overline{STB}$, IBF e INTR. La fig. 6.11 ilustra el diagrama de tiempos de las señales de control en el modo de entrada.

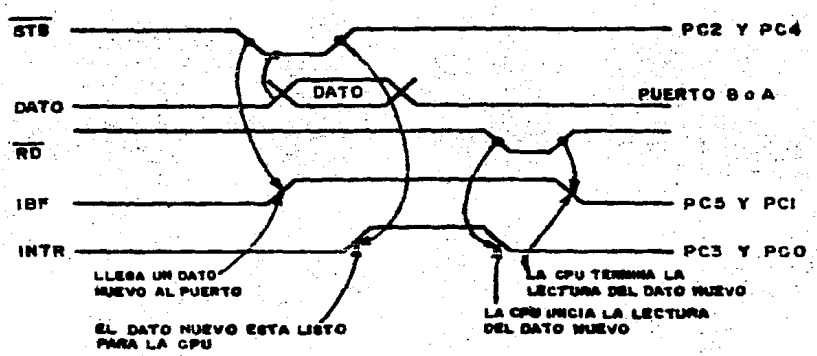


FIGURA 6.11 TIEMPOS EN MODO DE ENTRADA

$\overline{STB}$ (Entrada de Muestrear) --- Un nivel bajo en esta entrada permite a un periférico a cargar un dato de 8 bits en los latches de entrada.

IBF (F/F Buffer de entrada lleno) --- Un nivel alto en esta salida indica que el dato del periférico se ha cargado en los latches de entrada. Se pone con la caída de las señales $\overline{STB}$ y se limpia con la subida de la entrada $\overline{RD}$ cuandoo el microprocesador lee el puerto con la instrucción IN. Entonces, mientras la salida IBF tenga nivel alto está indicando que existe un dato que no se ha procesado y por lo tanto el periférico de entrada no debe enviar otro dato, de realizarlo, el dato anterior se pierde.

INTR (Solicitud de Información) --- Un nivel alto en

esta salida se puede utilizar para interrumpir a el micropocesador si se conecta a su entrada INT. La salida INTR se pone con la subida de un pulso en la entrada $\overline{STB}$ si se cumplen las condiciones de que la salida IBF tiene nivel alto y el F/F INTE está puesto en alto. Este proceso permite que el pulso $\overline{STB}$ almacene el dato en el puerto con la caida de la señal y solicite interrupción a la CPU (con la subida del pulso) para indicarle que tiene un dato nuevo y que puede leerlo.

El F/F INTE (del puerto A) está controlado por el bit Set/Reset PC4 y el F/F INTE B (del puerto B) está controlado por el Bit Set/Reset PC2.

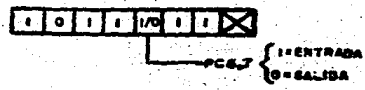


FIGURA 6.1Ø PALABRA PARA PUERTOS A Y B ENTRADA

1.- El periférico carga el dato nuevo en los lanches del puerto de entrada por medio de un pulso negativo en la entrada $\overline{STB}$.

2.- La bajada de un pulso negativo en la entrada $\overline{STB}$ envía a nivel alto la salida IBF, en el cual permanecerá hasta que la CPU lea el dato nuevo. El periférico no deba

enviar un dato nuevo mientras la salida IBF tenga nivel alto.

3.- La subida del pulso negativo en la entrada $\overline{STB}$ envía a nivel alto la salida INTR solicitando interrupción a la CPU para que lea el dato. El F/F INTE debe estar en "1".

4.- La CPU lee el dato nuevo en el Acumulador.

5.- La bajada de un pulso negativo en la entrada $\overline{RD}$ envía a nivel bajo la salida INTR para quitar la solicitud de interrupción.

6.- La subida del pulso negativo RD envía a nivel bajo la salida IBF indicándole al periférico que ya leyó el dato nuevo y puede enviar otro dato.

6.4.2 MODO DE SALIDA

La fig. 6.8, ilustra las palabras de control y las configuraciones para los puertos A y B en Modo 1 como puertos de salida y la fig. 6. 12 ilustra la palabra de control para que los puertos A y B sean puertos de salida al mismo tiempo. Observar que el valor del bit Ø de la palabra de control es indiferente debido a que las terminales PC0 - PC3 se definen

al configu.ar los puertos A y B en el Modo 1.

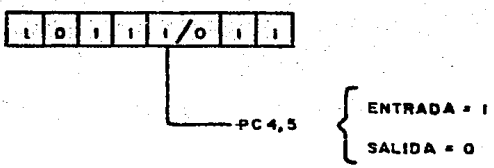


FIGURA 6.12 PALABRA DE CONTROL PUERTOS DE SALIDA

Si los puertos se han asignado como puertos de salida en el Modo 1, la CPU tiene la posibilidad de indicarle a la lógica externa que ha depositado un dato nuevo en el bus de datos. Los grupos A y B programados en el Modo 1 como salidas tienen tres señales de control: $\overline{OBF}$, $\overline{ACK}$ e INTR. La fig. 6.13 ilustra el diagrama de tiempos de las señales de control en el modo de salida

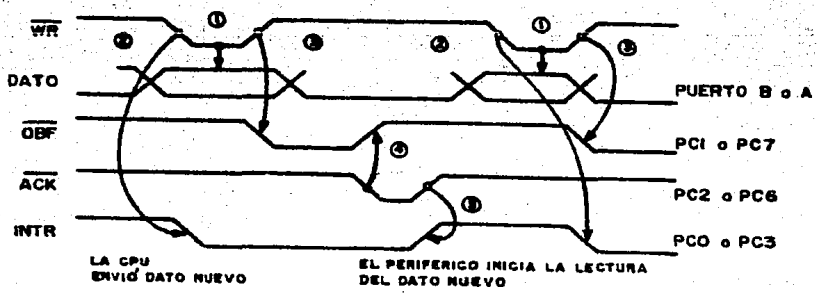


FIGURA 6.13 DIAGRAMA DE TIEMPOS SALIDA

$\overline{ACK}$ (Entrada de Reconocimiento) --- Un nivel bajo en esta entrada, enviando por el periférico de salida, informa a la 8255 que ha leído el dato del puerto especificado. Escencialmente, es una respuesta del dispositivo periférico indicando a la 8255 que ha recibido el dato que envió la CPU.

$\overline{OBF}$ (F/F Buffer de salida lleno) --- La salida $\overline{OBF}$ pasa a nivel bajo para indicarle al periférico de salida que la CPU ha escrito un dato en el puerto especificado. el F/F $\overline{OBF}$ se pone con la subida de la entrada $\overline{WR}$ y se limpia con la caída de la señal de entrada $\overline{ACK}$.

INTR (Solicitud de Interrupción) --- Un nivel alto en esta salida se puede utilizar para interrumpir a la CPU cuando el periférico de salida ha aceptado el dato del puerto, La salida INTR se pone con la subida de $\overline{ACK}$ si la salida $\overline{OBF}$ tiene nivel alto y el F/F INTE está puesto en alto.

El F/F INTE A está controlado por el bit Set/Reset PC6 y el F/F INTE B está controlado por el bit Set/Reset PC2.

El funcionamiento de este modo de salida en forma general es el siguiente:

1.- La CPU deposita un dato en el puerto especificado en un pulso negativo en la entrada $\overline{WR}$ al ejecutar una instrucción OUT.

2.- La bajada del pulso en la entrada $\overline{WR}$ envia a nivel bajo la salida INTR.

3.- La subida del pulso en la entrada $\overline{WR}$ envia a nivel bajo la salida $\overline{OBF}$ en el cual permanecerá hasta que el periférico reconozca que ha leido el dato. La CPU no debe enviar otro dato mientras la salida $\overline{OBF}$ tenga nivel bajo.

4.- El periférico de salida reconoce que ha leido el dato del puerto enviando un pulso negativo a la entrada $\overline{ACK}$. La bajada de la señal $\overline{ACK}$ regresa a nivel alto a la salida $\overline{OBF}$.

5.- La subida del pulso de la entrada $\overline{ACK}$ envia a nivel bajo la salida INTR solicitando interrupción a la CPU para indicarle que puede enviar otro dato. El F/F INTE debe estar en "1".

6.- El envio de un dato nuevo pasa a la salida INTR a nivel alto quitando la solicitud de interrupción.

6.4.3 COMBINACIONES DEL MODO 1

Los puertos A y B se pueden programar individualmente como entradas ó como salidas en el Modo 1 para soportar una amplia variedad de aplicaciones de E/S muestreadas.

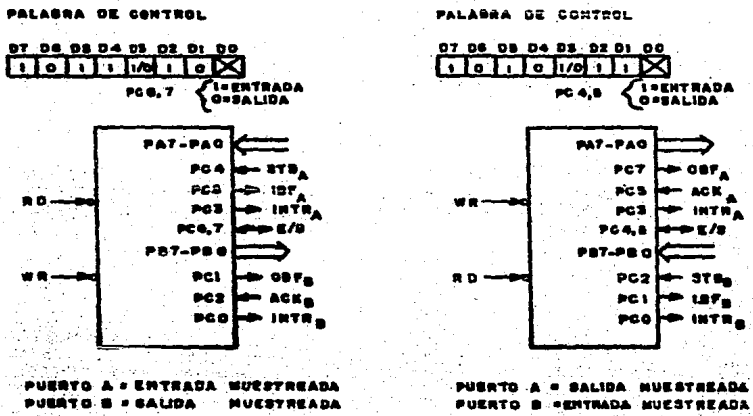


FIGURA 6.14 ENTRADAS Y SALIDAS MUESTRADAS

6.5 BUS BIDIRECCIONAL DE E/S MUESTREADOS

El modo 2 permite al puerto A a actuar como puerto bidireccional de datos. Se proporcionan señales de protocolo a través del puerto C (cinco señales) para mantener una disciplina del flujo de datos del periférico. Estas señales de control son una combinación de las señales de control de entrada y de salida del Modo 1. La generación de solicitudes

de interrupción y las funciones Set/Reset de los flip-flops INTE también se encuentran disponibles. La 8255 programada en el Modo 2 tiene las siguientes características.

---El puerto A como puerto bidireccional de datos de 8 bits y el puerto C como control con 5 bits.

---Las salidas y entradas se almacenan en latches.

La fig. 6.15 ilustra la palabra de control y la configuración del Puerto A en el Modo 2. El diagrama de tiempos de este modo son una combinación de diagrama de tiempos de entrada y de salida del Modo 1, Fig. 6.16.

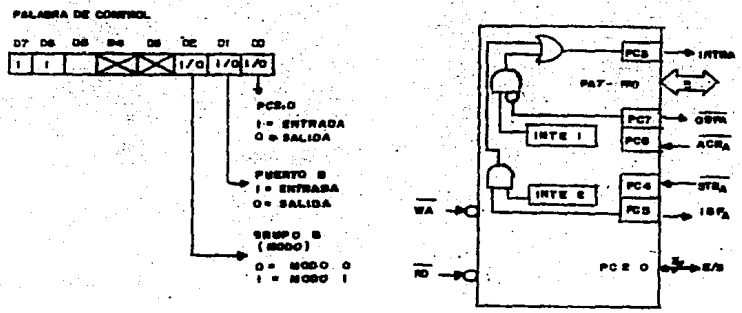


FIGURA 6.15 PALABRA DE CONTROL MODO 2

El Modo 2 no utiliza las terminales PCØ, PC1, y PC2, por lo que el puerto B se puede programar en el Modo Ø ó en el Modo 1.

6.5.1 DEFINICION DE SEÑALES DE CONTROL DE E/S DEL BUS BIDIRECCIONAL.

INTR (SOLICITUD DE INTERRUPCION) --- Un alto en esta salida se puede usar para interrumpir a la CPU para las operaciones de entrada ó salida.

6.5.2 OPERACION DE SALIDA

$\overline{OBF}$ (BUFFER DE SALIDA COMPLETO) --- La salida $\overline{OBF}$ pasa a nivel "bajo" para indicar que la CPU ha escrito datos en el Puerto A.

$\overline{ACK}$ (RECONOCIMIENTO) --- Un nivel bajo en esta entrada habilita al buffer de salida de tres estados del Puerto A para enviar los datos. De otro modo, el buffer de salida estará en el estado de alta impedancia.

INTE 1 (EL FLIP-FLOP INTE ASOCIADO CON $\overline{OBF}$) --- Controlado por el bit Set/Reset del PC6.

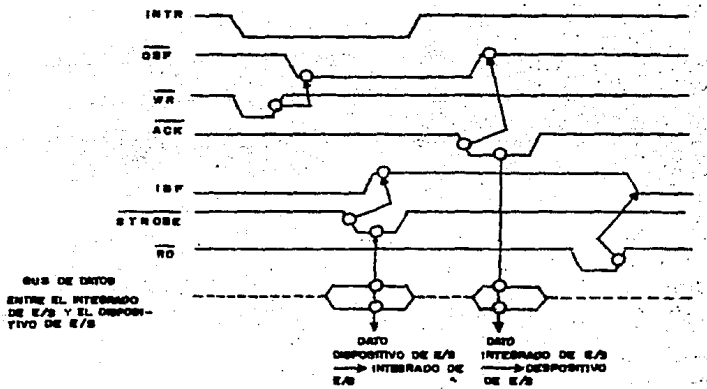


FIGURA 6.16 TIEMPOS DEL MODO 2 (BIDIRECCIONAL)

6.5.3 OPERACIONES DE ENTRADA

$\overline{STB}$ (ENTRADA DE MUESTREO) --- Un nivel bajo en esta entrada carga los datos en el latch de entrada.

IBF (F/F BUFFER DE ENTRADA COMPLETO) --- Un nivel alto en esta salida indica que los datos se han cargado en el latch de entrada.

INTE 2 (EL FLIP-FLOP INTE ASOCIADO CON IBF) --- Controlado por el bit Set/Reset del PC4.

6.6 CONSIDERACIONES DE COMBINACIONES ESPECIALES DE LOS MODOS

Hay algunas combinaciones de modos cuando no todos los bits en el Puerto C se usan para control ó estado. Los bits restantes se pueden usar como sigue:

1.- Si están programados como entradas

- Todas las líneas de entrada se pueden accesar durante una lectura normal del Puerto C.

2.- Si están programados como salidas

- Los bits en la parte alta de C (PC7-PC4) se deben accesar individualmente usando la función bit Set/Reset.

3.- Capacidad de Alimentar Corriente en los Puertos B y C.

Cualquier conjunto de 8 buffers de salida seleccionados aleatoriamente desde los Puertos B y C pueden suministrar 1 ma a 1.5 volts. Esta característica permite a la 8255 manejar directamente los alimentadores tipo Darlington y "displays" de alto voltaje que requieran tal fuente de corriente.

4.- Cuando la 8255 recibe una señal de nivel bajo

en su entrada RESET todos los bits de los puertos y señales de control asociadas se limpian. Esto es significativo cuando la 8255 está operando en los Modos 1 y 2 ya que los periféricos recibirán una señal con nivel bajo en la salida $\overline{OBF}$. El periférico de salida considerará que se encuentra listo un dato de salida. El periférico debe leer y descartar el primer byte después de una señal RESET y posteriormente continuar con el protocolo normal. La 8255 A a diferencia de la 8255 envia a nivel alto a la salida $\overline{OBF}$ después de una señal RESET.

5.- Cuando los Puertos A ó B se asignan como puertos de salida de las patas terminales de salida pueden tener valor alto ó bajo. Cuando el Puerto C (cualquiera de las dos partes de 4 bits) se asigna como puerto de salida todas las patas terminales pasan a nivel bajo.

CAPITULO VII

EL MICROPROCESADOR Z80

7.1 INTRODUCCION

La forma más elemental de definir a un microprocesador es como un circuito integrado capaz de ejecutar un programa y controlar las unidades necesarias para dicha ejecución.

Para poder realizar adecuadamente sus funciones es necesario que actúe junto con otros circuitos electrónicos que lo complementan y le permiten 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/salida.

Las principales caracteristicas de un microprocesador son:

- Longitud de palabra procesada

Las longitudes de palabra más comunes de los microprocesadores actuales son de 8 y 16 bits, aunque también existen algunos que trabajan con palabras de 4 ó 32 bits. Cuanto más largas son las palabras tratadas mayor será la precisión de cálculo del microprocesador y su capacidad de direccionamiento.

- Capacidad de memoria

Esta característica está potencialmente relacionada con la longitud de palabra procesada. La capacidad máxima de memoria accesible por un microprocesador viene marcada por sus posibilidades de direccionamiento. No obstante, microprocesadores de igual longitud de palabra pueden tener distinta memoria en su configuración inicial.

- Velocidad de ejecución de las instrucciones

Se denomina ciclo de instrucción al tiempo que invierte el microprocesador en ejecutar completamente una instrucción; con esta característica queda determinada la velocidad de ejecución de un microprocesador. El factor a considerar a la hora de adoptar esta medida como dato característico, es que el ciclo difiere según el tipo de

instrucción ejecutada.

- Registros especiales

La mayoría de los microprocesadores disponen de un único acumulador en la unidad aritmético-lógica; no obstante, existen microprocesadores que incluyen dos acumuladores, con lo que se amplía su potencia y velocidad de operación.

- Capacidad de interrupción

Una característica básica del microprocesador es la capacidad de recibir y gestionar determinado número de interrupciones. Mediante estas interrupciones se pueden establecer las comunicaciones necesarias, tanto con el usuario como con otras unidades del microcomputador, sin que ello afecte a la correcta ejecución del programa en curso.

- Familia de circuitos complementarios

La necesidad de complementar la operatividad del microprocesador exige el empleo de una serie de circuitos integrados adaptables al mismo. De esta forma surgen distintas familias de circuitos complementarios.

7.2 ARQUITECTURA DE LA CPU Z80

Un diagrama a bloques de la arquitectura interna de la CPU Z80 es mostrada en la fig. 7.1.

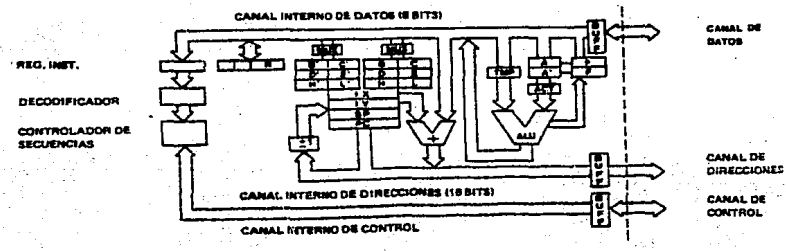


FIGURA 7.1 CPU Z80

7.2.1 REGISTROS

Registros de Propósito Especiales

1.- Contador de Programa (PC): El PC es un registro de 16 bits que contiene la dirección de la instrucción a ser ejecutada. Una vez colocado el contenido del PC en el canal de direcciones, éste se incrementa en uno automáticamente. Cuando en un programa ocurre un brinco el PC se coloca

automáticamente con el nuevo valor.

2.- Apuntador del Stack (SP): El SP es un registro de 16 bits que contiene una dirección de memoria RAM a partir de la cual, en forma descendente, se puede guardar los contenidos de un par de registros ó a partir de la cual, en forma ascendente, se obtiene los dos últimos datos de 8 bits almacenados en esa área. El SP facilita la implementación de múltiples niveles de interrupción, el anidamieto de subrutinas ilimitado y simplifica muchos tipos de manipulaciones de datos.

3.- Dos registros de Indice (IX e IY): Contiene dos registros independientes que son de 16 bits y estos conservan direcciones base que se usan para modo de direccionamiento índice. En este modo, un registro índice se usa como base para apuntar a una región de memoria.

4.- Registro Vector de Interrupción (I): Este registro puede operar en un modo de interrupción en el que responderá como una "llamada indirecta" en respuesta a una solicitud de interrupción. Este registro se usa para este propósito almacenando los 8 bits más significativos de la dirección indirecta mientras que el dispositivo que interrumpe proporciona los 8 bits menos significativos de la dirección indirecta. Esta característica permite que las rutinas de

servicio de las interrupciones pueda localizar en cualquier parte de la memoria y accesar en un tiempo muy corto.

5.- Registro de Refrescar Memoria (R): Este registro es de bits que permite direccionar a las memorias dinámicas a través del canal de direcciones (AØ-A6) cuando la CPU no está accesando a la memoria.

Registros de Acumuladores y de Propósito General

La CPU contiene un acumulador, seis registros de propósito general y un juego completo idéntico de registros alternos, todos ellos de 8 bits.

Los registros BC, DE, HL, BC, DE, y HL pueden ser utilizados como registros pares de 16 bits.

Registros de Banderas

Contiene dos registros (F y F') de cada uno ocho Flip-Flops para monitorear ciertos resultados de las operaciones de la unidad aritmética y lógica (ALU). A la información que almacenan estos Flip-Flops se conoce como banderas de estado. Las banderas se actualizan y no todas las operaciones modifican a todas las banderas.

7.2.2 UNIDAD ARITMETICA Y LOGICA (ALU)

Esta unidad es la que ejecuta realmente el procesamiento. La ALU efectúa todas las operaciones aritméticas y lógicas entre el acumulador y cualquier registro de direcciones ó memoria.

Las condiciones que resultan de cada operación en esta área son transferidas a los bits de las banderas.

7.2.3. REGISTRO DE INSTRUCCIONES Y CONTROL

El registro de instrucciones (RI) recibe del canal de datos un byte, lo interpreta y dependiendo del resultado informa a la unidad de control para que efectúe la secuencia correspondiente a la instrucción recién decodificada.

El circuito de control se encarga de mantener las secuencias necesarias para la operación de la CPU. Después el circuito de control proporciona las señales apropiadas para iniciar correctamente la acción.

Este circuito de control también atiende las señales externas como INTERRUPCION, WAIT, RESET, etc.

7.3 DESCRIPCION DEL CPU-Z8Ø

El CPU-Z8Ø se integra en una pastilla electronica de 4Ø conexiones (patas) con el exterior (fig. 7.2). Esta configuración es clasificada en seis grupos básicos que son:

1.- Bus de direcciones

2.- Bus de datos

3.- Control del sistema

4.- Control de la CPU

5.- Bus de control

6.- Señales de alimentación y reloj

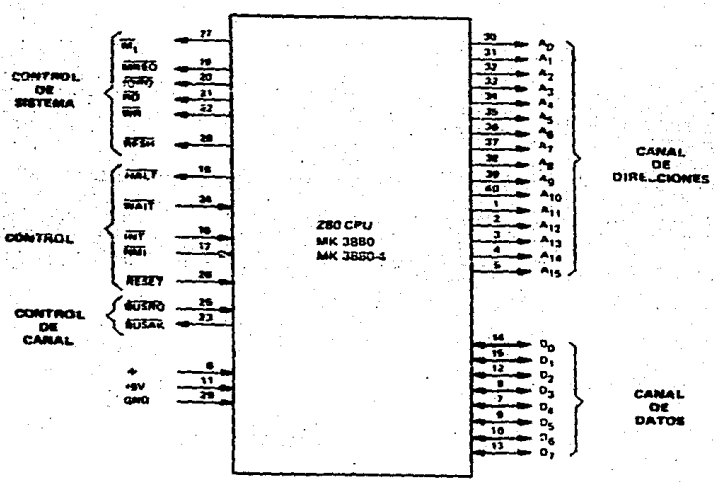


FIGURA 7.2 CONFIGURACION DEL Z8Ø

1.- Bus de Direcciones (AØ-A15): Este Bus es unidireccional y de 16 bits (AØ-A15), donde AØ es el menos significativo. La CPU con estas 16 líneas tiene una capacidad para direccionar hasta 64 K localidades de memoria. Este bus también se usa para enviar el código de Selección de dispositivos de E/S. Durante cierto tiempo en la ejecución de cada instrucción el contenido del Registro R (registro de refrescar memoria) se envía por las siete líneas de más bajo orden para efectuar la función de refrescar memoria. Salida de tercer estado, activa en estado alto.

2.- Bus de Datos (DØ-D7): E/S de tercer estado, activa en estado alto. Este bus es bidireccinal y de 8 bits (DØ-D7), donde DØ es el menor significativo. El bus de datos proporciona la comunicación entre la CPU y la memoria, y los dispositivos de E/S.

3.- Control del Sistema:

A: Ciclo de Máquina Uno (M1): Salida, activa en estado bajo. La línea M1 indica que el ciclo de máquina en proceso es un ciclo de búsqueda (fetch) que se utiliza para obtener el código de operación de la próxima instrucción a ejecutar. M1 es generada cuando cada byte de código de operación es buscado. La señal M1 se activa junto con la señal IORØ para

indicar un ciclo de reconocimiento de interrupción.

B: Solicitud de Memoria (MREQ): Salida de tercer estado, activa en estado bajo. La señal de solicitud de memoria indica que el bus de direciones mantiene una dirección válida para operaciones de leer ó escribir en memoria.

C: Solicitud de Dispositivos de E/S (IORQ): Salida de tercer estado, activa en estado bajo. La señal IORQ indica que las 8 líneas de dirección de más bajo orden (A0-A7) tienen una dirección para el dispositivo E/S válida para operaciones de leer ó escribir en ellos. La señal IORQ se genera junto con la señal M1 cuando la CPU reconoce una interrupción para indicar que se debe colocar un inyector de interrupción en el bus de datos.

D: Leer (RD): Salida de tercer estado, activa en estado bajo. La señal RD indica que la CPU va a leer un dato de la memoria ó un dispositivo de E/S. El dispositivo de E/S ó la localidad de memoria direccionada deberá utilizar esta señal para disparar el dato al bus de datos de la CPU.

E: Escribir memoria (WR): Salida de tercer estado, activa en estado bajo. La señal WR indica que el bus de datos contiene un dato que debe ser almacenado en la memoria ó en un dispositivo de E/S.

F: Señal de refresco (RFSH): Salida, activa en estado bajo. La señal RFSH indica que las siete líneas de más bajo orden del bus de direcciones (A0-A6) contiene la dirección de refresco para las memorias dinámicas y durante la presencia de MREQ debe utilizarse señal de refresco.

4.- Control de la CPU:

A.- Estado de suspensión (HALT): Salida, activa en estado bajo. Este estado HALT indica que la CPU ha ejecutado una instrucción HALT por programa y está esperando una interrupción no enmascarable antes de que pueda reanudar su operación.

B.- Espera (WAIT): Entrada, activa en estado bajo. La señal WAIT indica a la CPU que una localidad de memoria ó un puerto de E/S lentos (que no tiene listo todavía el dato a transferir) ha sido direccionado. La CPU permanece en estado de espera mientras la señal de WAIT esté activada. Todas las señales de datos, de direcciones y de control permanecen en el estado en que se encontraban al momento en que la CPU quedó en espera.

C.- Solicitud de Interrupción (INT): entrada, activa en estado bajo. La señal INT es generada por los dispositivos E/S. Una solicitud será atendida al final de la instrucción

en proceso, si el Flip-Flop interno que permite interrupciones está habilitado y la señal BUSRQ no está activada. Cuando la CBU acepta la interrupción, envia una señal de reconocimiento (IORQ durante el tiempo M1) al inicio del próximo ciclo de instrucción.

D.- Interrupción No Enmascarable (NMI): Entrada. La solicitud NMI tiene una prioridad más alta que la entrada INT y se reconoce al final de la instrucción en proceso, independientemente del estado del Flip-Flop que permite interrupciones. La señal NMI forza automáticamente a la CPU a continuar con la dirección 0066H. El contador del programa se salva automáticamente en el Stack externo de tal manera que el usuario puede regresar al programa que fue interrumpido.

E: Señal de Inicialización (RESET): Entrada, activa en estado bajo. Esta señal RESET forza al contador del programa a cero e inicializa a la CPU. La inicialización de la CPU incluye:

1) deshabilita el Flip-Flop que permite interrupciones
2) coloca al Registro I-00H
3) coloca al Registro R=00H
4) coloca al modo de interrupción 0

Durante el tiempo de inicialización, el bus de datos, bus de direcciones y bus de control se colocan en estado de alta impedancia. Durante el tiempo de inicialización no ocurre la señal de refresco.

5.- Bus de control

A: Solicitud del bus (BUSRQ): Entrada, activa en estado bajo. La señal BUSRQ se usa para pedir que el bus de direcciones, el bus de datos y el bus de control de la CPU pasan al estado de alta impedancia, de tal manera que otros dispositivos externos puedan tomar el control de los buses. Cuando la CPU acepta la solicitud, envía una señal de reconocimieneto (BUSAK) indicando que los buses están en el estado de alta impedancia.

B: Reconocimiento de la solicitud del bus (BUSAK): salida, activa en estado bajo. Esta señal BUSAK, indica que la CPU ha puesto al bus de direcciones, al bus de datos y al bus de control en estado de alta impedancia, es decir, que los buses están disponibles para que otro dispositivo externo pueda tomar el control.

6.- Voltaje de Operación: 5 Volts

Frecuencia de Operción Máxima: 4 Mhz.

7.4 DIAGRAMAS DE TIEMPOS DE LA CPU Z-80

La CPU Z-80 ejecuta las instrucciones pasando a través de la puesta muy precisa de unas cuantas operaciones básicas. Estas incluyen:

Escritura ó lectura de memoria
Escritura ó lectura en dispositivos de E/S
Reconocimiento de interrupciones

Todas las instrucciones son una serie de estas operaciones básicas. Cada una de estas operaciones básicas pueden llevarla de tres a seis períodos de reloj para completarse ó pueden ser alargadas para sincronizar la CPU con la velocidad del dispositivo externo. Los períodos de reloj básico son llamados ciclos T y las operaciones básicas son llamadas ciclo (de máquina) M. La fig.7.3 muestra como una instrucción típica es una serie de ciclos M y T específicos. Estas instrucciones consisten en tres ciclos de máquina (M1, M2 y M3). El primer ciclo de máquina de cualquier instrucción es un ciclo de búsqueda (fetch) el cual puede estar formado por cuatro, cinco ó seis ciclos T (a menos de que sea alargado por una señal de espera). El ciclo de búsqueda (M1) es usado para buscar el código de operación (OP code) de la siguiente instrucción que será ejecutada. Los siguientes ciclos de máquina mueven los datos entre la

CPU y la memoria ó dispositivo de E/S. Cada uno de estos ciclos puede llevar de tres a cinco ciclos T (nuevamente este número puede ser alargado por los estados de espera).

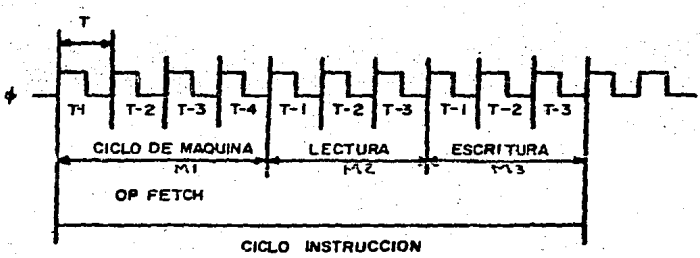


FIGURA 7.3 CICLO TIPICO DE INSTRUCCION

Todos los cronogramas de la CPU caen dentro de alguno de los siguientes diagramas de tiempos:

Instrucción de búsqueda de código de operación (ciclo M1)

Ciclos de lectura ó escritura de memoria

Ciclo de lectura ó escritura de E/S

Ciclo de búsqueda/reconocimiento de bus

Ciclo de búsqueda/reconocimiento de interrupción

Ciclo de búsqueda/reconocimiento de interrupción no enmascarable

Salida de una instrucción HALT.

Los anteriores diagramas muestran las operaciones básicas con y sin estados de espera.

En la fig. 7.4 se muestra los tiempos para las señales durante el ciclo M1. Es muy importante el aprender a leer los diagramas de tiempos porque constituye el medio más utilizado por los fabricantes y diseñadores digitales. En seguida examinaremos con gran detalle el diagrama de tiempo del ciclo M1 para mostrar como leer los diagramas de tiempo de los demás ciclos de máquina de la CPU Z-80.

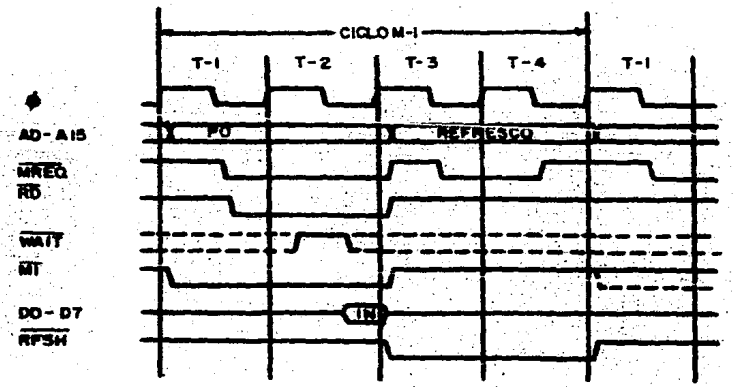


FIGURA 7.4 CICLO M1

La fig. 7.4 muestra los tiempos relativos para ocho señales distintas: Ø, AØ -A15, $\overline{MREQ}$, $\overline{RD}$, $\overline{WAIT}$, $\overline{M1}$, DØ-D7,

$\overline{\text{RESH}}$. La referencia de tiempo es la señal Ø, la cual es la representación gráfica de la señal de entrada del reloj de la CPU Z-8Ø. Las señales que se presentan en la gráfica representan las transiciones en los niveles lógicos bajo a alto y alto a bajo, con lógica positiva de forma que un nivel lógico Ø se díbuja gráficamente como una linea horizontal que es menor que la correspondiente a un "1" lógico.

El diagrama intenta ser real al mostrar transiciones no instantáneas al pasar de Ø a 1 lógico (ó viceversa) que estan representados mediante segmentos no verticales que unen los estados altos con los bajos.

La representación gráfica del bus de direcciones, AØ-A15 en primer instancia algo confuso debido a que no es una sola linea como para la representación de Ø. La razón es que el gráfico debe representar 16 patillas. Idealmente debería haber una correspondencia de uno a uno entre las señales y los gráficos en un diagrama de tiempos; exceptuando las patillas para las direcciones y los datos, éste es casi siempre el caso. Sin embargo, para las direcciones y los datos, la representación que se observa en la fig. 7.4 ha sido adoptada por las siguientes razones:

a. Resulta muy inconveniente representar gráficamente todas las señales de las direcciones ó datos como una línea separada.

b. Es mucho más conveniente, y casi siempre mucho más informativo dibujar la figura en doble línea y escribir un nombre para la dirección ó dato contenido en las líneas, para el tiempo que se muestra.

c. Las X en donde se cruzan dos líneas separa el tiempo para el cual el dato nombrado está contenido en las líneas de direccion ó de datos.

Así, el diagrama de tiempo de la fig. 7.4 muestra que las dieciséis patillas de dirección contienen la información del registro contador de programa (PC) y a continuación una dirección de refresco. En el caso en que el contenido del contador de programa se ha colocado en las líneas de dirección, un tiempo más interesante que el que se muestra, es cuando el PC ha quedado estabilizado en las líneas de dirección. La estabilización de los datos siempre se produce algunos nano_egundos después de que la información ha sido colocada inicialmente en las líneas. Así, generalmente hay menos interés en conocer cuando ha sido colocado el dato en las líneas de datos, que el interés por saber cuando se ha estabilizado esta información en las mismas. Vamos ahora a hacer algunas observaciones de los tiempos relativos acerca del ciclo M1 de la CPU Z-80.

Inmediatamente después del flanco de subida de T1, la señal $\overline{M1}$ es activada: es decir pasar a un nivel bajo desde un estado normalmente alto. También, el contador de programa, ó contenido del registro PC, es colocado en las dieciseis líneas de dirección. A continuación con el flanco de bajada de T1 son activadas las señales $\overline{MREQ}$ y $\overline{RD}$. Un ciclo completo más tarde, se espera que la memoria responda colocando el contenido de la posición especificada de memoria en las ocho líneas de datos D0-D7.

Esto significa que los circuitos integrados de la memoria y sus circuitos asociados tienen cerca de un ciclo lo cual es la posición determinada que se ha fijado y acceden a sus ocho bits de datos. Cuanto más corto es el ciclo T, son más exigentes los tiempos necesarios para la memoria. La señal $\overline{M1}$ es desactivada después del flanco de subida de T3. También en este tiempo, el dato de la memoria ha sido leido por la CPU y puede así quitarse del bus de datos. Una vez que ha termin_do la operación de lectura de memoria, empieza una nueva actividad conocida como refresco. La señal RFSH es activada hasta el flanco de subida de T3, lo cual es simultáneo con la colocación de una dirección de refresco en el bus de direcciones. La señal $\overline{WAIT}$, es una patilla de entrada que es muestreada por la CPU solamente en ciertos momentos.

En todos los demás tiempos no importa cual puede ser el estado de la patilla $\overline{WAIT}$. La CPU durante un ciclo Mi muestrea la patilla $\overline{WAIT}$, solamente en el flanco descendente del reloj durante T2; si esta patilla está en nivel alto, la señal esta desactivada, y la CPU continúa con la secuencia normal de las operaciones de un ciclo Mi. Sin embargo cuando $\overline{MI}$ esta en nivel bajo significa que se debe intercalar un estado de espera dentro del ciclo. Los tiempos de espera se utilizan para extender el tiempo permitido para la respuesta de memoria a más de un ciclo T. Para cada estado de espera intercalado dentro de la ejecución de un ciclo Mi, se le permite a la memoria un ciclo T adicional para su respuesta.

Durante el flanco descendente de cada ciclo T añadido, la CPU muestrea de nuevo la patilla $\overline{WAIT}$ para ver si ha sido desactivada. Si no se añade otro estado de espera la CPU procede a la lectura del dato en el bus de datos y realiza una opración de refresco.

En la fig. 7.5 se muestran los tiempos para un ciclo Mi en el cual se han incluido estados de espera.

En el apendice "B" se muestran los diagramas de tiempos de los restantes ciclos de máquina de la CPU Z80.

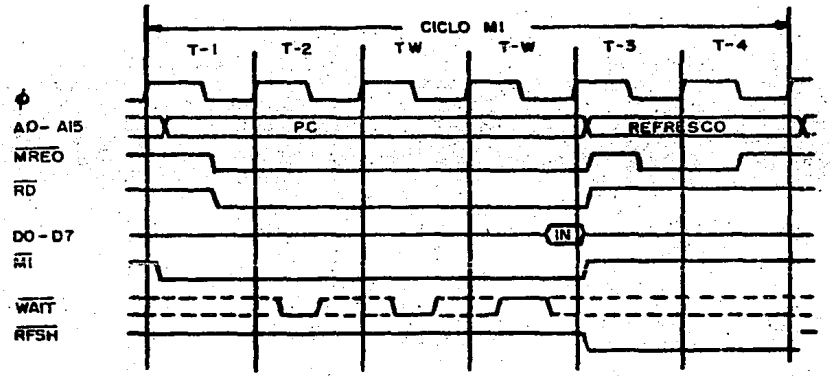


FIGURA 7.5 CICLO MI CON ESTADOS DE ESPERA

7.5 TIPOS DE INSTRUCCIONES DEL Z80

El Z80 puede ejecutar 158 instrucciones independientes incluyendo las 78 instrucciones del 8080A. Estas instrucciones pueden agruparse como sigue:

A. Aritmética e intercambio. Las instrucciones de carga desplazan datos entre registro ó entre registros y memoria. La fuente y el destino de estos datos se especifican dentro de la instrucción. Las instrucciones de intercambio permiten intercambiar el contenido de dos registros.

B. Aritmética y lógicas. Estas instrucciones actúan sobre los datos en el acumulador, un registro ó una posición de memoria designada. Los resultados se colocan en el acumulador y los indicadores de estado se establecen consecuentemente. Las operaciones aritméticas incluyen la adición y la resta de 16 bits entre pares de registros.

C. Búsqueda y transferencia de bloques. El Z80 utiliza una sola instrucción para transferir cualquier bloque de memoria a cualquier otro grupo de posiciones de memoria contiguas. La búsqueda de bloques utiliza una sola orden para examinar un bloque de memoria para un caracter de 8 bits particular.

D. Rotación y desplazamiento. Es posible rotar y desplazar los datos que contenga en acumulador, un registro del procesador central ó memoria. Estas instrucciones tienen también medios de tratamiento del código decimal codificado en binario (BCD).

E. Manipulación de bits. La manipulación de bits incluye las funciones de posicionamiento de inicialización y de prueba (set, reset, test). Los bits individuales pueden modificarse ó probarse en el acumulador, un procesador central ó memoria. Los resultados de las operaciones de pruebas se indican en el registro de estado.

F. Salto, llamada y retorno. Un salto es una bifurcación a una posición de programa especificada por el contenido del contador de programa (pc). El contenido de este último puede proceder de tres modos de direccionamiento: inmediato, extendido e indirecto por registro. Una llamada es una forma especial de salto en donde la dirección que sigue a la instrucción de llamada, se introduce en la pila antes de que se realice el salto. Un retorno es la inversa de la llamada. En esta categoría se incluyen las instrucciones especiales de reinicio.

G. Entrada y salida. Estas instrucciones transfieren datos entre registros y memoria a dispositivos de E/S externos. Se dispone de 256 puertos de entrada y salida. Instrucciones especiales proporcionan el desplazamiento de bloques de 256 bytes a, ó desde, los puertos de E/S y memoria.

H. Control básico de la CPU. Estas instrucciones incluyen la parada funcional de la CPU ó de la de hacer que se ejecute una NOP (sin operación). La capacidad para desinhibir ó inhibir entradas de interrupción es un medio de control suplementario.

7.5.1 MODOS DE DIRECCIONAMIENTO

La mayoría de las instrucciones del Z8Ø operan sobre datos almacenados en los registros internos de la CPU, memorias externas ó en puertos de E/S. El direccionamiento se refiere a la forma como es generada la dirección de cada una de las instrucciones. A continuación se da un breve resumen de los tipos de direccionamiento usados en el Z8Ø.

- Direccionamiento implícito

El direccionamiento implícito se refiere a operaciones en donde el código de operación implica automáticamente a uno ó más registros de la CPU como los que contiene los operandos.

CODIGO DE OPERACION

D7 DØ

- Direccionamiento inmediato

En este tipo de instrucciones el byte siguiente al código de operación es el dato de transferencia (operando).

CODIGO DE EPERACION

OPERANDO

D7 DØ

- Direccionamiento extendido

En este modo de direccionamiento los 16 bits siguientes al código de operación forman el dato de transferencia (operando)

CODIGO DE OPERACION

OPERANDO

OPERANDO

D7 DØ

- Direccionamiento de registro

Este modo de direccionamiento utiliza los registros de la CPU para recibir ó proporcionar el dato. El código de operación contiene el campo de registro ó registros que se utilizan en la ejecución de la instrucción.

CODIGO DE OPERACION

- Direccionamiento de registro indirecto.

Este tipo de direccionamiento emplea registros pares que se utiliza como apuntador de memoria.

CODIGO DE OPERACION Dir. Op.

- Direccionamiento extendido

En esta forma de direccionamiento se proporciona la dirección de memoria en donde puede guardar u obtener un operando.

CODIGO DE OPERACION .....Uno o dos Bytes.
Dir menos significa operando
Dir más significa operando

- Direccionamiento de página cero modificada.

Este tipo de direccionamiento es usado únicamente con la instrucción RESTART (RST). Esta instrucción causa que el proceso del programa continúe en una de las ocho localidades específicas de la "página cero". Estas localidades se definen en ciertas areas de memoria.

CODIGO DE OPERACION ... Un Byte

- Direccionamiento relativo.

El direccionamiento relativo usa el byte siguiente al código de operación para especificar un desplazamiento dentro de un programa al cual éste debe saltar.

El desplazamiento es un número signado con complemento a dos que se suma a la dirección del contador de programa actualizado.

CODIGO DE OPERACION ...Salto relativo
OPERANDO ...Despalzamiento
con complemento
a dos sumado a la
dirección (A+2)

- Direccionamiento indexado

El direccionamiento indexado usa el byte que sigue al código de operación para especificar un desplazamiento que se suma a uno de los dos registros índices para obtener la dirección efectiva de memoria.

CODIGO DE OPERACION
CODIGO DE OPERACION
DESPLAZAMIENTO

- Direccionamiento de bit

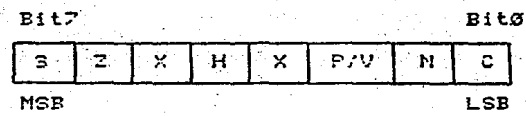
Este tipo de direccionamiento permite poner en 1, limpiar y probar un bit de un operando. Lo anterior facilita efectuar operaciones a nivel de bits sobre los contenidos de

localidades de memoria ó registros seleccionados por medio de un tipo de direccionamiento (registro, registro indirecto e indexado), mientras que tres bits del código de operación especifican cual de los 8 bits del operando es el involucrado.

Los siguientes simbolos y abreviaturas son usados en las tablas del apéndice "A":

SIMBOLO	SIGNIFICADO
r	Cualquiera de los registros de la CPU (A,B,C,D,E,H,L).
s	Cualquier localidad de 8 bits para todos los modos de direccionamiento permitidos por la instrucción particular.
ss	Cualquier localidad de 16 bits para todos los modos de direccionamiento permitidos por esta instrucción.
ii	Cualquiera de los dos registros subindizados IX ó IV.
R	Contador de refresco.
n	Cualquier número de 8 bits en el rango de Ø a 255.
nn	Cualquier número de 16 bits en el rango de Ø a 65535.

7.6 INDICADORES DE ESTADO DEL Z80 (FLAGS)



C= indicador de acarreo. C= 1 si la operación produce acarreo del MSB (bit más significativo) del operando ó resultado.

Z= Indicador de cero. Z= 1 si el resultdo de la operación es cero.

S= Indica el signo. S=1 si el MSB del resultado es 1

P/V= Indicador de paridad ó desbordamiento. La paridad (P) y el desbordamiento (V) se muestra en el mismo indicador de estado, ya que las operaciones lógicas para los dos es la misma. Si P/V toma la paridad, P/V=1 si el resultado de la operación es par, P/V Ø si de la operción es impar. Si P/V toma el desbordamiento, P/V= 1, si el resultado de la operación produce un desbordamiento

H= Indicador de semiacarreo. H=1 si la operación de suma ó resta produce un acarreo dentro ó prestado del bit 4 del acumulador.

N= Indicador de suma/resta. N=1 si la operación anterior fue una resta.

X= No utilizado.

FIGURA 7.6 POSICION E IDENTIDAD BITS DE ESTADO

Los registros de estado (F.F.) suministra información al usuario con respecto al estado del procesador central en cualquier momento. Hay cuatro bits de estado comprobables y dos no comprobables en cada registro. En la fig. 7.6 se indica la posición y la identidad de estos bits de estado.

En el sumario que se da en la tabla 7.1 se muestran los diferentes simbolos en los indicadores de estado:

- • El indicador no se afecta con la operción.
- 0 El indicador es inicializado por la operación.
- 1 El indicador es activado por la operación.
- x El indicador no es tomado en cuenta.
- V El indicador P/V es afectado de acuerdo al resultado de desbordamiento de la operación.
- P El indicador P/V es afectado de acuerdo al resultado de la paridad de la operación.
- I El indicador es afectado de acuerdo al resultado de la operación.
- IFF El contenido del Flip-Flop de habilitación de interrupciones es copiado por el indicador P/V.

En la tabla 7.1 se muestra como cada indicador de estado es afectado por varias instrucciones de la CPU. Notese que la instrucción de búsqueda de bloques (CPI; CPIR; CPD; CPDR) activa el indicador Z si la operación, comparar

con el anterior, indica una igualdad entre la fuente y el dato del acumulador. También, el indicador de paridad es activado si el contador byte (registo par BC) no es igual a cero.

INSTRUCCION	C	Z	P/V	S	N	H	COMENTARIOS
ADD A, s; ADC A, s	‡	‡	V	‡	0	‡	Suma de bits con o sin acarreo
SUB s; S BC A, s; CP s	‡	‡	V	‡	1	‡	Resta de 8 bits con o sin acarreo, comparada con el acumulador
AND s	0	‡	P	‡	0	1	Operaciones Lógicas
OR s; XOR s	0	‡	P	‡	0	0	
INC s	•	‡	V	‡	0	‡	Incremento de 8 bits
DEC m	•	‡	V	‡	1	‡	Decremento de 8 bits
ADD DD, ss	‡	•	•	•	0	X	Suma de 16 bits
ADC, HL, ss	‡	‡	C	‡	0	X	Suma de 16 bits con acarreo
SBC, HL, ss	‡	‡	V	‡	1	X	Resta de 16 bits con acarreo
RLA; RLCA; RRA; RRCA	‡	•	•	•	0	0	Rotación del acumulador
RL m; RCL m; RR m; RR C m SLA m; SRA m; SRL m	‡	‡	P	‡	0	0	Rotación y corrimientos de la localidad m
RLD; RRD	•	‡	P	‡	0	0	Rotación decimal
DAA	‡	‡	P	‡	•		Ajuste decimal del acumulador
CPL	•	•	•	•	1	1	Complementa al acumulador
SCF		•	•	•	0	0	Enciende bandera de acarreo
CCF		•	•	•	0	X	Complementa acarreo
IN r, (C)	•	‡	P	‡	0	0	Entrada de puerto a registro
INI; IND; OUTI; OUTD	•	‡	X	X	1	X	Entrada y Salida de bloques
INIR; INDR; OTIR; OTDR	•		X	X	1	X	Z = 0 si solo si B = 0 0
LDI; LDD	•	X		X	0	0	Transferencia de bloques
LDIR; LDDR	•	X	0	X	0	0	P/V = si solo si B = 0
CPI; CPIR; CPD; CPDR	•	‡	‡		1	X	Búsqueda de bloques Z = si solo si A = (HL) P/V = 1 si solo si B = 0
LD A, I; LD A, R,	•	‡	IFF	‡	0	0	El contenido del biestable que permite interrupciones (IFF) es colocado en la bandera P/V
BIT b, s	•	‡	X	X	0	1	El complemento de bit b de s es colocado en la badera Z
NEG	‡	‡	V	‡	1	‡	Negación del acumulador

TABLA 7.1 OPERACION DE BANDERAS

La instrucción de movimiento de bloque, en este caso, puede tener el mismo uso que el indicador de paridad. Otro caso especial se tiene durante la instrucción de entrada ó

salida de bloques,, aquí el indicador Z es usado para mostrar el estado del registro B el cual es usado como un contador de byte. Notece que cuando el bloque de transferencia de E/S es completado, el indicador de cero será restablecido a cero (es decir B=Ø) mientras que el caso del comando de movimiento de bloque el indicador de paridad es restablecido cuando la operación es concluida. Por último, cuando el registro de refresco (registro I) es cargado dentro del acumulador, el Flip-Flop de habilitación de interrupciones es cargado en el indicador de paridad para que así el estado completo de la CPU pueda ser salvado en cualquier momento.

7.7 RESPUESTA A LAS INTERRUPCIONES

En general las transiciones que provoca una interrupción están caracterizadas por los siguientes acontecimientos:

a,. Se produce la interrupción.

b. Se detiene la tarea actual y el estado actual se guarda para reanudarlo más tarde.

c. Se reconoce la interrupción indicándoselo al que ha solicitado la interrupción.

d. El servicio de la interrupción continúa hasta que se produce otra interrupción ó que se ha terminado de servir ésta.

Las transiciones provocadas cuando se acaban las tareas están caracterizadas por los siguientes acontecimientos:

a. Se termina la tarea interruptiva.
b. Se vuelve a llamar el punto de parada de la tarea interrumpida.
c. Se vuelven a asignar recursos a la tarea interrumpida.

Existen alternativas a los acontecimientos procedentes.

Supongamos que por alguna razón el microprocesador no puede ser interrumpido. En lugar de reconocer la interrupción puede elegir ignorarlos y dejarlos sin servicio. En este estado el microprocesador a bloqueado las interrupciones. Desde luego es probable que exista una interrupción que no pueda ser ignorada. A este tipo de interrupción se le llama "no enmascarable" y se distinguen de las enmascarables porque tienen una mayor prioridad.

Entre el conjunto de todas las interrupciones enmascarables se pueden asignar prioridades que permitan que las tareas con mayor prioridad se efectúen sin ser interrumpidas por las tareas de menor prioridad.

La CPU Z80 tiene dos entradas de interrupción una interrupción enmascarable en software y una no enmascarable. La interrupción no enmascarable ($\overline{NMI}$) no puede ser deshabilitada por el programador y será aceptada siempre que un dispositivo periférico la requiera. Esta interrupción es generalmente reservada para funciones muy importantes que deben ser servidas siempre que ocurran, como es el caso de una falla en la energía eléctrica. La interrupción enmascarable ($\overline{INT}$) puede ser seleccionada como habilitada ó deshabilitada por el programador. Esto permite al programador deshabilitar la interrupción durante periodos donde su programa tiene ciclos restringidos que no pueden ser interrumpidos. El Z80 tiene un Flip-Flop (llamado IFF) que puede ser activado ó desactivado por el programador usando las instrucciones de Interruptor Habilitado (EI) e Interruptor deshabilitado (DI). Cuando el IFF es restablecido, no será aceptada ninguna interrupción por la CPU.

Actualmente la CPU Z80 utiliza dos Flip-Flop internos para saber si las interrupciones enmascarables estan habilitadas ó bloqueadas. Estos Flip-Flops son llamados IFF1 y IFF2, en donde IFF1 se utiliza para controlar la activación de las interrupciones enmascarables, mientras que IFF2 se utiliza algunas veces como posición de almacenamiento

temporal para IFF1. Los dos Flip-Flop están controlados automáticamente por la CPU bajo las siguientes circunstancias:

a. Si la CPU recibe la señal reset los dos Flip-Flop son puestos a cero de forma que las interrupciones enmascarables quedan bloqueados.

b. Cuando una interrupción es aceptada por la CPU los dos Flip-Flop son puestos a cero de forma que las interrupciones enmascarables quedan bloqueadas. Para habilitar y bloquear las interrupciones enmascarables bajo el control del software, la CPU Z80 tiene dos instrucciones: DI (F3H) para bloquear las interrupciones, y el (FBH) para habilitar las interrupciones. La instrucción EI provoca que IFF1, y IFF2 se coloquen en 1, mientras que la ejecución de DI hace que los dos Flip-Flop se pongan en cero. Una propiedad importante de la instrucción EI es que hay un retardo de una instrucción antes de que puedan ser aceptadas interrupciones enmascarables por la CPU Z80. Por ejemplo la secuencia de instrucciones:

```
EI
RTI
```

no provocará la aceptación de interrupciones enmascarables

hasta después de que se haya ejecutado la instrucción RTI (notece que RETI es un retorno desde una instrucción de interrupción que es utilizada en las rutinas de servicio de interrupción RETN). El retardo de una instrucción para habilitar las interrupciones es esencial ya que es muy frecuente en las rutinas de servicio de interrupciones, finalizar con la anterior secuencia de instrucciones. El retardo de una instrucción asegura que el retorno desde una interrupción será ejecutado antes de la aceptación de otra interrupción. Esto evita que el stack vaya creciendo innecesariamente debido a las interrupciones que se producen durante la ejecución de la instrucción EI. Notece que el efecto de la instrucción DI es inmediato. En seguida se presenta un resumen de las diferentes instrucciones de los Flip-Flops.

Acción	IFF1	IFF2	
Reset de la CPU	Ø	Ø	
DI	Ø	Ø	
EI	1	1	
LS A, I	.	.	IFF2 indicador de paridad.
LD A, R	.	.	IFF2 indicador de paridad
NMI aceptada	Ø	.	
INT aceptada	Ø	.	

RETN IFF2 . IFF2 IFF1

RETI

.= SIN CAMBIO

El Z80 puede ser programado para responder a las interrupciones enmascarables entre posibles modos.

MODO Ø

Este modo es idéntico al modo de respuesta de interrupción del 8Ø8Ø. Con este modo, el dispositivo interruptor puede colocar cualquier instrucción en el bus de datos y la CPU lo ejecutará. De esta forma el dispositivo que esta interrumpiendo proporcionará la siguiente instrucción que será ejecutada.

A menudo esta es una instrucción de rearranque que consiste en un solo byte. Alternativamente cualquier otra instrucción, tal como una llamada de 3 bytes a cualquier localidad de memoria, puede ser ejecutada.

El número de ciclos de reloj necesarios para ejecutar esta instrucción es de dos o mas que el número normal para la instrucción. Esto ocurre porque la CPU automáticamente agrega dos estados de espera para un ciclo de respuestas a interrupción que da suficiente tiempo para implemente (un

eslabonamiento externo para control de prioridad).

MODO 1

Cuando este modo ha sido seleccionado por el programador, la CPU responderá a una interrupción con la ejecución de rearranque a partir de la locadidad ØØ38H. De esta manera la respuesta es idéntica a la interrución no enmascarable excepto por la llamada a la localidad ØØ38H en lugar de ØØ66H. Otra diferencia es que el número de ciclos requerido para completar la instrucción de rearranque es 2 más que lo normal debido a los dos estados de espera agregados.

MODO 2

Este modo de interrupción es el que tiene mayor prioridad. Con un simple byte del usuario se puede hacer una llamada indirecta a cualquier localidad de memoria.

Con este modo el programador mantiene una tabla de 16 bits de direcciones de inicialiación para todas las rutinas de servicio a interrupcinones. Esta tabla puede localizarse en cualquier parte de la memoria. Cuando una interrupción es aceptada, un apuntador de 16 bits puede ser formado para obtener la rutina de servicio a interrupción deseada

inicializando la dirección desde la tabla. Los primeros 8 bits de este apuntador estan contenidos dentro del registro I. El registo I puede ser cargado previamente con el valor deseado por el programador (por ejemplo LD I,A). Los ocho bits menos significativos del apuntador puede ser reemplazados por un dispositivo de interrupción.

Este modo de respuesta requiere 19 periodos de reloj para ser ejecutado (7 para el ciclo de búsqueda, 6 para salvar el contador de programa, y 6 para obtener el salto de dirección)

CAPITULO VIII

EL MICROKIT MKE-Z80

8.1 CARACTERISTICAS GENERALES

El sistema MKE-Z80 es considerado como una computadora, sin embargo esta clasificación que se le da esta muy generalizada, ya que debido a los elementos que la componen y a su tamaño puede ser considerada como una nanocomputadora ó microcomputadora.

El microcomputador está ensamblado en una sola placa de circuito impreso y tiene como unidad central de procesamiento, el microprocesador Z80. Cuenta con un teclado y un desplegado pseudo-alfanumérico como interfase hombre-máquina.

8.1 ESPECIFICACIONES DEL MKE-Z80

MICROPROCESALJR EMPLEADO: Z80

CARACTERISTICAS DEL MICROPROCESADOR:

- Fabricante: Zilog
- Cantidad de instrucciones: 158

- Registros internos: 22
- Modos de direccionamiento: 10
- Longitud de dirección: 16 Bits
- Longitud de dato/instrucción: 8
- Número de niveles de interrupción: 1
- Frecuencia máxima de reloj: 4 MHz.
- Tiempo de instrucción mínimo 1 microseg.
  máximo 5.75 microseg.
-Rango de direccionamiento: 64 kbytes.

MEMORIA

-EPROM 2716; 2 Kbytes (expandible a 4 Kbytes) utilizado como almacenamiento de programa Monitor.
-RAM 2114: Kbytes (expandible a 2 Kbytes) para el usuario.
-Direcciones:

EPROM 0000H-07FF (Programa Monitor y Subrutinas).
EPROM 0800H-0FFFH (Opcional).
RAM 1000H-13FFH (Usuario).
RAM 1400H-17FFH (Opcional).

PUERTOS DE ENTRADA/SALIDA

-Paralelo: 24 canales programables como entradas ó salidas (expandibles a 48 canales). El MKE utiliza 15 de los primeros canales para el manejo de su teclado, desplegado y

varios.

INTERFACES

-Dos conectores de 26 terminales cada una con:

33 canales (E/S) TTL.
Reloj del sistema (S) TTL.
Interrupción NMI (E) TTl.
Wait (E) TTL.
Reset (E) TTl.
Nivel de referencia (Tierra).

SOFTWARE

-Programa Monitor HOLA grabado en memoria EPROM 2716 a partir de la dirección 0000H-07FFH.

COMANDOS

-COM 1: MEM (para accesar la memoria).
E/S (para accesar a los puertos de E/S).
REG (para accesar a los registros de la CPU).

-COM 2: CDR (para cálculo de desplazamientos relativos).
MBM (para mover bloques de memoria).

CM (para comparar areas de memorias).

DAM (para despliegue automático de memoria).

DAR (para despliegue automático de registros).

PM (para programar memorias).

PUNTOS DE INTERRUPCION POR PROGRAMA

- 7 puntos (RSTØ8H-RST38H).

TECLADO

- De membrana con 23 teclas.

DESPLEGADO

- Pseudo-alfanumérico de 6 digitos (7 segmentos).

DIMENSIONES

- Ancho 242 mm.
- Largo 221 mm.
- Alto 35 mm.

ALIMENTACION

- +5 Volts +5%, 1 Ampere en máxima configuración.

MEDIO AMBIENTE

- Operación normal entre Ø y 6Ø C.

En la fig 8.1, se muestra la estructura física del MKE-Z8Ø vista anteriormente

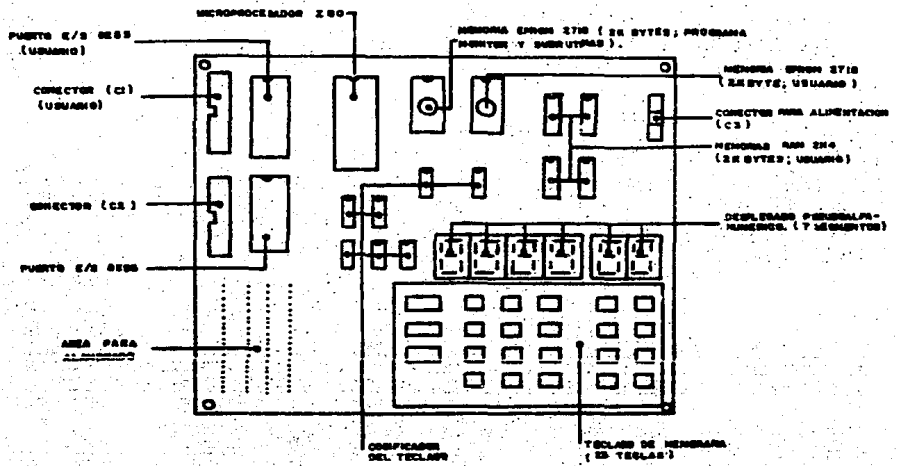


FIGURA 8.1 ESTRUCTURA FISICA DEL MKE-Z8Ø

8.2 TEORIA DE OPERACION

Como cualquier microcomputador, el MKE-Z8Ø está formado por tres partes principales que son:

- Unidad de Procesamiento Central (CPU)
- Memoria
- Dispositivos de Entrada/Salida

Estas tres partes están interconectadas por medio del bus del sistema. Siempre que la CPU desa transferir un dato a la memoria ó a los puertos ó bien pedirlo a la memoria ó recibirlo de los puertos lo hace a través de este bus el cual se divide en:

- Bus de direcciones
- Bus de control
- Bus de datos

A través del primer bus la CPU informa a quien llama; el segundo bus ordena la acción ya sea de leer ó escribir a memoria, puertos ó registros (es decir como); y a través del tercer bus (el qué) información que requiere enviar ó recibir la CPU.

Al instante en que se conecta a la fuente de alimentacion de 5 Volts +5%, la señal de Reset es aplicada a la CPU automáticamente. Los puertos de E/S para inicializarse requiere una señal de un "1" lógico en su Reset (esta es proporcionada oprimiendo la tecla de inicio ó colocando un "0" lógico en la terminal 19 del conector 3. En

forma inmediata el Contador de programa localizado internamente en la CPU apunta a la localidad ØØØØH y por el bus de direcciónes informará a quien le llaman.

Posteriormente la CPU enviará por el bus de control las señales MREQ, RD y M1 en "Ø" lógicos indicando como se llevará a cabo esa acción, esto es, se trata de una llamada a la memoria (Memory Request), es una operación de lectura (Read) y además se debe iniciar un ciclo de búsqueda de intrucción (OP CODE FETCH). Una vez que la instrucción ejecutada por la CPU, el Contador de Programa será incrementado y ahora apuntará a la localidad de memoria siguiente ØØØ1H repitiéndose un nuevo ciclo.

8.2.1 INSTRUCCIONES DE OPERACION

Los commandos del programa monitor del MKE-Z8Ø están divididos en dos clases:

- Comandos 1
- Comandos 2

Los comandos 1 permiten al manejo de la microcomputadora como tal. Mediante estos comandos es posible tener acceso a la memoria, a los registros y a los puertos para examinar y/ó modificar sus contenidos.

Los comandos 2 permiten hacer uso de programas útiles que facilitan algunos trabajos como son: Calcular el valor de un desplazamiento para saltos relativos, obteniendo el resultado tanto hacia atrás como hacia adelante, mover bloques de memoria de una área a otra, comparar dos áreas de memorias, solicitar desplegados automáticos de memoria ó de registros y programar EPROMS.

```
            --- MEMORIA.............(MEM)
COMANDOS 1  --- REGISTROS...........(REG)
  (COM1)    --- ENTRADA/SALIDA.....(E/S)

            --- DESPLAZAMIENTO RELATIVO..(CDR)
            --- MOVER MEMORIA.......(MBM)
COMANDOS 2  --- COMPARAR MEMORIA....(CM)
  (COM2)    --- DESPLEGADO MEMORIA.(DAM)
            --- DESPLEGADO REGISTROS.(DAR)
            --- PROGRAMAR EPROMS...(PM)
```

Los pasos a seguir para utilizar cualquiera de los dos comandos son:

1) Indicar que clase de comando deseamos, tecleando COM1 ó COM2.

2) Indicar que tipo de comando deseamos: MEM, REG, E/S para

COM1 ó CDR, MBM, CM, DAR, ó PM para COM2.

3) Introducir los datos requeridos.

4) Ordenar que se ejecute el comando pulsando la tecla OTRO.

A continuación se dan algunos ejemplos para el manejo del MKE-280.

a) Examinar localidades de memoria.

Teclas	Desplegado
Inicio	HOLA
COM1	CO 1
MEM	dir
1	ØØØ1
4	Ø014
Ø	Ø14Ø
Ø	14ØØ
OTRO	14ØØ XX
OTRO	14Ø1 YY
OTRO	14Ø2 ZZ

b) Modificar localidades memoria.

Tecla	Desplegado
COM1	CO 1
MEM	dir
1	ØØØ1
6	Ø016
C	Ø16C
8	16C8
OTRO	16C8 XX
5	16C8 -5
A	16C8 5A
OTRO	16C9 XX
COM1	HOLA

NOTA: El contenido de la localida de memoria 1400 (RAM) que se indica como "XX" es inprecindible (aleatorio) cuando la energía es aplicada el MKE-Z80, pero una vez que el MKE-Z80 este alimentado no cambia al menos que nosotros lo modifiquemos.

8.3 TECLADO

El Teclado es manejado por un codificador gobernado por programa que funciona de la siguiente manera:

Al hacer una operación INPUT (entrada por un puerto) de la forma IN r, (C) que significa "colocar en el registro r el contenido del puerto direccionado por el registro C", el Z80 pone en la parte baja del bus de direcciones (bits 0-7) el contenido del registro C que sirve para direccionar el puerto deseado, pero en la parte alta del mismo bus (bits 8-F) acomoda el contenido del registro B.

Al colocar un 10H en él (registro B) y al hacer la instrucción IN A, (C0) se pone en un "0" lógico el primer renglón del teclado y si alguna tecla de ese renglón es pulsada, el "0" lógico se transmitirá al puerto de entrada por la terminal correspondiente a la columna de dicha tecla.

Si no hay ninguna tecla oprimida en el primer renglón

el registro B sufre un corrimiento a la Izquierda, quedando ahora con 20H para que sea enviado ahora al "0" lógico en el segundo renglón y así sigue hasta "barrer" los cuatro renglones.

Cada vez que se despliega un digito, se barre todo el teclado en busca de una tecla y en caso de encontrar alguna presionada, el programa se queda en un lazo cerrado, esperando a que esta tecla sea liberada para tomarla en cuenta.

En la fig. 8.2 se muestra el teclado en forma general del MKE-Z80 con sus respectivas funciones.

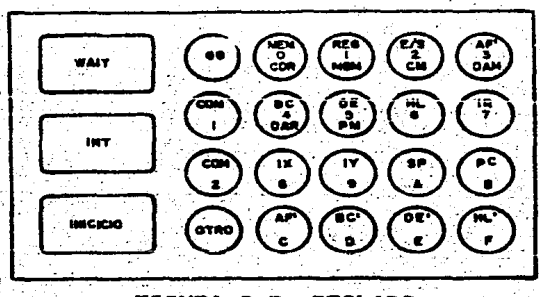


FIGURA 8.2 TECLADO

Especificaciones del teclado

-INICIO : Tecla mediante la cual se proporciona un Reset general al MKE-Z80 al momento de usarla.

-WAIT : Cuando esta tecla es pulsada el procesador detiene el proceso hasta que esta es liberada. Una aplicación útil para esta tecla es: cuando un periférico lento se conecta al MKE-Z80 y no puede mantener la velocidad de comunicación, por lo que requiere solicitar estados de espera mientras procesa los datos que le son enviados. Para este caso existe una terminal de la tecla WAIT presente en el conector 3 mismo que se activa con un "0" lógico.

-INT: Tecla mediante la cual el procesador termina de ejecutar la instrucción en curso y brinca a la localidad 1709H para ejecutar la subrutina de atención a interrupción (SAI). En esta localidad de memoria el usuario debe colocar previamente un brinco al área de memoria donde se encuentra la SAI escrita por él. En una terminal del conector C3 se encuentra la entrada NMI (interrupción no enmascarable) para aquellos casos en que el proyecto requiera del manejo de interrupciones (INT se activa con cero lógico)

- COM1 : Tecla mediante la cual se permite el acceso a memoria, registros de la CPU y puertos de E/S para examinar y/o modificar sus contenido

- COM2: Tecla mediante la cual se permite el uso de programas útiles que faciliten la programación, como son:

+ Desplazamiento relativo

+ Mover bloques de memoria

+ Comparar áreas de memoria

+ Desplegado automático de memoria

+ Desplegdo automático de registros

+ Programar EPROMS

- GO: Tecla mediante la cual se ordena la ejecución de un programa.

- OTRO: Tecla mediante la cual se dá la señal de ejecución para desplegar el contenido de una localidad de memoria ó registro, avanzar a localidades ó registros siguientes, correr un programa, etc.

- Ø al F: Teclas complementarias para la ejecución de COM1, COM2 y GO.

El desplegado pseudo alfanumérico está formado por seis desplegados denominados D3, D4, D5. D6, D7, y D8. Estos son manejados por el puerto Ø1 mediante el empleo de tres de sus ocho canales disponibles a fin de escoger a uno de los seis. Por el puerto Ø2, con el empleo de siete de sus canales, se envián los siete bits que encenderán los segmentos deseados en el desplegado seleccionado por el puerto Ø1.

Esto permite que todos los desplegados (D3-D8), reciban la misma información en un tiempo dado, pero sólo uno de ellos encenderá a la vez. Primero enciende D3 (el digito más significativo cuyo valor hexadecimal es Ø5) con su información, luego se manda encender el digito Ø7 que no existe físicamente y que equivale a apagar a todos los desplegados presentes, a continuación se cambia la información que será el digito Ø4 (D4) y se manda encender este digito. Después se enciende otra vez el digito Ø7 "invisible", se coloca nueva información y se enciende el digito Ø3 (D5), así siempre usando el digito Ø7 para que el cambio de información no se vea y ante nuestros ojos aparezca como si todos los desplegados estuvieran encendidos al mismo tiempo.

El programa monitor tiene un área de memoria para el desplegado cuyas localidades son utilizadas para mantener la información que ha de estar enviándose a cada desplegado, por lo que a un desplegado le corresponde una localidad de memoria que guarda el código de siete segmentos que deberá enviarse al desplegado por medio de los puertos Ø1 y Ø2.

En la fig. 8.3 se muestra una tabla de conversión de código binario a código de siete segmentos utilizada para enviar información a los desplegados

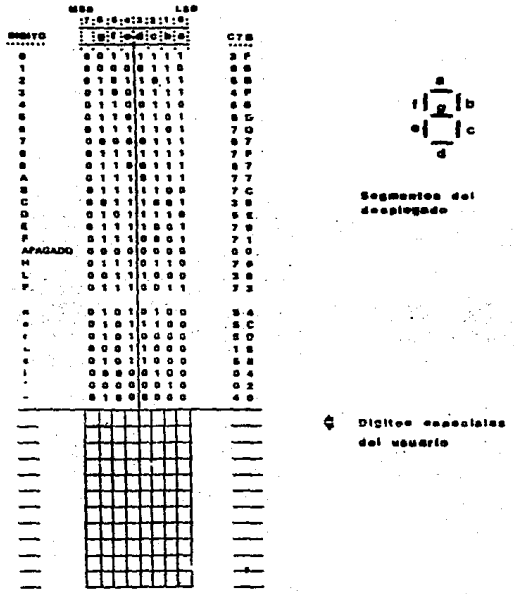


FIGURA 8.3 CONVERSION DE BINARIO A CODIGO DE 7 SEGMENTOS

8.4 AREA DE MEMORIA

La memoria en el MKE-Z80 está dividida en dos áreas:

a) Memoria EPROMS 2716 (borrable y programable de lectura exclusivamente), utilizada para el programa monitor e subrutinas de utileria para poder explotar todos los recursos del microcomputadoras.

b) Memoria RAM 2114 (memoria volátil de lectura y escritura), para ser utilizada por el usuario y una pequeña porción como área de trabajo para el programa monitor. La fig.4 muestra el mapa de memoria del MKE-Z80.

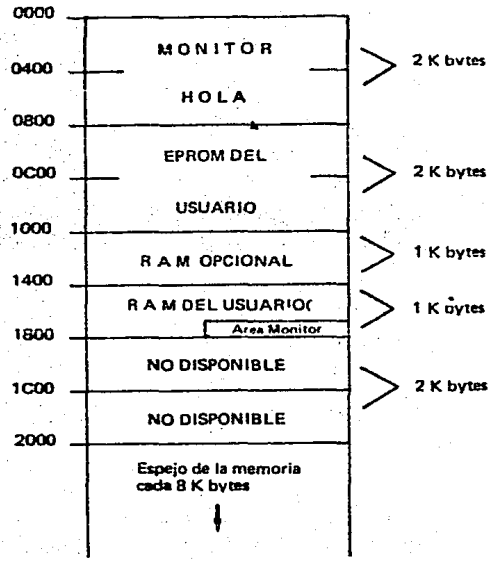


FIGURA 8.4 MAPA DE MEMORIA

8.5 PUERTOS DE ENTRADA/SALIDA

Los puertos utilizados son de E/S en paralelo programables. Estos puertos al igual que la memoria tienen lo que conocemos como decodificación incompleta, esto significa que sólo se utilizan los mínimos bits necesarios para la decodificación tanto en la memoria como de los puertos, lo que se traduce en el hecho de que después de la última localidad (entrada/salida ó memoria) encontremos un espejo de memoria ó un espejo de puertos ya que el bit que define el doble del mapa no es tomado en cuenta para la decodificación.

La decodificación de los puertos de entrada/salida se lleva a cabo con los bits Ø, 1 y 2 del bus de direcciones, lo que admite un máximo de ocho puertos. El MKE-Z8Ø emplea circuitos integrados 8255, cada uno con tres puertos de entrada/salida y uno de control lo cual le permite tener los puertos de entrada/salida (PIOS), mismos que se indican en el mapa de puertos de la fig. 8.5.

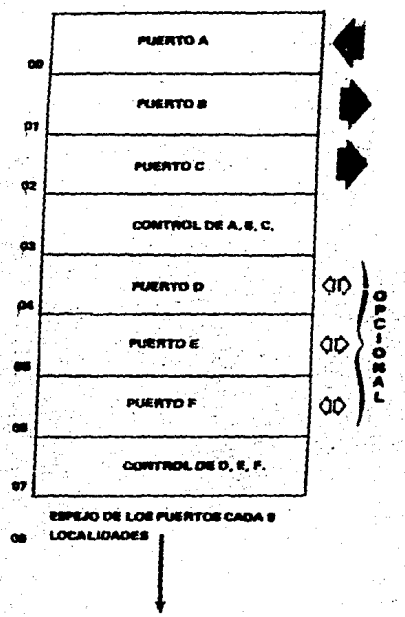


FIGURA 8.5 MAPA DE PUERTOS

8.6 MONITOR HOLA

El programa monitor Hola es el que nos permite comunicarnos con la máquina, este programa que maneja los recursos está contenido en una memoria del tipo EPROM de dos kilobytes que mediante el teclado permite al usuario:

- Examinar localidades de memoria.
- Insertar programas.
- Modificar localidades de memoria.
- Examinar y/o modificar los 22 registros internos del Z80.
- Examinar los puertos de entrada.
- Modificar los puertos de salida.
- Simular solicitudes de "WAIT" al CPU.
- Calcular desplazamientos para saltos relativos.
- Mover bloques de memoria.
- Solicitar desplegado automático de memoria y/o registros internos.
- Correr programas.
- Inicializar el sistema.
- Programar EPROMS.

Este programa monitor es dividido en tres partes:

- Programa principal.
- Subrutinas de utileria
- Tablas.

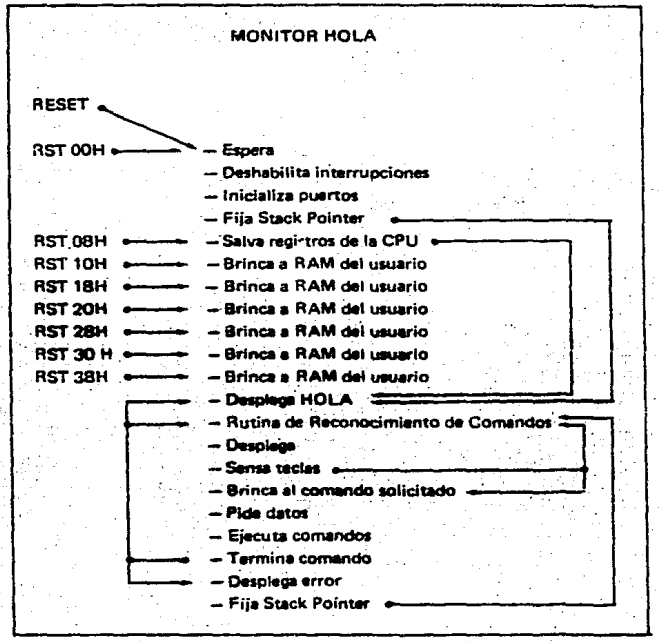


FIGURA 8.6 FLUJO DEL PROGRAMA PRINCIPAL

El programa principal es el que permite al usuario la explotación de la microcomputadora. Las subrutinas de utilería le facilitan el trabajo del programa principal y pueden ser utilizadas por el usuario para hacer más sencillo el trabajo. Las tablas sirven como almacen de datos y direcciones del monitor Hola.

La fig. 8.6 muestra el diagrama de flujo requerido para el programa monitor "Hola".

CAPITULO IX

SISTEMA CROMEMCO

Desde que CROMEMCO lanzó su primer microcomputador en 1975, ha presentado al mercado numerosos modelos que han hecho adquirir a esta firma una reputación de calidad reconocida.

El sistema CROMEMCO TVI-920C, se presenta como un sistema monousuario orientado a aplicaciones profesionales y con excelentes cualidades para trabajar como un sistema especializado.

9.1 DESCRIPCION GENERAL DEL SISTEMA

El sistema CROMEMCO, cuenta con los siguientes elementos de Hardware: una terminal modelo TVI-920C, un módulo para las tarjetas del procesador, memoria y controlador de los diskettes, un módulo para dos controladores de diskettes (DDF).

9.1.1 TECLADO

El teclado, de pequeñas dimensiones y con 101 teclas, incluye todas las funciones de un teclado profesional

expandido, este teclado es del tipo QWRTY e independiente de la unidad central, está constituido por un sólo bloque de teclas dispuestas al igual que un teclado estandar de una màquina de escribir. También cuenta con un teclado numèrico, posee una tecla exclusiva para el movimiento del cursor. Posee además repetición automática, todas las teclas tienen esta posibilidad, excepto las de control, shift y alpha lock.

El teclado cuenta con 11 teclas de funciones especiales, 6 teclas de edición y 2 teclas de transmisión.

Es capáz de generar caracteres en mayúsculas y minúsculas, transferirlas a la CPU, en forma de caracteres ASCII y con bit de paridad para su control.

El teclado numèrico (Keypad), resulta útil en aplicaciones de contabilidad y cálculos financieros y en los que son necesarios la introducción de gran volúmen de datos numericos.

9.1.2 PANTALLA

El monitor estandar es monocromo con un TRC, de 12" y de fósforo blanco sobre fondo negro. Dispone de dos conjuntos de caracteres (mayúsculas y minúsculas), un total

de 96 caracteres ASCII. La resolución es de 24 lineas de 80 caracteres, por lo tanto la capacidad de pantalla es de 1920 caracteres.

El tipo de transmisión puede ser Full ó Half Duplex, cuenta con 9 velocidades de transmición, que son: 75, 110, 150, 300, 600, 1200, 2400, 4800 y 9600 baud. La velocidad es seleccionada por medio de unos interruptores, localizados en la parte posterior de la terminal.

9.1.3 PERIFERICOS

El fabricante ofrece una gran variedad de tarjetas, para la conexión de periféricos como son :

- PRI Controlador de Impresora
- 64KB Memoria RAM Dinámica
- D+7A Convertidor Analógico/Digital y D/A.
- Z80 Microprocesador
- 16FDC Controlador de disco
- TU-ART Puertos de Entrada/Salida en paralelo
- CDDS Controladpr doble de disco
- 32KB Programador de EPROM

La unidad central se basa en el microprocesador Z-80A de 8 bits.

La zona de memoria RAM, destinada al usuario es de 64 Kbytes, no se dispone de la ampliación que permita aumentar la capacidad de almacenamiento interno en RAM.

La comunicación con el exterior se realiza a través de dos interfaces, una de tipo RS-232C, que es para control de la consola (pantalla y teclado) y una tipo paralelo (PRI), que es la que controla la impresora, lo que permite una mayor versatilidad a la hora de configurar un puesto de trabajo.

9.2 SISTEMAS OPERATIVOS

9.2.1 DEFINICION

Para conseguir un uso más racional y un mejor aprovechamiento de las microcomputadoras, se ha desarrollado una serie de programas que constituyen el software funcional, los constructores de sistemas lo suministran bajo diversos nombres, aunque el más general es el de sistemas operativos.

Sistema operativo, es la colección ordenada de rutinas y procesamientos que acompañan a la microcomputadora y que normalmente realizan todas ó algunas de las siguientes funciones:

- Planificación de memoria, unidades de entrada/salida

y otros dispositivos.

- Gestión de memoria.
- Tratamiento de errores.
- Coordinación de las comunicaciones entre sistema usuario.
- Mantenimiento de un registro con las operaciones del sistema.
- Inicialización y control de todas las operaciones de E/S.
- Control de las operaciones en los trabajos de multiprogramación, multiproceso y tiempo compartido.

En resumen, el sistema operativo es el conjunto de los programas del sistema que permite al usuario utilizar el sistema comodamente y que optimiza su rendimiento. Una característica fundamental del sistema operativo, es que debe incluir un programa monitor, que controle la ejecución de los demás programas y mantega el funcionamiento del sistema sin intervención del operador más que en caso de necesidad.

Dentro de los sistemas operativos se distinguen cinco tipos principales, que son:

- Secuencial por Lotes: Permite ejecutar los trabajos uno a uno. Los programas pueden ejecutarse, nada más ser introducidos ó memorizarse en dispositivos de acceso rápido,

ejecutándose secuencialmente más tarde.

- Multiprogramación: Permite que varios trabajos se ejecuten simultáneamente, se consigue mediante el uso de las interrupciones.

- Tiempo Real: Permite el uso de sistemas por varios usuarios que utilizan terminales remotas y efectúan constantemente operaciones de entrada y salida de datos.

- Tiempo Compartido: Permite a muchos usuarios utilizar el mismo sistema, que aparentemente solo está dedicado a cada uno de ellos, ya que cada usuario recibe el control de la CPU, durante un determinado intervalo de tiempo.

- Memoria Virtual: El sistema operativo asume responsabilidades de gestión de la memoria principal.

MONITOR O SUPERVISOR

El monitor ó supervisor debe estar presente siempre en la memoria; a veces sólo reside en la memoria central ó una parte de él, en la llamada residente, el resto es llamado cuando se necesita. El supervisor contiene, en general, todos los subprogramas que realizan las funciones básicas. El sistema supervisa la actividad del programa rechazando las operaciones no válidas y evitando de esta manera la detención del sistema por errores del programa del usuario.

Los elementos del monitor son: Control de trabajos,

Control de E/S, Comunicaciones y Recuperación del Sistema.

El control de trabajos comprende las funciones que controlan y regulan el uso de los recursos del sistema y en particular de la planificación de trabajos, de la asignación, de carga y de las terminales remotas.

El control de E/S, regula las actividades de los dispositivos de E/S, comprende: la planificación de los recursos de las entradas y salidas, la transferencia de datos y el soporte de las terminales remotas.

El sistema de comunicaciones se responsabiliza de los intercambios de información entre el sistema operativo y los usuarios, cuando un error impide la continuación normal de un trabajo, intervienen rutinas de recuperación, que permiten la reanudación de un trabajo a partir de un determinado punto de proceso.

9.2.2 RDOS

El sistema operativo incorporado en la versión básica, es el RDOS (Resident Disk Operating System) incluye un repertorio de instrucciones y funciones que se encuentran permanentemente decodificadas por la Unidad Central del sistema.

Permite la introducción de comandos que son reconocidos y ejecutados por la Unidad Central de Proceso. El RDOS, es residente en la memoria ROM del sistema.

Todos los comandos del RDOS, deben terminar para que sean reconocidos como tales, con una acción sobre la tecla RETURN. Los comandos más significativos son:

- AS,BS: Utilizados para seleccionar una entre dos unidades de disco flexible (la A ó la B).
- DM: Para la visualización del contenido de la memoria.
- G: Utilizado para ordenar el comienzo de la ejecución de un programa.
- M: Comando para el desplazamiento de zonas de memoria.

9.2.3 CDOS

El Sistema Operativo de CROMEMCO CDOS (CROMEMCO Disk Operating System), es un producto diseñado y escrito originalmete en código de màquina de Z80 por la compaK9ia CROMEMCO, para su propia línea de microcomputadoras. Sin embargo debido al gran número de programas que actualmente se pueden conseguir para correr bajo el sistema operativo CP/M, CDOSDS fue diseñado para ser compatible con CP/M. La función principal de CDOS es la de controlar la entrada/salida a ó de

los dispositivos de almacenamiento masivo, tales como el disco duro ó el floppy.

Además está diseñado para que los usuarios del sistema CROMEMCO puedan generar y manipular archivos del tipo secuencial y aleatorio utilizando nombres simbólicos.

Por otro lado CDOS contiene también las rutinas de manejo de la impresora, así como un gran número de llamadas para el manejo de pantalla, por medio de lo cual se pueden hacer gráficas, manejar una linea de mensaje de pantalla, etc.

El sistema CDOS, reside en disco y puede ser invocado por RDOS, de dos formas: La primera es llamada a CDOS desde RDOS, con el comando B y RETURN; la segunda entra automáticamente, cuando CDOS tiene el auto-boot ya en disco.

Tanto el RDOS, como el CDOS, tienen entre sus funciones la verificación de las operaciones que se realizan, y son capaces de presentar en pantalla mensajes de error si alguna de estas temina ó se ejecuta incorrectamente.

Debido a la total compatibilidad del sistema operativo estandar, incorporado en este sistema CROMEMCO, con el CP/M, el volumen de aplicaciones disponibles en el mercado

para este sistema es muy grande. No obstante, el fabricante ofrece un potente software de desarrollo propio, orientado a aplicaciones financieras y de tratamiento de texto.

9.3 DISTRIBUCION DE LA MEMORIA

La memoria de la microcomputadora bajo CDOS está dividida en dos grandes partes. La primera parte es una parte de memoria RAM que CDOS se reserva para él mismo. CDOS ocupa la memoria de la localidad ØØØØH a la Ø1ØØH (parte baja) así como también se apropia aproximadamente de 11K a 18K de la parte alta de la memoria RAM. La segunda parte es la memoria RAM del usuario, la memoria del usuario ocupa la memoria de la localidad Ø1ØØH a la parte alta de la misma hasta donde comienza la parte alta de CDOS.

La longitud del área de memoria depende del sistema operativo que se esté utilizando, y éste a su vez del número de controladores de disco que se tengan declarados en el sistema operativo, si se utilizan las teclas de funciones para la terminal, etc., generalmente esta área es de 48K aproximadamente.

La fig. 9.1 muestra un mapa de memoria, con el sistema operativo.

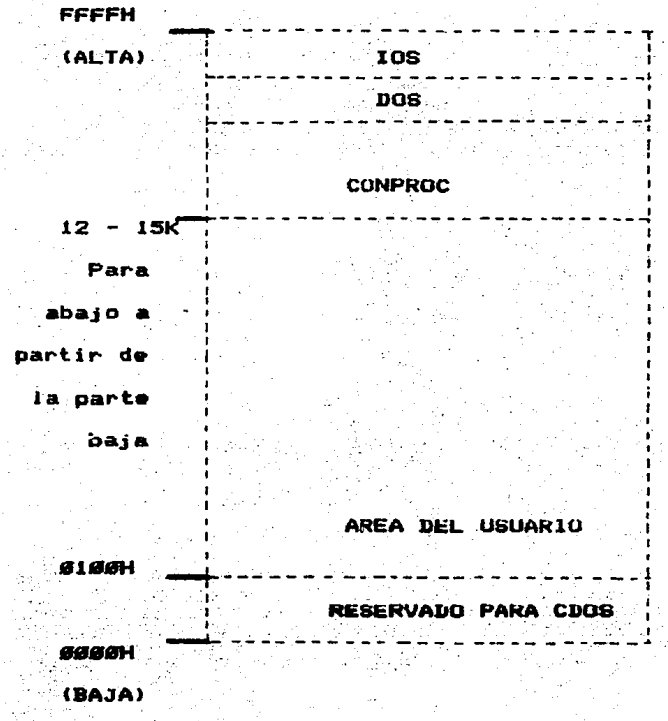


FIGURA 9.1 MAPA DE MEMORIA

Refiriendose al mapa de memoria, podemos ver que ésta se encuentra dividida en tres grandes partes, que son las siguientes:

a) PARTE ALTA DE MEMORIA.

b) MEMORIA DEL USUARIO.

c) PARTE BAJA DE MEMORIA.

MEMORIA DEL USUARIO: Es en esta parte donde los programas corren.

PARTE BAJA DE MEMORIA: Esta parte de la memoria esta reservada por CDOS para las siguientes funciones:

0000H - 0002H	Vector del sistema Warm Start.
0003H	Byte de E/S.
0005H - 0007H	Vector de llamadas a sistema para preguntas del usuario.
0008H	Especifica si se está corriendo bajo CDOS, si se tiene FFH y bajo CROMIX, si se tienen C3H.
0030H - 0032H	Puntos de ruptura para DEBUG.
0033H - 003AH	Salto al mensaje de: "INVALID JUMP".
0040H - 005BH	Reservada para el sistema.
005CH - 007BH	File Control Bloks estandares para el usuario.
0080H - 00FFH	Buffer estandar de Entrada/Salida al disco.

CDOS es cargado al área del sistema desde el disco por medio de la rutina de BOOTSTRAP. Si el usuario requiere de alguna configuración diferente de hardware (periféricos) se deberá de generar un nuevo sistema operativo, por medio de "CDOSGEN" (CROMEMCO Disk Operating System Generator) el cual nos permite definir la configuración que se desee y genera un

sistema operativo distinto.

Para poder hacer uso tanto de las utilerias, como de los paquetes debemos de estar bajo control del sistema operativo de CROMEMCO, lo cual se reconoce por que pone en la pantalla un "PROMPT" ó señal de control que nos indica que podemos darle cualquier comando válido. Este control es una letra seguido de un punto, la letra nos indica en que controlador de disco estamos trabajando.

9.4 SOFTWARE DEL SISTEMA

9.4.1 ARCHIVOS

Un nombre de archivo en CROMEMCO está formado de tres partes:

(X:) NOMBRE (.EXT)

La primera parte es un identificador de "DRIVE", especifica en que drive "A" ó "B" está colocado el diskette que contiene el archivo NOMBRE.

La segunda parte es el nombre del archivo (hasta 8 caracteres), el cual puede estar formado por letras, números y cualquier caracter ASCII excepto los siguientes: * * ? = / . , : espacio.

La tercera parte es la extención de este nombre de archivo, consta de un punto y tres caracteres, esta extención sirve para identificar distintos tipos de archivos, algunas extensiones son:

.BAK Archivo de respaldo del Editor

.BAS Archivo fuente de BASIC

.CDM Archivo de comandos "Batch"

.COB Archivo fuente de COBOL

.COM Programa ejecutable de ensamblador

.FOR Archivo fuente de FORTRAN

.HEX Archivo objeto con formato hexadecimal

.PRN Archivo de listado o impresora

.REL Módulo relocalizable

.TXT Archivo de texto

.Z80 Archivo fuente de ensamblador

.BIN Archivo Binario

9.4.2 CONTROLES ESPECIALES DEL TECLADO

Para el manejo adecuado tanto de la terminal como de las impresoras existen una serie de controles especiales, los cuales son ejecutados a través de la consola, oprimiendo la tecla de "Control" y la letra que designe el control necesario a ejecutar. A continuación se da una lista de

estos controles.

CNTRL-S Con este control se puede detener un listado que esté apareciendo en la pantalla, para continuar el listado se oprime cualquier tecla.

CNTRL-K Mueve el cursor hacia arriba.

CNTRL-L Mueve el cursor hacia la derecha.

CNTRL-J Mueve el cursor hacia abajo

CNTRL-H Mueve el cursor hacia la izquierda

Los siguientes controles son para el manejo de la impresora.

CNTRL-L Envia un salto de página a la impresora

CNTRL-P Prende la opción de la impresora, esto es, todo lo que aparece en la pantalla saldrá también hacia la impresora.

CNTRL-T Apaga la opción de salida de la impresora. Este control debe de ser enviado por medio de algun programa del usuario, si se utiliza desde la terminal no tiene ningún efecto.

CNTRL-W Este control también prende la opción de salida a impresora, la diferencia es que éste debe ser enviado desde un programa del usuario, para habilitar la salida a la impresora.

9.4.2 INTRINSECOS DEL SISTEMA

Los intrinsecos del sistema son un conjunto de comandos, para diversos usos, que contiene el sistema operativo bajo el cual se está trabajando.

DIRECTORIO DE UN DISCO "DIR"

Para poder ver el contenido de un disco dado, sólo se teclea el comando "DIR" e inmediatamente será desplegado en la pantalla el contenido del directorio indicando en forma tabular la siguiente información:

- El nombre del archivo
- La extensión del archivo (calificativo) si existe
- Tamaño en caracteres entre un factor de 1024 (Kilobytes).

Al final del directorio da una linea en la cual se indica cuanto espacio hay ocupado en el disco, cuanto espacio disponible queda en éste, así como cuantos archivos tiene el disco.

BORRADO DE ARCHIVOS "ERA"

Para borrar uno o más archivos de un disco se utiliza el intrínseco del sistema "ERA". Para borrar un archivo llamado LISTA.COM sólo se tendrá que teclear:

ERA LISTA.COM

También se pueden borrar una serie de archivos por medio de la referencia ambigua de archivos o de caracteres de reemplazo. Así mismo se pueden borrar todos los archivos que contiene un disco de la siguiente manera:

ERA *.*

CAMBIO DE NOMBRE "REN"

Por medio de este intrínseco del sistema se puede cambiar el nombre a un archivo, (sin alterar su contenido). La forma de cambiar el nombre a un archivo es el siguiente: primero teclear "REN" seguido del nuevo nombre del archivo, un signo de igual "=", y el nombre de archivo que tiene actualmente. Por ejemplo cambiar el nombre a un archivo que se llama "LISTA". TXT", que se llame "LINEA".TXT", teclear lo siguiente:

REN LINEA.TXT = LISTA.TXT

Con lo cual el archivo aparecerá en el directorio del

disco con el nombre de "LINEA.TXT"

PROTECCION DE MEMORIA "SAVE"

"SAVE" permite guardar en disco partes de memoria, que puede accesar el usuario, esto es de la localidad de memoria 100H hacia arriba y dependiendo el número de páginas que se le especifiquen. Este comando es útil para recuperar archivos que el programa encadenador generó en memoria. Como ejemplo, guardar 5 páginas de 256 bytes de memoria en el disco y con el nombre de ARMA.COM:

SAVE ARMA.COM 5

La extención de éste tipo de archivos debe de ser "COM" y será un archivo ejecutable de comandos.

LISTAR UN ARCHIVO "TYPE"

Para listar un archivo en la pantalla teclear:

TYPE ARMA.TXT

Donde ARMA.TXT es el archivo a listar en la pantalla. Para detener un listado que está siendo desplegado en la pantalla se deberá de teclear CNTRL-S con lo cual se detendrá, y para continuar el listado se teclea otra vez CNTRL-S. También se puede mandar el listado a la impresora

con el comando CNTRL-P. Además si el programa contiene el comando CNTRL-W el listado saldrá directamente a la impresora.

9.4.4 UTILERIAS DEL SISTEMA

Las utilerías son una serie de programas que se entregan con el sistema de cómputo, los cuales ofrecen facilidades para el manejo de archivos y de dispositivos periféricos, que contine.

PROCESO EN "BATCH" @

La utileria de "Batch" o proceso en lote permite ejecutar una serie de comandos válidos para CDOS secuencialmente. En forma inmediata o en forma de archivo de comandos, estas dos opciones se describen a continuación:

1.- Forma de ejecución inmediata. Si se tiene que ejecutar una serie de comandos una sola vez se puede utilizar éste en forma de proceso en lote.

2.- Si se tiene una serie de comandos los cuales se tienen que ejecutar continuamente, se puede crear un archivo con estos comandos, por medio de alguno de los editores del sistema y ejecutarlos cuando se requiera. Este archivo debe de contener un comando por línea y además el nombre de este archivo debera de tener la extensión de "CMD".

ANALIZAR UN ARCHIVO "DUMP"

Este programa es útil para analizar el contenido de un archivo que no se puede listar por medio del comando "TYPE". En este caso podemos obtener un listado en hexadecimal de cualquier archivo, de la siguiente manera:

DUMP TESIS.TXT

Lo cual dará como resultado un listado en Hexadecimal con su correspondiente caracter en ASCII, del archivo llamado TESIS.TXT. Este listado estará compuesto de registros de 128 bytes.

OBTENCION DEL ESTADO DE DISCO "STAT"

Por medio de esta utilería, se pide el informe acerca del estado de algún disco ó del sistema, también se puede modificar ciertas variables del sistema. La forma de utilizar esta utileria es la siguiente: Primero teclear "STAT" seguido de una diagonal y una ó más letras las cuales denotan la opción de la utilería que se quiera ejecutar, además también se puede especificar otro "Drive" distinto del cual se ejecutó la utileria, poniendo la letra del "Drive" que se desee obtener información. A continuación se enuncian algunas de las opciones con que cuenta esta utilería:

/a Despliega en orden alfabético los archivos que contiene el disco, con el espacio utilizado de cada uno de ellos.

/b Da una breve descripción del espacio utilizable del disco.

/d Por medio de esta opción podemos poner la fecha al sistema.

/e Permite borrar archivos selectivamente.

/l Con esta opción se etiqueta un disco.

/t Por medio de esta opción se pone la hora del dia al sistema, la cual aparecerá en la linea de estado del sistema.

Además se puede teclear sólo la palabra "STAT" y se obtendrá un informe detallado del disco y del sistema.

9.5 CARACTERISTICAS DE HARDWARE DEL SISTEMA

A continuación se describen en forma general algunas de las tarjetas, ya mencionadas anteriormente, que componen la arquitectura básica del sistema CROMEMCO. Todas las

tarjetas son compatibles con el bus S-100.

Este sistema tiene una gran expansión en la actualidad, pues cada dia son más periféricos, asi como paqueteria que cada dia sale al mercado para esta máquina.

9.5.1 ZPU UNIDAD CENTRAL DE PROCESAMIENTO

- PROCESADOR

Se utiliza el Z-80A, que es una version del Z80, pero que es capaz de utilizar un reloj de 4MHz. Este procesador tiene las siguientes caracteristicas:

Registros de 8 bytes:	16
Registros de 16 bytes:	10
Registros de indice:	2
Modos de dirección:	10
Dirección de E/S:	256
Bits de bandera:	6
Voltaje requerido:	+5V
Velocidad de reloj máxima:	4MHz
Compatible con TTL:	Si
Modos de interrupción:	4
Refrescamiento de memoria automático:	Si
Velocidad relativa en ejecución:	1.0

- ESPECIFICACIONES TECNICAS DE LA TARJETA DEL ZPU

Frecuencia de reloj:.....Versión en 4MHz

Conjunto de instrucciones:.....158 incluyendo las 78 del 8080

Salto automático al encendido:....Habilitado por Hardware

Direcciones posibles al salto automático:.....16 direcciones seleccionables por interruptores

Generación de estados de espera:.....0-4 estados de espera seleccionables por Hardware

Ciclo de espera de máquina:.....Seleccionables por Hardware

Compatibilidad del BUS:.....S-100

Necesidades de alimentación:.....+8 V DC y 1.1 Amp.

Rango de operación:.....0-55 grados °C.

- SALTO AUTOMATICO DE MEMORIA AL ENCENDIDO

El circuito de salto automático instalado en la tarjeta del ZPU, permite a ésta el ser usada en un sistema de BUS S-100 sin controles en los paneles externos de la microcomputadora. Cuando el sistema es encendido, el

hardware del ZPU forza un salto automático a una de las 16 posibles localidades anteriormente seleccionada. En la dirección que se seleccione con los interruptores, se encuentra el sistema operativo residente en ROM (RDOS).

El salto automatico a las direcciones de memoria correspondiente a cada codificación de los interruptores, se tabula a continuación:

INTERRUPTOR				Cuando se enciende
A15	A14	A13	A12	SALTA A:
Ø	Ø	Ø	Ø	ØØØØH
Ø	Ø	Ø	1	1ØØØH
Ø	Ø	1	Ø	2ØØØH
Ø	Ø	1	1	3ØØØH
Ø	1	Ø	Ø	4ØØØH
Ø	1	Ø	1	5ØØØH
Ø	1	1	Ø	6ØØØH
Ø	1	1	1	7ØØØH
1	Ø	Ø	Ø	8ØØØH
1	Ø	Ø	1	9ØØØH
1	Ø	1	Ø	AØØØH
1	Ø	1	1	BØØØH
1	1	Ø	Ø	CØØØH
1	1	Ø	1	DØØØH
1	1	1	Ø	EØØØH
1	1	1	1	FØØØH

Cuando se carga el sistema operativo una señal, deshabilita la memoria ROM (en donde está RDOS) para obtener 64K de memoria RAM (65,536 bytes de memoria de acceso aleatorio).

- SELECCION DE FRECUENCIA DEL RELOJ

El microprocesador Z-80A puede trabajar con un reloj de 4MHz (con un ciclo de tiempo de 250 nseg.) ó 2MHz (con un ciclo de tiempo de 500 nseg.). La frecuencia de operación es seleccionable por medio de un interruptor tipo "toggle", la línea del bus S-100 previamente etiquetada como "STCK" es utilizada por el ZPU como una línea indicadora de 4 MHz.

- SELECCION DE ESTADOS DE ESPERA

El ZPU prevee un generador de estados de espera en la misma tarjeta, de tal manera que son compatibles la frecuencia del reloj del Z-80A y el tiempo de acceso al sistema de memoria que se tenga.

El ZPU permite la inserción de dos tipos de estados de espera duruante cada ciclo de máquina.

- SELECCION DE MEMORIA ESPEJO

El microprocesador 8080 repite ó espejea la dirección de los 8 bits del puerto de Entrada/salida en el orden alto y bajo de la dirección del BUS. A pesar de que esta característica no pertenece al procesador Z-80A, la tarjeta del ZPU está diseñada para imitar esta característica a través de una circuitería especial, asegurando que será compatible con un sistema que se actualice de 8080 al Z-80A.

- REFRESCAMIENTO DE MEMORIA

Todos los tipos de tarjetas de memoria dinámica requieren un refrescamiento de las direcciones propocionadas por el Z-80A reflejadas en las líneas de las direcciones de órden alto.

9.5.2 16FDC

El 16FDC es un controlador de disco, esta tarjeta está provista de un sistema completo para la operación de los floppies, que pueden ser indistintamente de 8 pulgadas y 5 pulgadas. Incluye un puerto serie de Entrada/Salida para una terminal de video con interfaz RS-232, una memoria pre-programada ROM, en donde se encuentra grabado un programa monitor llamado RDOS (Resident Disk Operating System), por

medio del cual se inicia el sistema (boot), así mismo posee comandos por medio de los cuales se realiza un diagnóstico básico de la memoria y drives. Este monitor normalmente funciona sólo cuando se enciende la máquina, ó despues de un reset. Con este monitor se inicia el sistema y en este momento se desconecta el monitor transfiriéndose el control al sistema operativo.

El 16FDC normalmente maneja hasta 4 drives en cadena de prioridades; sin embargo hasta 16 drives pueden ser encadenados si es necesario.

9.5.3 64KZ

La tarjeta CROMEMCO 64KZ es una memoria dinámica para Lectura/Escritura de 65,536 bytes (64K bytes), compatible con el bus S-100. esta tarjeta incorpora una memoria dinámica RAM, para esto utiliza un circuito integrado llamado: TMS 4116-15 (16 x 1 bit). Opera en un tiempo de acceso máximo de 250 nseg. Esto significa que la 64 KZ puede operar en los sistemas Z80 con un reloj de 4MHz sin utilizar estados de espera en lo absoluto. Esta tarjeta maneja la señal de selección de banco, es posible el aumentar lógicamente la memoria, pues se puede intercambiar bancos de memoria por software, esta es una opción muy importante, pues se utiliza grandemente en sistemas operativos para multiusuarios. Esta

tarjeta ofrece las siguientes características:

- 64K bytes de memoria para Lectura/Escritura (R/W) en una sola tarjeta de memoria para el bus S-100.
- Un tiempo máximo de acceso de 250 nanosegundos.
- Organización en dos bancos independientes de memoria de 32K (A y B).
- Selección del banco, permitiendo una expansión de la memoria más alla de los 64K bytes.
- Compatibilidad con el Z80 ó el 8080.

Especificaciones técnicas:

Requerimientos de potencia:	+8 Volts y 1.8 ampers(max)
	+18 Volts y .45 ampers(max)
	-18 Volts y .03 ampers(max)
Tipo de memoria:	TMS 4116-15,16xi RAM dinámica
Estados de espera a 4MHz:	No se requieren
Estados de espera a 2MHz:	No se requieren
Compatibilidad con bus:	S-100

9.5.4 D+7A

La tarjeta D+7A se utiliza como Entrada/Salida de señales analógicas y digitales. Contiene 7 canales de conversión analógico-digital y 7 canales de conversión

digital-analógico ambos con una resolución de 8 bits y un tiempo de conversión de 5.5 microsegundos.

Compatible con el Bus S-100. La tarjeta cuenta con 8 puertos de entrada/salida, 1 digital y 7 analógicos, la dirección de estos puertos se selecciona por medio de cinco conectores en la tarjeta.

Especificaciones técnicas:

- Puertos de entrada:	7 analógicos.
	1 digital en paralelo
	8 bits.
- Puertos de salida:	7 analógicos.
	1 digital en paralelo
	8 bits.
- Voltaje de entrada:	-2.56 a +2.54 Volts.
- Voltaje de salida:	-2.56 a +2.54 Volts.
- Impedancia de entrada:	20 Megaohms.
- Impedancia de salida:	0.25 ohms.
- Exactitud:	+- 20 milivolts.
- Máxima corriente de carga:	1.50 mA.
- Tiempo de conversión:	5.5 microsegundos.
- Alimentación:	+8 Volts a 0.4 A.
	+18 Volts a 30 mA.
	-18 Volts a 60 mA.

La asignación de los puertos es:

	Dirección
Puerto digital	18H
Puerto analógico 1	19H
Puerto analógico 2	1AH
Puerto analógico 3	1BH
Puerto analógico 4	1CH
Puerto analógico 5	1DH
Puerto analógico 6	1EH
Puerto analógico 7	1FH

CAPITULO X

SOFTWARE

Una computadora no es un solucionador de problemas independiente. De cualquier forma, debido a la velocidad con la que puede recuperar y manipular grandes volúmenes de datos, la computadora es una ayuda esencial en el proceso de solución de problemas. El proceso de solución se presenta a una computadora en forma de programas, una lista de las acciones requeridas para llegar a los resultados.

Este capitulo analiza las fases del diseño del programa y su implantación.

SOFTWARE : Es un soporte de programación de un sistema de procesador, ó como un conjunto de programas utilizables por un determinado sistema de procesador. Con frecuencia al software del sistema se le hace referencia en forma breve como software.

10.1 SOLUCION DE PROBLEMAS CON COMPUTADORAS

La solución de un problema dado, siempre corresponde al programador. Si elegimos incorporar una computadora en este trabajo, se puede distinguir las siguientes etapas en el

proceso de solución de problemas:

1) Definición del problema: La solución que estamos buscando por lo general tiene que aplicarse más bien a una clase de problemas que a un problema sencillo. Para definir esta clase, se tienen que hacer suposiciones con respecto a la información disponible (se diseñarán las entradas al programa). Se debe establecer también la naturaleza de los resultados deseados (las salidas del programa para todas las clases de entradas posibles).

2) Análisis del problema: Se debe determinar la aproximación a la solución del problema más efectiva y eficiente, puede ser posible dividir el problema en subproblemas. Se debe investigar la posibilidad de incorporar programas ya existentes.

3) Diseño de la estructura de datos: Se determina la composición de los datos para ser manejados por el programa. Esto se hace junto con el paso siguiente:

4) Diseño del algoritmo: El algoritmo ó procedimiento de solución para el problema, se debe de escribir con el uso de una notación algoritmica. Estas notaciones evitan los detalles contenidos en programas, permitiendo al solucionador de problemas, concentrarse en el propio problema.

5) Codificación: Se debe seleccionar un lenguaje de programación apropiado. Se debe escribir un programa legible con estructura clara.

6) Implementación del programa: El programador debe de estar convencido, que usando la computadora y los datos de entrada de la muestra representativa, el programa funciona como él lo desea. Se debe investigar el comportamiento del programa en respuesta a todas las posibles variaciones de las entradas durante la definición del problema y las etapas de análisis.

7) Documentación del programa: Una descripción de la operación del programa, es su estructura de datos, y se debe proporcionar instrucciones de entrada y de salida.

8) Uso del programa y modificaciones posibles: El programa está operando en muchos casos periódicamente para obtener los resultados necesarios. Es una herramienta y quizá sea mejorada.

Para resolver un problema con el uso de una computadora, un programador necesita diseñar un algoritmo, un procedimiento inequivoco que especifica un número finito de pasos que se deben tomar. Cada algoritmo opera sobre ciertos

datos que describen los objetos del mundo real concernientes al problema. Los pasos del algoritmo manejan estos datos.

El uso de datos de tipo apropiado, organizado, origina algoritmos más claros y sencillos. Después de que se ha diseñado al algoritmo y las estructuras de datos usadas por este, se puede crear un programa en el lenguaje de programación seleccionado.

Los algoritmos consisten de pasos que manejan los datos y las estructuras de control que especifican la secuencia en las que se llevan a cabo los pasos. Los pasos básicos del manejo de datos son la asignación del valor a una partida de datos con valores variables y la entrada y salida de datos. Estos pasos pueden ser llevados a cabo, secuencial, repetida ó condicionalmente, según sean dirigidos por las estructuras de control.

Los subalgoritmos ó subprogramas pueden ser llamados para realizar una subtarea específica en la tarea completa de solución del problema. Por tanto, un algoritmo y el programa que se realiza a partir de éste se pueden construir en varios módulos. La programación modular hace posible diseñar programas complejos de una manera ordenada, con un grado alto de confiabilidad y claridad.

Para especificar algoritmos, se puede emplear seudocódigos ó diagramas de flujo. Cuando se usa el seudocódigo, el programador especifica los pasos del algoritmo usando básicamente el lenguaje natural con estructuras de control sobrepuestas. En consecuencia, se introduce más precisión en esta especificación. Los diagramas de flujo son herramientas gráficas tradicionales con símbolos estandarizados.

Ambas herramientas se pueden usar para especificar un algoritmo, de una manera de arriba a abajo mediante la técnica de refinar los pasos. Así, el programador refina una idea inicial de la solución del problema en términos cada vez más específicos mientras identifica los módulos que realizarán subtareas particulares. Al mismo tiempo, la definición de los datos se vuelve más especifica. El último refinamiento del algoritmo será codificado para su ejecución en la computadora.

10.2 ALGORITMOS Y SU EXPRESION

La etapa vital de la solución de un problema con una computadora es el diseño del algoritmo y de la estructura fundamental de datos. Un algoritmo es un procedimiento expresado precisamente para obtener la solución del problema, la que se presenta de manera subsecuente a una computadora en

el lenguaje de programación seleccionado. Los algoritmos se presentan de una manera conveniente para un lector humano, mientras los programas sirven a las necesidades de las computadoras.

Es importante recordar mientras diseñamos un algoritmo que una computadora sólo sigue las instrucciones y no puede actuar si no se le ha ordenado de una manera explícita. Por tanto, el solucionador de problemas debe prever cualquier aspecto del problema en el propio algoritmo

Este capítulo especifica las propiedades que deben tener los procedimientos para la solución de problemas si van hacerse por algoritmos. Estas son: la no limitación, ausencia de ambiguedad, definición de la secuencia, definición de entrada y de salida, efectividad y definición del alcance. Los diagramas de flujo se presentan como una herramienta para la expresión de los algoritmos. Más adelante se muestra otra herramienta, con frecuencia más útil, para esta tarea: el seudocódigo.

10.3 DEFINICION DE ALGORITMOS

Todas las tareas que puede llevar a cabo una computadora se pueden expresar como algoritmo. Una vez que se ha diseñado un algoritmo, se codifica en un lenguaje de

programación, y el programa se realiza en una computadora.

Un algoritmo es un conjunto finito de instrucciones que especifican una secuencia de operaciones a realizar en orden para resolver un programa específico ó clase de problema. En otras palabras, un algoritmo es una fórmula para la solución del problema.

Un algoritmo se puede presentar en varios niveles de detalle. El hardware de una computadora sólo puede obedecer las instrucciones si están expresadas en un lenguaje de máquina de la computadora. El diseñador del algoritmo puede encontrar dificultades al pensar en términos de estas instrucciones, ya que los detalles pueden oscurecer la escencia del procedimiento.

10.4 PROPIEDADES DE LOS ALGORITMOS

Un procedimiento que no tiene las propiedades que a continuación enunciamos no es un algoritmo y generalmente no proporciona el resultado deseado cuando un programa basado en él se presenta en una computadora.

10.4.1 LIMITACION

La ejecución de un algoritmo programado se debe

completar después de que se haya llevado a cabo una cantidad finita de operaciones. De otra manera, no podemos exigir que la ejecución produzca una solución:

(a) El número actual de pasos depende de la minuciosidad (grado de detalle) de la presentación de un algoritmo.

(b) El número de operaciones realizadas raramente es igual al número de pasos en la descripción del algoritmo (ó al número de instrucciones en el programa). El número de pasos realizados en la actualidad durante la ejecución de un programa depende de la entrada de datos y no siempre puede ser determinado de antemano.

(c) Un algoritmo dirigido a terminar un programa en 100 años, es deficientemente útil.

La escencia del método del algoritmo consiste en la repetición del mismo paso ó pasos, probablemente con algunas modificaciones, muchas veces durante la realización de un programa basado en un algoritmo dado. La longitud del programa es, por tanto, un indicador pobre del tiempo de ejecución.

10.4.2 AUSENCIA DE AMBIGUEDAD

La representación de cada paso de un algoritmo debe

tener una representación única, aunque una representación para una computadora puede diferir para la de un humano. Es conveniente para los humanos tratar con algoritmos presentados en una notación con detalles separados (por ejemplo un seudocódigo ó una carta de flujo) mientras que la computadora necesita que el algoritmo sea codificado en un programa.

Esta codición significa que cada vez que se presente para su ejecución un algoritmo con los mismos datos de entrada, se obtendrán los mismos resultados.

10.4.3 DEFINICION DE LA SECUENCIA

Se debe especificar sin lugar a duda la secuencia en la que deben llevar a cabo los pasos del algoritmo. Un algoritmo debe tener una instrucción inicial única y cada instrucción debe tener un sucesor único para un dato de entrada dado.

En las especificaciones del algoritmo, incluyendo los programas, las instrucciones son llevadas a cabo de arriba hacia abajo, a menos que las instrucciones por si misma especifiquen otra cosa.

10.4.4 DEFINICIONES DE ENTRADA Y SALIDA

Las entradas son las partidas de datos presentadas al algoritmo. Un algoritmo tiene ó no entrada ó un número predeterminado de ellas. Las entradas deben ser del tipo para el cual se ha diseñado el algoritmo.

Las salidas son partidas de datos presentadas al mundo exterior como el resultado de la ejecución de un programa basado en el algoritmo. Un algoritmo debe producir al menos una salida (de otra manera, que uso tiene?).

10.4.5 EFECTIVIDAD

Las instrucciones de un algoritmo pueden ordenar a la computadora que sólo lleve a cabo tareas que sea capaz de hacer. Una computadora no puede llevar a cabo una instrucción si tiene información insuficiente ó si el resultado de la ejecución de la orden no está definido.

10.4.6 DEFINICION DEL ALCANCE

Un algoritmo se aplica a un problema ó clase de problemas específicos; el rango de las entradas se tiene que definir previamente; el rango determina la generalidad del algoritmo.

10.5 DIAGRAMAS DE FLUJO

Debido a los detalles requeridos de estos, los lenguajes de programación no son una herramienta apropiada para el diseño del algoritmo inicial. El medio de notación más usado comúnmente para los algoritmos es un diagrama de flujo.

La ventaja más significativa del diagrama de flujo es una presentación clara del flujo del control en el algoritmo, es decir, la secuencia en la que se van a llevar a cabo las operaciones.

Un diagrama de flujo es una representación en dos dimensiones de un algoritmo; los símbolos de un diagrama de flujo definidos previamente se usan para indicar las diversas operaciones y el flujo del control.

En la fig 10.1 se presenta un conjunto básico de símbolos de diagrama de flujo establecidos. Seis de estos símbolos son contornos (también llamadas cajas) de varias formas. Cuando se usan en un diagrama de flujo, contienen las palabras apropiadas, las que las hacen más precisas ya que el diagrama de flujo se desarrolla para resolver un problema dado. El símbolo restante, la línea de flujo, determina la secuencia entre las tareas representadas por los contornos.

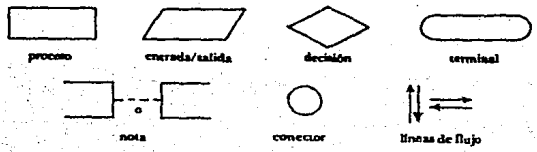


FIGURA 10.1 CONTORNOS DEL DIAGRAMA DE FLUJO

10.5.1. SIGNIFICADOS DE LOS SIMBOLOS.

a) Proceso.- Una ó más tareas de cómputo que se van a realizar de manera secuencial.

b) Entrada/Salida.- Datos que se van a leer en la memoria de la computadora desde un dispositivo de entrada ó datos que se van a pasar de la memoria a un dispositivo de salida.

c) Decisión.- Son posibles dos trayectorias de ejecución alternas. Durante la ejecución se escoge la ruta a seguir, probando si se satisface ó no la condición especificada dentro del símbolo.

d) Terminal.- Aparece ó bien al inicio de un diagrama de flujo (y contiene la palabra "Comienzo" -Start-), ó bien en su conclusión (y contiene la palabra "Detengase" -Stop-).

e) Nota.- Contiene observaciones que simplifican la comprensión del algoritmo ó la descripción del dato.

f) Conector.- Hace posible la separación en partes del diagrama de flujo. Se colocan símbolos de referencia cruzada idénticos en este contorno en donde la línea de flujo se

interrumpe y en donde continúa.

g) Líneas de flujo.- Indican al siguiente contorno.

Los diagramas de flujo permiten al lector seguir la lógica del algoritmo con mayor facilidad que la que tendría una descripción lineal en inglés. Dado que la selección del nivel de detalle está a discreción del diseñador del algoritmo, los diagramas de flujo son apropiados para el método de diseño de arriba hacia abajo, en el que se establece inicialmente la estrategia general del algoritmo y los refinamientos se introducen después. Un método alterno de la presentación del algoritmo, el seudocódigo, se enunciará más adelante.

10.6 MANEJO DE DATOS EN PROGRAMACION

En esta parte del capítulo se presentarán los conceptos generales de los algoritmos. Estos conceptos subrayan todos los lenguajes de programación de uso general y forma parte para aprender cualesquiera de ellos. Al mismo tiempo, las formas de expresión basadas en esos conceptos se usan para el diseño de algoritmos, antes que la codificación real. También se estudia los componentes del programa usados para la manipulación de datos.

Las partidas de datos proporcionan información sobre

los objetivos para los cuales el algoritmo (programa) se refiere en términos generales. Por ejemplo, un programa que calcula los impuestos de los ingresos personales, necesita entre otros datos, el monto de los ingresos de la persona y las tasas de los grupos de impuestos.

Las dependencias naturales entre las partidas de datos, si se usan para estructurar los datos, dan por resultados algoritmos más simples. Por ejemplo, si los grupos de impuestos se acomodaran en una tabla ordenada manipulada por el programa de cálculo de impuestos, los impuestos se podrían calcular muy fácilmente, con los grupos listos para modificarlos si fuera necesario.

Se usan entonces varios tipos de datos y estructuras para presentar los objetos del mundo real y las dependencias entre ellos.

10.6.1 INSTRUCCIONES

Un programa es una representación de un algoritmo, apropiado para su ejecución en una computadora.

Una instrucción es el componente elemental de un programa. Una instrucción de programa es semejante a una oración en un lenguaje natural. Las instrucciones pueden ser

ejecutables ó no ejecutables.

Instrucciones ejecutables son órdenes para que la unidad de procesamiento central de una computadora lleve a cabo una acción durante la realización del programa. Esta instrucciones son imperativas.

Intrucciones no ejecutables son órdenes que no producen instrucciones de lenguaje de máquina para que sean incorporadas en el programa objeto. Estas instrucciones son de caracter descriptivo.

Otra distinción entre las intrucciones de los lenguajes de programación depende de que influyan ó no el flujo de control de un programa. Las instrucciones se ejecutan en forma secuencial, en el orden en que hayan sido presentadas a la computadora, a menos que sea una tranferencia de control. Algunas instrucciones ejecutables transfieren el control a otra instrucción que le sigue de inmediato. Por lo general, la decisión de transferencia se basa en la satisfacción de cierta condición. Las intrucciones ejecutables y las no ejecutables que no transfieren el control ni influyen en la secuencia de ejecución, en su lugar realizan tareas de manipulación de datos.

10.6.2 TIPOS DE DATOS SIMPLES

Los algoritmos y los programas que surgen de ellos, operan sobre los datos. La acción de una instrucción ejecutable se refiere como un cambio de valor de una partida de datos. Los datos de entrada son transformados por el programa, después de las etapas intermedias, en datos de salida. El proceso de solución de problemas, el diseño de la estructura de datos es tan importante como el diseño del algoritmo y del programa que sobre éste se basa.

Sólo se consideran aqui los datos simples (sin estructura). Los datos estructurados, son conjuntos de partidas simples con relaciones definidas entre ellos.

Las partidas de datos, así como las instrucciones de programa, están representadas en la memoria de la computadora por uno ó más bits (generalmente).

Un tipo de datos es una interpretación aplicada a un conjunto de bits que representan una partida de datos dada. Los siguientes son tipos de datos simples y los valores que pueden tomar:

8 bits: de 0 a 255

16 bits: de 0 a 65535

Los datos de 8 bits se almacenan en una localidad de memoria y el procesador los maneja en un solo registro, los datos de 16 bits se almacenan en dos localidades contiguas y el procesador los maneja en un registro par.

Lógicos (ó Booleanos): un tipo de datos con sólo dos valores: Cierto y Falso (representados como 1 y Ø). Básicamente los datos de este tipo se usan para representar las condiciones que refuerzan las decisiones para el flujo de control, aunque se puede utilizar para representar cualquier objetivo con valores binarios.

Caracter: un caracter alfabético ó dígito numérico (llamado conjuntamente caracteres alfanuméricos) ó un simbolo especial (tal como !,',%,&, etc.). Lo podemos introducir en una localidad de 8 bits en forma codificada (por lo regular en código ASCII, auque hay otros códigos). Por lo general, los caracteres se organizan en secuencias llamadas líneas. La representación de un número como una partida de datos numéricos y como una línea, !difieren!. Este tipo de datos aumenta la potencia de la computadora desde el cálculo hasta el procesamiento simbólico general: la manipulación de cualquier texto.

10.6.3 COSTANTES, VARIABLES Y ARREGLOS

El valor de una partida de datos simples en un algoritmo dado puede permanecer constante ó puede variar. Por lo tanto, estas partidas son subdivididas en constantes y variables. La clase de datos estructurados es utilizada la mayoría de las veces es un arreglo

CONSTANTES Y VARIABLES

Una constante es una partida de datos que permanece sin cambio del principio al final del cómputo basado en un algoritmo y está, por tanto, especificada por su valor, que se usa directamente en un algoritmo ó en un programa

Una variable se llama a una partida de datos,el valor de ésta puede cambiar durante la ejecución del programa.

Las constantes y variables pueden ser de varios tipos; por lo que podemos tener una constante lógica ó una variable real.

La referencia a un valor variable se puede hacer solamente por medio de la posición (ó posiciones) en la memoria en la que se ha almacenado ese valor. El valor de la variable, es el contenido actualizado de esta posición así

llamada.

La habilidad para usar variables diferencia a las computadoras de las calculadoras rudimentarias. Su empleo hace posible el posponer los valores actuales presentándolos hasta el punto apropiado durante la ejecución del programa.

El espacio de la memoria se asigna generalmente a las variables como resultado de una intrucción que lo confirme. De modo que, para hacer referencia a una variable, se debe utilizar la orientación de su ubicación. Cuando se usa un lenguaje de alto nivel, el programador no tiene que estar relacionando con la orientación numérica actual. Las posiciones son orientadas simbólicamente. La orientación simbólica significa que el programa se refiere a las posiciones por el nombre de la variable que contienen.

Se puede dar un valor a una variable en dos formas esenciales. Se puede leer desde un dispositivo de entrada ó asignarlo durante la ejecución del programa. La clase de valor que puede adquirir una variable depende de su tipo.

ARREGLOS

Un arreglo es un conjunto de partida de datos del mismo tipo referidos colectivamente por un nombre sencillo.

Las partidas de datos individuales, los elementos de arreglo, son ordenados por suscritos (indices); por esta razón en ocasiones se llaman variables suscritas. La cantidad de suscritos de un arreglo determina su tamaño.

Muy frecuentemente, se emplean los arreglos con una ó dos dimensiones. Un arreglo de una dimensión corresponde a u vector; un arreglo bidimensional a una matriz.

Cada elemento de un arreglo de un tipo dado tiene dos características; su posición dentro del arreglo según sus suscritos y su valor.

10.6.4. ENTRADA Y SALIDA

Las operaciones de entrada y salida que transfieren un valor sencillo son operaciones elementales en los lenguajes. Los detalles son realizados por el software de los sistemas.

En la entrada, el valor se presenta a la computadora por medio de dispositivos de entrada; la salida, la presenta la computadora mediante un dispositivo de salida.

El tipo de datos presentado generalmente se describe

por una instrucción de declaración.

Del dispositivo de entrada los datos se canalizan a las posiciones de la memoria de la computadora. Las constantes son definidas en el propio programa y no se necesita la entrada.

Cuando se ejecuta una intrucción de salida, se canaliza al dispositivo de salida una copia de los datos solicitados.

10.6.5 COMENTARIOS

Todos los lenguajes de programación de computadoras ofrecen la posibilidad de introducir comentarios que expliquen los datos y la lógica de los programas junto con las instrucciones.

Aunque los comentarios se enlistan junto con las instrucciónes del programa, no influyen la ejecución de los programas.

Los comentarios generalmente se identifican por un caracter especial en la línea. Es imperativo el uso responsable de los comentarios en cada programa.

10.7 ESTRUCTURAS DE CONTROL EN EL DISEÑO DE PROGRAMAS

En este punto del capítulo, se presentan las estructuras básicas necesarias para organizar el flujo de control en un algoritmo o en un programa. Tres de éstas son: La secuencia (inicio-fin), la decisión (si-entonces-de lo contrario) constituyen lo fundamental de la organización necesaria para respaldar un proceso sistemático de programación, a menudo llamado programación estructurada. Las estructuras de control adicional tales como la trayectoria de repite-hasta, la trayectoria clasificada y la elección múltiple (en caso que) se pueden emplear para simplificar este proceso. Una trayectoria se puede abandonar prematuramente con la ejecución de una instrucción de salida.

Mientras esta estructuras no estén disponibles directamente en los lenguajes de programación en cada propósito general; es posible construirlas usando las instrucciones de un lenguaje dado. El programador encontrará que pensando en término de estas construcciones producirá programas organizados claramente que son de manera relativa fáciles de escribir , leer y modificar.

En ciertas situaciones de programación no frecuentes, es conveniente, de culquier forma, emplear tranferencia

incondicional de control (ve a).

Se introducen dos herramientas importantes para el diseño de programas. El seudocódigo, utilizado frecuentemente de preferencia los diagramas de flujo, sirve para llevar el algoritmo listo para codificar en un proceso de diseño de programa de arriba a abajo, llamado refinamiento de etapas. Se emplean las tablas de decisión para la consideración sistemática de todas las acciones que va a hacer el programa si existen ciertas condiciones (o sus combinaciones).

10.7.1 CONDICIONES

Dos de las estructuras de programación básic, la decisión y la trayectoria, forman el flujo de control en un programa que depende de la existencia de una condición especificada. Para especificar tales condiciones, se usan las expresiones lógicas. en los casos más sencillos, ésta son relaciones.

QUE ES UNA CONDICION?

Una condición es una afirmación de un valor de una variable ó de una de una dependencia entre los valores de dos ó más variables.

Los valores de una condición son probados y pueden ser Cierto y Falso. Por tanto, esto es especificado por una expresión lógica.

Las condiciones se usan para proporcionar la posibilidad de trayectorias de ejecución alternas en el programa (en una instrucción de decisión) ó para asegurar la realización repetida de un grupo de instrucciones en una trayectoria.

Una condición simple es expresada por una relación lógica, su valor puede ser almacenado en una variable lógica como el resultado de una instrucción.

RELACIONES

Una relación es una expresión lógica que consiste de dos expresiones aritméticas conectadas por un operador relacional, por ejemplo, uno de los siguientes

expresión aritmética operador relacional expresión aritmética

La forma general de una relación es por tanto

< <= = <> >

Las relaciones sirven para expresar las condiciones simples

El valor de la relación es Verdadero si la condición expresada por ésta existe, de lo contrario el valor es Falso. Observe que este valor es de tipo lógico.

Para probar en orden la veracidad de una relación, se evaldan sus lados derecho e izquierdo, y se contesta la pregunta " el operador relacional expresa la verdad" Si lo hace, la prueba da el valor Verdadero; si no, Falso.

CONDICIONES COMPUESTAS

Una condición compuesta se expresa con el uso de cualquier expresión lógica.

Una expresión lógica puede contener, en orden de prioridad:

(a) Operadores aritmético

```
**
* /
+ -
```

Estos sólo pueden unir a los operandos numéricos.

(b) Operadores relacionales

< <- * > >

Estos sólo pueden unir datos numéricos.

(c) Operadores lógicos (Booleanos)

no
y
o

La acción de los operadores Booleanos está especificada por su tabla de verdad en la Tabla 10.1. Los cálculos de las variables lógicas, llamados álgebra booleana, se enunciaron en el capitulo 2

VALORES DE LAS VARIABLES		RESULTADOS DE LA OPERACION		
P	Q	P no	P y Q	P o Q
Falso	Falso	Verdadero	Falso	Falso
Falso	Verdadero	Verdadero	Falso	Verdadero
Verdadero	Falso	Falso	Falso	Verdadero
Verdadero	Verdadero	Falso	Verdadero	Verdadero

TABLA 10.1 TABLA DE VERDAD

Los operadpres lógicos operan exclusivamente en entidades de valor lógico. Esta puede ser relaciones, variables ó constantes lógicas (Verdadero y Falso).

Los niveles de prioridad de las operaciones usadas en las expresiones lógicas son incluidos en la fig. 10.2.

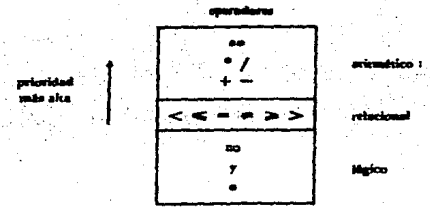


FIGURA 10.2 PRIORIDAD DE OPERADORES

10.7.2. DECISION

Cuando el programador desea especificar dos cursos de acción alternos en un algoritmo (ó un programa), la elección comienza suponiendo la existencia de ciertas condiciones, se usa la condición construida. Para seleccionar en orden entre varias alternativas, las decisiones se generan dentro de otra.

Una herramienta gráfica, llamada tabla de decisión, es útil en el diseño de programas cuando se tiene que considerar un conjunto complejo de condiciones y acciones.

CONSTRUCCION DE UNA DECISION

La instrucción de decisión (ramificación) es el mecanismo para la especificación de dos instrucciones

alternas (ó grupos de instrucciones), una de las cuales se escogerá para la ejecución como resultado de la instrucción de decisión dada. Esto proporciona la posibilidad de expresar lo siguiente: Si una condición dada existe, se debe llevar a cabo una acción alterna; de lo contrario se debe realizar la otra alternativa.

Las condiciones se especifican usando expresiones lógicas. La forma general de la instrucción de decisión es:

si C entonces S1, de lo contrario S2

donde S1 y S2 son instrucciones (ó grupos de instrucciones) y C es una condición.

La instrucción de decisión se ejecuta como sigue:

a) La condición C está probada y se obtiene el valor Verdadero ó Falso.

b) Si el valor es Verdadero, entonces S1 se ejecuta en seguida, de lo contrario (si el valor es Falso) S2 se ejecuta de inmediato. En consecuencia, la siguiente instrucción S2 se ejecuta.

En la fig. 10.3 se muestra el diagrama de flujo de la construcción si-entonces-de lo contrario

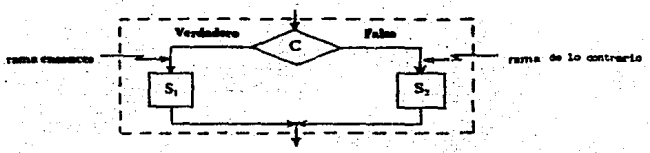


FIGURA 10.3 si-entonces-de lo contrario

Para incrementar la amenidad del estilo del programa, es una buena práctica sangrar esta instrucción como se muestra:

```
si C entonces
     S1
     de lo contrario
     S2
```

Sucede un caso especial de la instrucción cuando la alternativa para la instrucción por ejecutarse condicionalmente es la no acción. Así una instrucción expresa la siguiente idea: Si la condición dada existe, la instrucción S se ejecuta; de lo contrario se ejecuta la siguiente instrucción S en el programa.

El diagrama de flujo de una construccion si-entonces se muestra en la fig. 10.4.

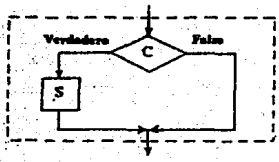


FIGURA 1Ø.4. si-entonces

Esta construccion se presenta como.

```
si C entonces
     S
```

DECISIONES SANGRADAS

Para implantar en orden una decisión múltiple (una selección entre varias alternativas), se usa estructura sangrada si-entonces-de lo contrario. Esta se construye colocando iterativamente otra instrucción como S1 ó S2.

Para el sangrado de dos niveles en ambas ramas obtenemos.

```
si C1 entonces
      si C2 entonces
            S1
            de lo contrario
            S2
      de lo contrario
      Si C3 entonces
            S3
            de lo contrario
            S4
```

El diagrama de flujo de esta estructura particular se muestra en la fig. 10.5.

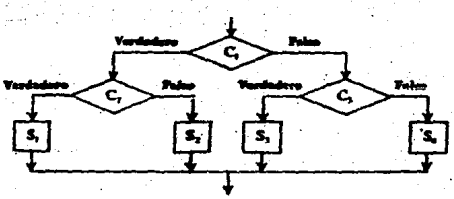


FIGURA 10.5. CONSTRUCCION SANGRADA

Es más fácil el algoritmo (ó programa) si no ocurre el sangrado en la rama de lo contrario de la construcción. Se puede expresar las condiciones para completar esto. Si se mantiene tal disciplina, una construccion si-entoces-de lo contrario tiene la siguiente forma general:

```
si C1 entonces
      S1
      de lo contrario
      si C2 entonces
            S2
            de lo contrario
                  .
                  .
                  .
                        si Cn entonces
                              de lo contrario
                              Sn+1
```

El diagrama de flujo de esta construccion se presenta en la fig 10.6.

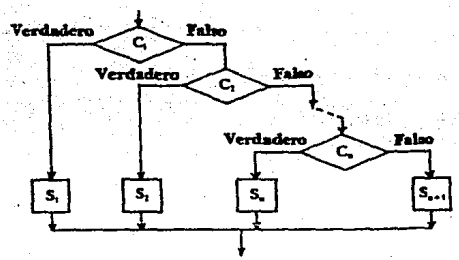
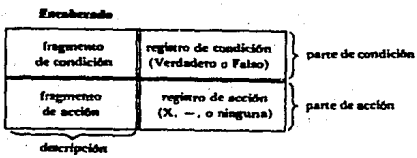


FIGURA 10.6. CONSTRUCCION SANGRADA si-entonces-de lo contrario

TABLA DE DECISION

Cuando existe un número de condiciones en varias combinaciones debe influir de manera más profunda al flujo de ejecución del programa, se puede usar una herramienta gráfica llamada tabla de decisión. Una tabla de decisión especifica en forma tabular las acciones que va a realizar el programa si existe una condición dada ó combinación de condiciones.

Las tablas de decisión tienen el formato que se muestra en seguida.



Por tanto, una tabla consiste de cuatro partes: dos, en la izquierda, listan todas las condiciones y acciones posibles; y dos, en la derecha, indican qué acciones van a tener lugar cuando existe una combinación de condiciones dadas (los registros de estas últimas dos, consideradas verticalmente, son llamadas reglas).

Una tabla de decisión se diseña como sigue:

a) Liste todas las condiciones posibles en el fragmento de la condición.

b) Liste todas las acciones posibles en el fragmento de acción.

c) Proporcione el número de reglas para igualar todas las combinaciones posibles de condiciones (2 para n condiciones) excluyendo las evidentemente imposibles o las innecesarias.

d) Para cada registro de condición, marque con una X el registro de la acción contra la acción (ó acciones) que se va(n) a tomar.

e) Si es posible, combine reglas que difieran solamente en condiciones no relevantes

Subsecuentemente, una tabla se puede simplificar introduciendo los llamados símbolos de sin cuidado (-), que significan que cierta condición no es relevante.

Una tabla de decisión se puede representar entonces directamente como una construcción si-entonces-de lo contrario. Procure llevar a cabo la codificación de manera agradable.

El uso de las tablas de decisión obliga al programador a considerar sistemáticamente todas las condiciones y acciones posibles y es de este modo una ayuda en el diseño del programa junto con las herramientas como el diagrama de flujo ó el seudocódigo.

10.7.3 BUCLE

El mecanismo del bucle (trayectoria) origina la ejecución repetida de una secuencia de instrucciones mientras que es verdadera cierta condición. Cuando al retener la condición cesa, el control pasa a la instrucción siguiente hasta la última instrucción del bucle (trayectoria). Esta ejecución repetida es llamada iteracción.

Si un bucle está registrado, la ejecución de la instrucción incluida en él (llamada condición del bucle) debe, después de un tiempo limitado, causar la inversión de la condición que causó la entrada. Por el contrario, puede existir un bucle infinito, que puede representar un error de programación y requeriría que la ejecución del programa fuera detenida por medios externos.

Cada vez que se ejecuta el cuerpo del bucle, el valor de cuando menos una variable cambia. Por tanto, la ejecución repetida tiene un efecto acumulativo.

La forma general de la instrucción de bucle es

mientras C haga S

donde C es una condición y S es una instrucción o un grupo de instrucciones.

La instrucción del bucle se realiza como sigue:

a) La condición C es probada.

b) Si el valor de C es Verdadero, S se ejecuta y el control se regresa entonces a la instrucción mientras para que la condición sea probada de nuevo; por el contrario (si el valor de C es Falso), la siguiente instrucción S es ejecutada.

El diagrama de flujo de la construcción mientras-haga se muestra en la fig. 10.7.

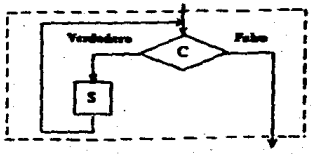


FIGURA 10.7 BUCLE mientras-haga

El sangrado de la forma

```
mientras C haga
      S
```

es buena práctica.

Observe que si C es Falso inmediatamente antes de que se registre la instrucción bucle, S no se ejecutará por completo. Si C es Verdadera inicialmente y S se ejecuta, la ejecución de S podría originar que C se volviera Falso después de un número finito de repeticiones.

10.7.4 INSTRUCCION COMPUESTA

Como se mencionó en las secciones 10.7.2. y 10.7.3. de este capítulo, con frecuencia es necesario colocar más de una instrucción en las ramas alternas de una instrucción de decisión ó dentro de un bucle. Esto hace posible una instrucción compuesta que encierra una cantidad de otras instrucciónes dentro de dos delimitadores, el inicio y el

fin.

La forma general de una instrucción compuesta es

inicio S1; S2;....;Sn fin

donde Si es cualquier instrucción.

El diagrama de flujo de esta construcción se muestra en la fig. 10.8.

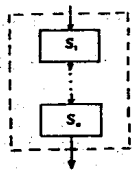


FIGURA 10.8 SECUENCIA inicio-fin

Si una instrucción compuesta no se coloca en una línea sencilla, el sangrado sugerido es

```
inicio
Si
.
.
.
Sn
fin
```

Cualquier programa que conste de más de una

instrucción es por sí mismo una instrucción compuesta.

Las instrucciones que forman una instrucción compuesta pueden ser instrucciones de decisión o de bucle, así como asignaciones e instrucciones de entrada/salida. Los primeros dos tipos pueden, en su oportunidad, incluir una instrucción compuesta en sus ramas ó como un cuerpo de bucle, respectivamente.

10.7.5 TRANFERENCIA INCONDICIONAL

Las construcciones presentadas; si-entonces-de lo contrario, mientras-haga e inicio-fin; son suficientes para expresar un algoritmo. Si un programa se construye usando estas construcciones ó similares exclusivamente, su organización tiende a ser clara y bien estructurada.

Sin embargo, en ocasiones es difícil aplicar estas construcciones en un algoritmo dado. En estos casos se puede emplear una transferancia incondiconal de control (instrucción ve a). Esta instrucción ordena la ejecución de una instrucción dada sin hacer caso de su lugar en la secuencia del programa. El flujo de control entonces procede de éste.

La instrucción de la transferencia incondicional de

control tiene la forma general

ve a

Un nivel es la dirección simbólica de una instrucción del programa. Los niveles son definidos dentro de un programa escribiéndolos al frente de la instrucción y delimitándolos con un simbolo especial (por ejemplo, dos puntos). Una instrucción requiere sólo un nivel si está, ésta referida por otra instrucción, como la instrucción ve a. Los niveles se forman generalmente siguiendo las mismas reglas como en el caso de los nombres de las variables. Los nombres creados por un programador, como los nombres variables y de niveles, son llamados "identificadores".

Dado que un nivel identifica una instrucción, en un programa dos instrucciones no pueden tener el mismo nivel

Un programa puede incluir las instruciones siguientes:

```
        ve a ESE;
             .
             .
             .
ESE: PRIMERO--Ø;
             .
             .
             .
```

Siguiendo la ejecución de la instrucción ve a, el control pasa a la instrucción llamada ESE. Las instrucciones que intervienen no son ejecutadas si está ve a.

El uso de una instrucción ve a se desea raramente y debe seguir una consideración y rechazo de otras alternativas de expresión lógica del programa. Una alternativa disciplinada al ve a, llamada instrucción de salida, está disponible en algunos lenguajes de programación, en casi todos los casos puede servir para reemplazar al ve a.

10.7.6 DISEÑO DE ALGORITMOS. SEUDOCODIGO

Un algoritmo para la solución de un problema surge gradualmente. Por tanto, el proceso de diseño resulta de refinamientos sucesivos del algoritmo de sus originales, generalmente, forma a la punta cuando el algoritmo puede ser codificado en el lenguaje de programación escogido. Esta técnica de diseño de algoritmo (ó programa) se conoce como refinamiento por etapas ó diseños de arriba hacia abajo. Para los programas más largos, este proceso incluye la descomposición modular.

El número de refinamientos necesarios depende de la complejidad del algoritmo. Para presentar los algoritmos se usan frecuentemente dos formas de expresión. La primera, los diagramas de flujo, han sido introducidos en el presente capitulo. Una técnica alterna, el seudocódigo es una presenetación textual de un algoritmo, en donde las acciones a realizar por la máquina son especificadas de una manera

aproximada a un lenguaje natural, con la estructura de control imponiendo la lógica.

Por tanto, un algoritmo final resulta de un proceso de arriba hacia abajo (de lo general a lo particular) de refinamientos sucesivos del seudocódigo. Ultimamente, el algoritmo se puede expresar por las instrucciones básicas presentadas en éste, con la omisión de detalles de un lenguaje de programación particular, tambien se hace un uso extenso de los subprogramas.

El seudocódigo de un algoritmo se vuelve una parte de la documentación del programa.

El estudio detallado del diseño y presentación de dos algoritmos relativamente simples se presenta aquí.

10.7.7 ESTRUCTURAS DE CONTROL ADICIONAL

Mientras que las construcciones "inicio-fin, si-entonces-de lo contrario y mientras que", bastan para diseñar cualquier algoritmo, con frecuencia es más natural reclasificar las diferentes construcciones.

Dos de estos mecanismos son bucles: el de repita-hasta y el clasificado, y el tercero es la instrucción de elección (en caso). También se puede emplear una

instrucción de salida como una forma más disciplinada de un ve a.

CONSTRUCCIONES DE BUCLE ALTERNAS

En ocasiones se usan dos construcciones de bucle alternas. En el bucle "repita hasta", la condición de bucle se prueba siguiendo la ejecución del cuerpo de bucle más bien que antes de éste como en el bucle de "mientras que-haga". Una construcción de bucle clasificado mantiene automáticamente a la variable (bucle clasificado) que determina el número de repeticiones.

BUCLE REPITA-HASTA

En ciertas situaciones de programación, es conveniente ejecutar repetidamente un grupo de instrucciones, con la comprobación subsecuente en cualesquier e las condiciones específicas que hayan surgido. Cuando esto sucede, el control p pasa la siguiente instrucción del bucle; de lo contrario, la iteracción continúa.

La forma general de construir un bucle es

repite S hasta C

donde S es una instrucción ó grupo de instruccines y C es una condición.

Este bucle se ejecuta como sigue.

1. Se ejecuta la instrucción S.
2. Se comprueba la condición C.
3. Si C es falsa, el control se regresa a la instrucción "repita" después de lo cual S se ejecuta de nuevo: de contrario (C es Verdadero) se ejecuta la siguiente instrucción del bucle.

Se debe hacer notar que la instrucción (ó grupo de instrucciones) S se ejecuta cuando menos una vez. Si entonces S se realiza de nuevo, su ejecución repetida debe cambiar el valor de la condición C a Verdadera.

En la fig. 10.9 se muestra el diagrama de flujo de la construcción "repite-hasta".

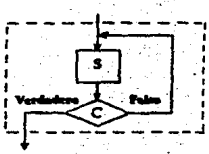


FIGURA 10.9 repite-hasta

Es conveniente sangrar la construcción como sigue

```
repite hasta C
    S
```

La construcción del bucle "repita-hasta" se puede

usar de preferencia al bucle "mientras-haga" cuando el uso del último parezca artificial.

BUCLE CLASIFICADO

La construcción de bucle clasificado (llamada también bucle de conteo) proporciona el control automático del índice del bucle que determina el número de iteracciones. La forma general del bucle clasificado es

para INDEX=VALOR INICIAL hasta VALOR FINAL por PASO haga S

El diagrama de flujo de esta construcción, mostrada en la fig. 10.10 explica cómo se ejecuta.

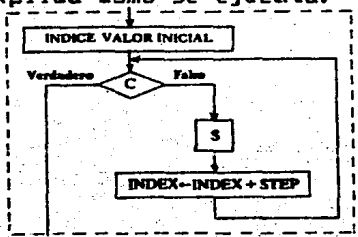


FIGURA 10.10 BUCLE CLASIFICADO

Las cuatro variables usadas en la construcción deben tener valores enteros. Observe que este bucle "hace mas trabajo" que las otras dos construcciones de bucle: inicia y mantiene el índice del bucle, también comprueba la terminación. Por tanto, es una forma apropiada de expresión, en donde se aplique.

ELECCION MULTIPLE

Cuando se toma una decisión que puede tener varias salidas, se puede usar la construcción en caso que I sea C haga S

La forma general de la instrucción "en caso que sea" es:

```
en caso que I
   Inicio
   sea C1 haga
          S1
   sea C2 haga
          S2
           .
           .
           .
   sea Cn haga
          Sn
   fin
```

donde I es una variable con valores que varian desde C1 hasta Cn, y S1,...,Sn son las instrucciones.

Un diagrama de flujo explicando la ejecución de esta construcción se muestra en la fig. 10.11 (un diagrama no standard se usa para indicar una prueba con varias salidas).

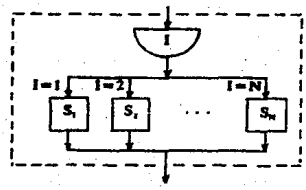


FIGURA 10.11 en caso que I sea c haga S

La instrucción "en caso que sea" se ejecuta como sigue:
(a) Se determina el valor de la variable I.

(b) Si este valor es j, se ejecuta la instrucción S j. En consecuencia, la instrucción que sigue al último componente textual de la instrucción "en caso que sea" (por ejemplo, sigue Sn) es realizado.

Varias alternativas de elección múltiple se pueden codificar fácilmente en una instrucción "en caso". Su uso en ocasiones evidencia la necesidad de una construcción "si entonces-de lo contrario" profundamente arraigada.

SALIDA

Cuando surgen ciertas condiciones es conveniente en ocasiones salir (to exit) de un bucle antes de su terminación. Por supuesto, esto puede ser acompañado con el uso de una instruccion VE A. Una forma más disciplinada de dejar un bucle y quizá las construcciones que la incluyen, sea la instrucción "salir".
La forma general de esta instrucción es

etiqueta salir

donde la etiqueta identifica la primera instrucción de la construcción de la cual se completa salir. Además la ejecución del control de una instrucción "salir" pasa

inmediatamente a la instrucción siguiendo la construcción.

Por tanto, pueden sacar varias etiquetas de construcción de control sangradas.

```
CLASS: repite hasta que CODE=Ø
       inicio
       Entrada CODE, DUE;
       si CODE = Ø entonces
                   salir CLASS
                   de lo contrario
                   inicio
                   en caso que CODE
                      .
                      .
                      .
                   fin
      fin
      finrepite
```

Observe que podemos escribir actualmente

siempre hasta que

dado que la ejecución del bucle "repita-hasta que" siempre se completará mediante la instrucción "salir".

Las instrucciones "salir" se usan con mayor frecuencia para revisar las condiciones especiales, incluyendo los errores, concernientes al procesamiento de datos por el bucle.

10.7.8 CONCLUSION, TIPOS DE INSTRUCCIONES EN LA PROGRAMACION

Los siguientes tipos de instruciones son deseables en los lenguajes de programación de propósito general.

Primero, existen tres construcciones de control, suficientes para controlar la secuencia de ejecución de cualquier programa:

(1) inicio-fin;
(2) si-entonces-de lo contrario
(3) mientras que-haga

La sencillez estructural de un algoritmo (ó programa) construido con el uso exclusivo de estas estructuras de control se debe a su propiedad de que uno entra, uno sale: sus diagramas de rojo tienen una línea de flujo de entrada y una de salida simple. Puesto que cualquier texto se lee de arriba a abajo, éstas se deben considerar como guías generales para la organización del flujo del programa. Algunas de estas estructuras están incluidas explícitamente en ciertos lenguajes de programación, otras se pueden construir de las instrucciones disponibles.

En los casos excepcionales, cuando parece forzado expresar un algoritmo usando estas estructuras, la transferencia de control
4) ve a
se puede usar.

Además de las instrucciones que controlan el flujo del programa son nesesarias las siguientes instrucciones del manejo de datos.

(5) entrada-salida;

(6) asignación;

(7) declaraciones (no ejecutables)

Para explicar la lógica del programa y el siignificado de los datos, se deben usar los comentarios.

Para la conveniencia del programador, se ofrecen muchas instrucciones adicionales en los lenguajes de programación de alto nivel. Las más importantes de éstas son las estructuras de control adicional, que incluyen:

(1) bucle repite-hasta que

(2) bucle clasificado;

(3) instrucción en caso que

(4) instrucción salir.

10.8 SUBPROGRAMAS Y PROGRAMACION MODULAR.

Un algoritmo de cómputo de cierta coplejidad y un programa que surge de él, se pueden considerar un sistema. Para diseñar cualquier sistema, necesitamos estar dispuestos a identificar su estructura jerárquica y entonces construirlo de elementos autocontenidos pero que interactúen módulos. Usados en la programación para este propósito son los

subprogramas (también llamados procedimientos): llamados secuencias de instrucciones que llevan a cabo tareas especificas y se les puede llamar por sus nombres. La presencia de subprogramas en un programa interrumpe al programa prinncipal que inicia la ejecución y a los subprogramas que son llamados por el programa principal u otros subprogramas durante la ejecución. Colectivamente, al programa principal y a los subprogramas se les llama "modulos o unidadades de programa".

Existen dos tipos esenciales de subprogramas: los funcionales y más comunes, las subrutinas.

Durante la invocación (llamado) de un subprograma, los dos módulos involucrados se comunican por medio de las partidas de datos especificados. Esta comunicación se puede completar por la transmisión explicita de datos (llamada paso de parámetros) ó a través de las posiciones de la memoria comprtida que contiene los datos. El segundo método de comunicación, donde se consideran ciertos datos globales para los subprogramas, y por tanto, accesibles a otras unidades del programa, es destacado particularmente en los llamados lenguajes de programación estructurados a bloques.

El empleo repetido de subprogramas, cuando es posible en un lenguaje de programación, tiende en ciertos casos a

simplificar los algoritmos.

El diseño de programas con una identificación consistente de tareas bien definidas y la asignción de éstos para separarlos en módulos, se le llama programación modular.

10.8.1 DEFINICION E INVOCACION DE SUBPROGRAMAS.

Los dos tipos esenciales de subprogramas (procedimientos) son las funciones y subrutinas. Siguiendo su ejecución, una función retorna, a la unidad de programa que la invocó, un valor simple como el valor del nombre de la función. Las subrutinas son los subprogramas más comunes que se pueden usar para comunicar a la unidad del programa llamado cualquier número de valores. Sin embargo, aunque una función se puede usar para regresar un número de valores demanera idéntica a la de las subrútinas, esta práctica se está evitando.

Los subprogramas, como unidades de programa autocontenidos, tienen todas las propieddes de los algoritmos con la posible excepción de la falta de salida, ya que otro módulo en el programa la puede llevar a cabo. Los subprogramas se pueden traducir de manera independiente por razones de prueba. Siguiendo la invocación y la ejecución de

un subprograma, el control regresa al lugar del llamado en la unidad del programa invocado. Resultando sin cambio elflujo de control en esa unidad del programa.

SUBRUTINAS

Una subrutina es la más general de las dos clases de subprogramas. Una subrutina es un subprograma que puede regresar explicitamente un número ilimitado de valores a la unidad del programa invocado. La invoacación de la subrutina es el término llamado de subrutina..

Una instrucción llamada de subrutina tiene la siguiente forma general:

llamado** nombre de la subrutina (lista de los parámetros actuales).

El nombre de la subrutina se forma en un lenguaje de programación como cualquier otro identificados, por ejemplo, un nombre de variable. Este identifica a la subrutina entre los subprogramas contenidos en un programa dado.

Los parámetros (tambien llamados argumentos,-tabla 10.1-) son partidas de datos mediante los cuales una subrutina se comunica explicitamente con la unidad del

programa llamado (que puede ser el programa principal ó cualquier otro programa). Los parámetros son predefinidos por una subrutina conforme al número y tipo de cada uno de ellos.

Los parámetros pueden comunicar valores a la subrutina (en ocasiones se les llama parámetros de entrada) ó desde la subrutina a la unidad del programa llamado (parámetros de salida), ó puede servir a ambos propósitos presentando un valor a la subrutina cuando ésta es llamada y adquirir subsecuentemente valores como resultado de su ejecución (parámetros de entrada/salida).

Los parámetros pueden constituir valores simples ó arreglos de valores.

Uno de los propósitos de los programas es su uso múltiple en un programa. Por tanto, en varios módulos del programa se pueden incluir llamadas a un subprograma dado. Durante cada llamada, los diferentes valores de los parámetros actuales son incluidos generalmente en la lista.

Para que un llamado en orden sea significativo, se tiene que definir e incluir en el programa la llamada subrutina.

Una definición de subrutina consiste del encabezado de una subrutina no ejecutable de la siguiente forma general.

"Subrutina" nombre de la subrutina (lista forma de los parámetros)

y el cuerpo de la subrutina, por ejemplo, las instrucciones que constituyen la subrutina. La aparencia textual del cuerpo de la subrutina es similar al de un grupo principal.

Siguiendo la ejecución de la instrucción última de una subrutina, el control de la ejecución se revierte a la unidad del programa llamado, específicamente, a la instrucción siguiendo de un modo textual a la subrutina llamada. Esto es el llamado retorno de la subrutina. Por tanto, la ejecución de la subrutina es una división temporal del control de la unidad del programa llamado.

Las instrucciones del encabezado de la subrutina listan los parámetros formales de la subrutina y/o comunicados por ésta a la unidad del programa llamado. La subrutina se escribe mediante el empleo de estos nombres más bien que con los nombres usados por las partidas de datos en el programa llamado. De este modo, las dos formas se pueden escribir y traducir independientemente, así como las listas de la pareja de argumentos formales y actuales. En

consecuencia, los parámetros formados constituyen retenedores de lugar para los actuales. Los parámetros formales pueden adquirir valores de los parámetros actuales durante el llamado y pueden regresar valores a los parámetros actuales durante el retorno. La asociación entre los parámetros actuales y formales ocurre durante la llamada de la subrutina.

Se usan nombres diferentes en los diversos lenguajes de programación para los parámetros formales y actuales. Los nombres más frecuentes usdos se enlistan en la tabla 10.1

EN EL MODULO LLAMANDO	EN EL MUDULO LLAMADO	NOMBRES USADOS EN
parámetro actual	parámetro formal	ALGOL, Pascal
argumento actual	argumento formal	FORTRAN
argumento	parámetro	PL/I

TABLA 10.1 TERMINOLOGIA PARA EL PASO DE PARAMETROS

Junto con los valores de los parámetros, una subrutina puede usar cualquier número de variables ó arreglos

locales a los que se les declaran y signan valores dentro de ésta. Su existencia es desconocida para otras unidades del programa.

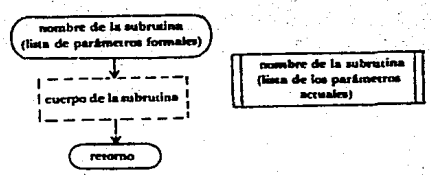
El programador está libre de elegir cualquier nombre válido para las entidades (por ejemplo, los nombres ó niveles de variales) Localizar para una unidad de programa. Si sucediera que los nombres de tales entidades en dos unidades coincidiederan, se tratarán como si estuvieran referidos a partidas diferentes.

Mientras que los parámetros formales siempre tienen que ser nombrados de variables ó arreglos -(ve algún sentido en emplear constantes como parámetros formales)- los parámetros actuales pueden ser expresiones (y, en particular, constantes, variables y arreglos de elementos), así como arreglos.

Se puede dibujar una analogía entre las funciones algebraicas y los subprogramas en programación en los que una función algebraica se escribe en términos de fijadores de lugar para los valores actuales.

Puesto que una subrutina es una unidad de programa independiente que se puede usar en cualquier programa, tiene su propio diagrama de fabricación (véase la fig. 10.1). La

línea standard que se usa para mostrar el llamado de una subrutina en el diagrama flujo del programa llamado está presente en la fig. 10.1



FIGUTA 10.12 DIAGRAMA DE FLUJO DE LAS SUBRUTINAS

A la instrucción en la subrutina en donde comienza la ejecución de ésta siguiendo el llamado se le denomina punto de entrada; la instrucción que regresa el control al programa llamado en el punto de retorno de la subrutina. Con mucha frecuencia, éstas son respectivamente, la primera y última instrucciones en la subrutina. Varios puntos de entrada tienen sus propios nombres y listas de parámetros.

CAPITULO XI

HARDWARE

11.1 INTRODUCCION

La máquinaria, la CPU, todos los periféricos, cualquier dispositivo microelectrónicos de almacenamiento como lo son discos y cintas, la memoria de la computadora y en general toda la circuiteria, constituyen el hardware de una computadora.

Operacionalmente un sistema de cómputo está constituido tanto por el hardware como por el software. Uno no puede trabajar sin el otro, y cada uno dicta reglas para el otro. En un sistema de cómputo el hardware establece las reglas del conjunto de instrucciones que puede ejecutar y las instrucciones del sofware le indican lo que desea hacer.

Aunque son inseparables dentro de un sistema de cómputo, el software y el hardware han tenido un crecimiento y una evolución muy diferentes. El hardware se ha convertido en el mundo del almacenamiento y transmisión: en tanto que el software es el mundo de la lógica y el lenguaje.

A mayor capacidad de almacenamiento en el disco y en

memoria del sistema de cómputo, mayor es el trabajo que puede realizar. Entre más veloz sea la transmisión entre la memoria y los discos, ó dispositivos periféricos, más rápido se realizará el trabajo. Aunque la lógica está involucrada en el diseño de los circuitos electrónicos y de la arquitectura de la computadora, su enfoque principal es aumentar la eficiencia del almcenamiento y la transmisión. Desde el punto de vista del hardware, los problemas del usuario se transforman en necesidades materiales basados, por ejemplo, en el tipo de procesamiento deseado ó en el tamaño de los archivos y bases de datos que se deben crear.

11.2 NORMAS Y COMPATIBILIDAD

En computación las normas especifican el diseño de códigos e interfaces básicas tanto para hardware como para software. Las normas proporcionan un marco de trabajo dentro del cual se diseña e implementa todo el software y el hardware. Todo sistema de cómputo tiene normas, lo cual no significa que hay una sola norma adoptada por todos los fabricantes. Las normas implementadas en la computadora determinan la flexibilidad que se tendrá ahora y en el futuro.

Hay varios tipos de normas de medios de almacenamiento. Las normas visibles son relativamente

fáciles de identificar. Existen varios tipos de paquetes de discos, cartuchos de cinta y casetes. Las normas visibles sólo aseguran que el medio de almacenamiento puede insertarse en una unidad periférica particular. Las normas invisibles de hardware estan determinadas por las características de grabación (número de pistas, número de sectores, bits por pulgada, etc.)

Las interfaces de hardware son los cables y conectores físicos que proveen la conexión entre los componentes del hardware. Estas interfaces determina el diseño del conector, su tamaño, número de alambres, etc., y las señales eléctricas que serán transferidas por ellas. Hay muchas normas e interfaces de hardware particulares utilizadas por los fabricantes. Para la compatibilidad en comunicaciones tienen mucho uso las normass, como la RS-232. Este tipo de normas permiten mezclar productos como los módems, terminales e impresoras de muchos vendedores diferentes.

11.2.1 PROTOCOLOS DE COMUNICACION

Una de las características claramente observable en la evolución de la tecnologia de los computadores es la tendencia a la modularidad. Los elementos estructurales de los computadores se conciben cada vez más como unidades

dotadas de cierta autonomía que cooperan cada vez más entre sí. Esta tendencia se fundamenta no sólo en la búsqueda de diseños más rápidos y eficientes si no también en el principio de la división de funciones que facilita la concepción, diseño y mantenimiento de los diversos elementos que forman el sistema.

También es fácil constatar la atención que se ha prestado en la última década al estudio de los sistemas informáticos distribuidos que brindan la posibilidad de compartir recursos informáticos y aumentar la fiabilidad y la disponibilidad de los sistemas a precio justificable. Los multiprocesadores y las redes de computadores son ejemplos de este tipo de sistemas. La aparición y extensión de los microprocesadores incrementan el interés en los sistemas distribuidos al posibilitar la concepción de redes de microprocesadores y de nuevas arquitecturas multiprocesador formadas por un número elevado de miroprocesadores que se comuniquen entre sí de forma bien definida, sin necesidad de un control centralizado.

Las anteriores consideraciones sugieren la posibilidad de contemplar todos los sistemas informáticos como un conjunto de unidades más ó menos autónomas con una función bien definida que colaboran entre si para consecución de tareas determinadas. La naturaleza y los objetivos

perseguidos con esta cooperación difiere radicalmente de un sistema a otro, existiendo, sin embargo, ciertos aspectos comunes a todos ellos. Uno de estos aspectos es la necesidad de intercambiar información entre los elementos que lo integran. Estos pueden ser circuitos, módems, concentradores, terminales, computadores, procesos, personas, etc.

Se denomina comunicación al intercambio de información entre los componentes de un sistema. La forma en que se realiza la comunicación depende de múltiples factores, pero en cualquier caso es indispensable establecer claramente las reglas que han de seguirse en el intercambio de información. Se denomina protocolo de comunicación al conjunto de reglas que siguen la comunicación entre los elementos de un sistema. La materialización hardware ó software de estas reglas recibe asimismo la denominación de protocolo.

En las anteriores definiciones no se ha impuesto ninguna restricción al tipo de información que intercambian los elementos del sistema. El concepto de comunicación se extiende desde el intercambio de datos, que presumiblemente involucrará procesos complejos de software, hasta la simple señal que notificará a un elemento el estado de otro.

11.2.2 SISTEMAS DISTRIBUIDOS Y COMUNICACION

Los sistemas distribuidos son aquellos que tienen cada uno de sus subsistemas básicos (proceso, control y memoria) físicamente distribuidos en mayor ó menor grado. Los multiprocesadores son un caso particular de sistemas distribuidos en los que solamente se exije la distribución de la unidad de proceso, pudiendo encontrarse la memoria y el control distribuidos ó no.

Desde el inicio de la construcción de la red Aparnet a cargo del Departamento de Defensa de los EE. UU., hacia 1967, hasta nuestros días, se ha hecho un esfuerzo considerable en el estudio de cómo deben concebirse y realizarse los protocolos necesarios para el funcionamientoo de las redes de comptudadoras.

Un criterio ampliamente extendido de clasifiación de las redes ésta basada en las características y componentes de la comunicación. Según este criterio las redes pueden dividirse en redes para compartir recursos, redes para cálculo distr buido y redes para comunicaciones remotas.

En el primer tipo de redes el objetivo perseguido es permitir el acceso y utilización remota de recuros informáticos tales como archivos en discos, impresoras, etc. Para conseguir esto es necesario poder establecer la

comunicación entre un programa que se ejecuta en el computador y el programa que controla en otro la utilización del recurso. En las redes orientadas al cálculo distribuido especializados y los sistemas distribuidos de control en tiempo real, la comunicación se establece entre programas localizados en diferentes computadores que cooperan en una tarea común. Finalmente, las redes para comunicaciones remotas ofrecen conexiones entre usuarios y sistemas de cálculo remoto.

En los tres tipos de redes los distintos elementos de sistema distribuido (programas, procesos, periféricos, terminales interactivas, etc.) intercambian información, unas veces a través de mensajes cortos y relativamente independientes los unos de los otros (por ejemplo transacciones de archivos). En cualquier caso es necesario disponer de protocolos capaces de controlar la transferencia de información entre los elementos del sistema.

11.2.3 JERARQUIA DE LA COMUNICACION

Originalmente los protocolos se diseñaban como un proceso único que controlaba todas las operaciones de la red destinada a la comunicación. El incremento en el número de redes y la diversificación de servicios que debe suministrar exige la creación de normas para la confección de

protocolos.

De este aspecto se han encargado fundamentalmente el CCITT (International Telegraph and Telephone Consultive Commitee), la ISO (International Organization for Standarisation), el ANSI (American National Standars Institute), y la IFIP (International Federation for Information Processing) en un intento por crear normas de aceptación internacional

El criterio adoptado por estos organismos concuerda con una concepción de los protocolos como estructura multinivel jerarquizadas. Para un nivel determinado los niveles inferiores son transparentes, es decir, estos le ofrecen un conjunto de funciones de comunicación que él utiliza sin tener que tomar en cuenta la forma en que aquella se realiza (fig. 11.1)

USUARIOS

NIVEL 4

NIVEL 3 FIN A FIN CONTROL DE LA RED

TRANSPORTE

NIVEL 2 CONTROL DE ENLACE

NIVEL 1 ENLACE FISICO

FIGURA 11.1 NIVELES DE JERARQUIA

La delimitación exacta de estos niveles depende en parte del sistema de que se trate. La fig. 11.2 muestra los diferentes niveles de una arquitectura típica

Un protocolo determinado controla el flujo de información entre las dos partes físicas separadas de un mismo nivel. Igualmente es necesario establecer las reglas de comunicación entre los distintos niveles. Para distinguir los protocolos "horizontales" (entre partes de un mismo nivel) de los "verticales" (entre niveles adyacentes) se denomina a estos últimos interfaces de comunicación.

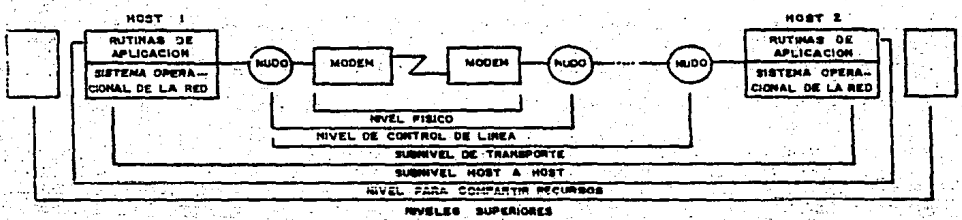


FIGURA 11.2. DIFERENTES NIVELES

La utilización de término interfaz es coherente con el hecho de que generalmente los equipos que realizan los protocolos de distinto nivel son diferentes, constituyendo las reglas de comunicación una verdadera interfaz. Contrariamente, la comunicación dentro de un nivel suele realizarse entre entidades idénticas ó cuando menos similares.

Las funciones básicas de los protocolos e interfaces

son:

- Direccionamiento: la especificación del origen y destino de la información.
- Control de error: La detección y recuperación de errores en la transmisión correcta de una única copia de cada mensaje.
- Control de flujo: En esta función se engloban las operaciones destinadas al mantenimiento del flujo de la información como son la selección de la ruta que han de seguir los mensajes, la reserva de espacio (de memoria) en la estación destino para información que ha de enviarse, etc.
- Sincronización: Todo protocolo ha de ser capaz de mantener en sincronismo las partes de un mismo nivel en el curso de la comunicación.

Otras funciones características de los protocolos son la conexión y desconexión de los circuitos físicos en ciertas aplicaciones, el mantenimiento de una secuencia ordenada de los mensajes, la segmentación de mensajes en unidades de transmisión más pequeñas (paquetes y su posterior reconstrucción, el tratamiento de mensajes con prioridades diversas, etc).

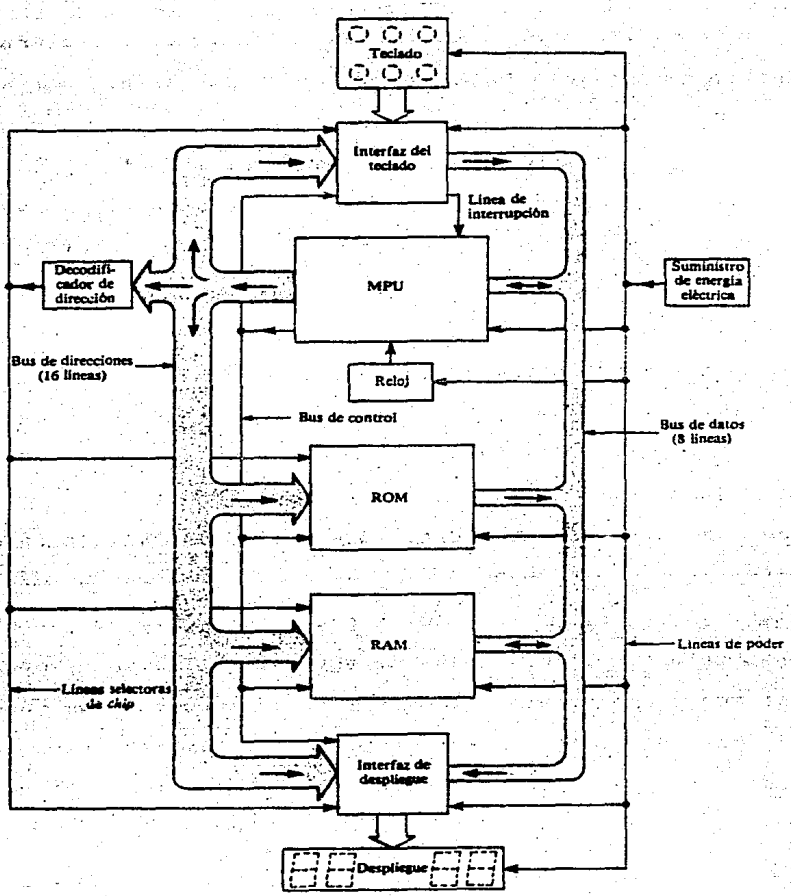


FIGURA 11.3 ARQUITECTURA DE UNA MICROCOMPUTADORA

11.3 ARQUITECTURA SIMPLIFICADA DE UNA MICROCOMPUTADORA

En la fig. 11.3 se diagrama la arquitectura de una microcomputadora. En el centro de todas las operaciones está la CPU (unidad microprocesadora). La CPU necesita conexiones de energía eléctrica y del reloj. El reloj puede ser un circuito separado ó estar dentro del chip de microprocesador. Una CPU ordinaria puede tener 16 líneas de dirección que forman un bus de dirección de un sentido. La CPU tiene también las acostumbradas 8 líneas de datos que conectan a un bus de datos de doble sentido.

La arquitectura de una microcomputadora mostrada en la fig. 11.3 muestra dos tipos de memoria de semiconductor que se utilizan en este sistema. La ROM es la memoria permanente que probablemente contiene el programa monitor del sistema y tiene entradas de direccionamiento junto con el chip selector y las líneas de entrada de lectura disponibles. El ROM tiene también ocho salidas de tres estados conectadas al bus de datos. Por supuesto el ROM debe tener también conexiones que le suministren energía eléctrica, aunque a menudo se omitan en los diagramas de bloques.

La arquitectura de la fig. 11.3 también muestra un RAM como dispositivo de almacenamiento de lectura/escritura temporal. El RAM tiene entradas de dirección junto con un

selector de chip y las entradas disponibles de lectura//escritura. El RAM tiene ocho salidas de tres estados conectadas al bus de datos.

Un tipo de memoria que no esta incluido en la fig. 11.3 es la EPROM, que es una memoria de lectura exclusiva reprogramable. En el EPROM es posible almacenar instruciones e información ó ambas. Son utilizadas para almcenar software permanente. Cuando se desea borrar el contenido de una EPROM debe colocarse la ventana de la misma bajo una luz ultravioleta.

El sistema de microcomputadora diagramdo en la fig. 11.3 utiliza un teclado como dispositivo de entrada. Esa linea del teclado estan conectadas a un circuito especial llamado interfaz del teclado. En el momento apropiado, la interfaz interrumpe a la CPU por medio de la linea que sirve para tal efecto. Esta señal de interrupción ocasiona que la CPU:

- Termine de ejecutar la instrucción que tenia en el momento de la interrupción.
- Suspenda la operación normal
- Brinque hacia un grupo especial de instrucciones en su programa monitor que maneja la entrada de datos desde el teclado.

El circuito de interfaz de teclado tiene dirección, selector de chip y entradas de control para activar la unidad. Cunado se activa la interfaz del teclado coloca los datos en el bus correspondiente. La CPU acepta los nuevos datos de entrada a través del bus. Cuando las entradas de la interfaz de tres estados no se activan, regresan a su estado de alta impedancia.

La microcomputadora de la fig. ii.3 utiliza como salida un grupo de despliegue de siete segmentos. Un circuito especial de interfaz de despliegue se usa para almacenar los datos y manejar los despliegue. Cuando es activada por la dirección, el selector de chip y las entradas disponibles, la interfaz acepta los datos fuera del bus y los almacena. Luego, la interfaz maneja los despliegues en forma continua mostrando visualmente los datos almacenados.

Las 16 líneas del bus de dirección pueden contener 65,536 patrones diferentes de ceros y unos (2). Las líneas de dirección del bus pueden ser conectadas a diversos dispositivos tales como RAM, ROM e interfaces, para encender ó habilitar solamente el dispositivo correcto, un decodificador de dirección muestra los datos en el bus de dirección. La combinación lógica del decodificador de dirección al chip selector apropiado y de esta forma habilita el dispositivo correcto. El importante mencionar que no

todas las 16 líneas de dirección van hacia el decodificador de direcciones, memorias ó interfaces.

11.4 LA CPU Y SU BUS (UNIDAD CENTRAL DE PROCESO)

El bus de las mini ó las microcomputadoras son trayectorias electrónicas, ó alambrado, usado para enviar y recibir información entre los elementos de la computadora, memoria y periféricos. Los anteriores buses son empleados para la comunicación. Existe también un bus de potencia que provee de voltaje pra cada uno de los componentes qe estan conectados a la computadora. Generalmente se supone la existencia del bus de potencia, siendo omitido del diagrama general del sistema de cómputo. En la fig. 11.4 se muestra un diagrama general de un sistema de computo y sus buses.

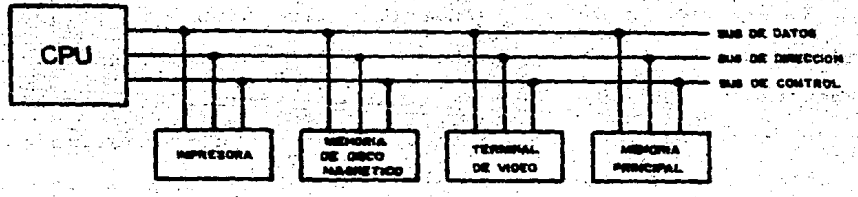


FIGURA 11.4 CPU Y SUS BUSES

Un bus de direcciones generalmente consista en 16 líneas en paralelo. La CPU manda la dirección binaria al bus de la localidad de memoria ó dispositivo periférico hacia ó desde donde se envía ó recibe información.

El bus de datos puede ser unidireccional ó bidireccional. En un bus bidireccional la entrada ó salida de datos es colocada en la misma línea. Siendo una línea de bus de control la que indica si los datos van a ser leídos ó escritos. En un bus unidireccional se usan dos líneas separadas, una para la entrada y otra para la salida la CPU.

El bus de control es un grupo de alambres que contienen señales de control que indican el estado de la CPU y las funciones de control para las señales de dirección y de datos. En el más popular de los buses de microcomputadora es el bus S-1ØØ.

El bus del sistema de una computadora es a menudo puesto en un conector eléctrico tipo hembra en el que se introduce un conector macho de una tarjeta de circuito impresos. Por lo tanto este bus del sistema de cómputo se puede conectar de una a otra tarjeta de circuito impreso. De esta manera la computadora esta costruida en forma de una serie de tarjetas de circuito impreso, conectadas dentro de un gabinete que contiene una fuente de alimentación.

Para conectar un periférico a la CPU y al bus de la computadora, es necesario tener acceso eléctrico. En la mayoría de las computadoras esto significa que es necesario

tener disponibles uno ó más conectores vacios para una conexión física de una tarjeta de interface. En el caso de una impresora que tenga una tarjeta de interface, esta sensará cuando la impresora es direccionada, identificando cuando la impresora es direccionada y el tiempo en que la señal de datos esta disponible en el bus de datos. En algunas ocaciones las señales de datos en paralelo son trasladados a una forma serie, lo cual puede requerir que se amplifique la señal de dato lo suficiente para que pueda menejar al periférico con la potencia adecuada. La impresora en este ejemplo es conectada en la terminal opuesta, al conector, en la tarjeta del circuito de interface.

11.4 BUS NORMALIZADO S-100

Para la comunicación a nivel más interno en una microcomputdora, se han descrito tres buses:

- MUBUS : Un intenso de normalización a nivel europeo.
- MULTIBUS : Bus propuesto por un fabricante de microprocesadores, que gracias a la gran cantidad de seguidores, se ha convertido casi en una norma y que además permite trabajar a varios procesadores sobre el mismo bus.
- S-100 : Este bus ha pasado de los microcomputadores personales a norma internacional, propuesta por el IEEE.

El bus normalizado S-100 es una norma de bus en paralelo util para la comunicación entre módulos en alta velocidad. Es de aplicación a elementos de interfaz para componentes de sistemas computadores interconectados a través de un conjunto de 100 líneas en paralelo, que estan formadas por líneas para la trasferencia simultánea de informaciones de tres tipos: direcciones, datos y controles. Cada subconjunto particular de hilos físicos (bus particularizado) transportará información correspondiente al ciclo especifico de operación que esté en curso de ejecución.

El bus S-100 es actualmente el más usado (es empleado por más de un cenetenar de fabricantes para varios cientos de tarjetas y sistemas completos), aunque conviene advertir que no son completamente compatibles todos los que afirman serlo.

En sistemas microcomputadores (ó en sus subsistemas) el número total de dispositivos conectados no debe superar la veintena. Se deben usar en una via de transmisión que sea eléctricamente corta (no influyen los retardos de propagación). Se puede trabajar con velocidades máximas de transferencia bajas, con señales en el bus de hasta 6 MHz.

Esta norma trata de definir un sistema de interfaz de manejo fácil y racional, que asegure la compatibilidad de los

diseños actuales y futuros sobre el bus S-100, con facilidad de ampliaciones modulares y de interconexiones de dispositivos de fabricantes distintos para construir sistemas completos, con la menor limitación posible en el rendimiento de cada una de las partes componentes.

Históricamente este bus fue usado en primer lugar en los microcomputadores Altair por la firma MITS Inc. con diseños basados sobre el 8080. Actualmente se utiliza el bus S-100 para sistemas con otros microprocesadores (Z-80, 6800,6502 etc.), es el más amplio soporte comercial en el área de los microcomputadores. Debido al diseño de los primitivos sistemas que usaban el bus S-100 la norma original contiene gran cantidad de señales de control (39) que eran necesarias para cubrir la falta de pastillas del circuito Intel, pero que ya no resultan precisas en su totalidad.

A pesar de que la utilización del bus S-100 se ha hecho extensiva desde hace más de diez años, su estandarización estricta sólo ha sido propuesta hasta hace pocos años, en base a los trabajos que una subcomisión del IEEE realizó desde 1976. La especificación preliminar se presentó en junio de 1978, y posteriormente una propuesta modificada de esta norma se publicó en 1979, con vistas a sus presentación ante el IEEE Standards Board, para su exposición y discusión. Esta versión, que se conoce como IEEE Task

696,1/D2, es la que se va a exponer en lo que sigue.

En esta norma el bus S-100 se trata de eliminar algunos problemas que tenía la versión anterior, y de actualizar su nivel para que resulte apropiado para los microprocesadores de 16 bits. A tales fines se ha ampliado de 16 a 24 el número de líneas de direcciones, y el bus de datos que estaba formado por dos buses unidireccionales de 8 bits (uno para entrada y otro para salida de datos) se presenta con posibilidd de constituir un bus bidireccional de 16 bits.

Se incluyen dos señales de diálogo (handshaking) para permitir la coexistencia de tarjetas de memoria sobre 8 a 16 bits, se ha añadido 3 líneas más para alimentación y otra para fallas en la misma.

Esta nueva norma especifica un protocolo para acceso directo a memoria (DMA), con posibilidd de que hasta 16 dispositivos puedan engancharse al bus con facultad de tener control del mismo, siempre que se incluya en el sistema el circuito de arbitraje correspondiente.

Para comunicar mensajes entre diversos dispositivos a través del S-100 precisa de dos elementos funcionales básicos:

a) Un dispositivo que funcione como amo del bus encargado de generar un ciclo de bus generalizado para efectuar la transferencia, y que tenga capacidad para direccionar a los esclavos del bus y para generar todas las señales y mensajes de gestión de interfaz, así como para la trasferencia de mensajes de datos hacia (ó desde) el esclavo direccionado durante aquel ciclo del bus.

b) Un dispositivo que actúe como esclavo del bus (un dispositivo de E/O ó la memoria por ejemplo), que puedan examinar todos los ciclos de bus y sea capaz de transferir mensajes de datos hacia (ó desde) el amo del bus, siendo direccionable. No se encarga de ningún tipo de gestión de interfaz.

Todos los dispositivos conectados al bus quedan así clasificados en uno de los dos grandes grupos anteriores.

Los amos del bus se subdividen a su vez en dos categorías: permanentes y temporales. Un amo permanente del bus es aquel que tiene la prioridad más alta en el sistema de interfaz (normalmente la CPU) y el que tiene el control del bus normalmente. Un amo temporal del bus es el que tiene que solicitar mediante una operación de bloqueo (hold), al amo permanente que le cede el control del bus durante un número variable de ciclos, para después devolvérselo. Un ciclo DMA

consiste en la transferencia temporal de control del bus desde la CPU a un amo temporal y el retorno correspondiente. No se considera la posibilidad de tener varios niveles DMA.

En el caso de que se tengan en un mismo sistema varios procesadores como amos temporales del bus, en lugar de un amo permanente puede resultar más eficaz implantar un "amo simulado" que no dirija y gestione los ciclos del bus, sino que se limite a suministrar un intervalo de arbitraje para que pueda estabilizarse el bus de control del DAM. De esta manera el amo simulado consigue el control del bus entre las tomas de control por los amos temporales, y pone en estado nulo al bus de salida de controles, pasando después el control al siguiente solicitante tras un intervalo de arbitraje que dure un ciclo de reloj.

En la norma IEEE 696.1/D2 las 100 posibles señales del bus se normalizan numeradas según las patillas del 1 al 100, especificandose la función para 93 señales. Todas las señales, excepto las de alimentación, están limitadas a niveles positivos entre 0 y 5 y no pueden tener tiempos de subida ó bajada (en carga) inferior a 5 ns.

Distribuyendo funcionalmente todo el conjunto de señales del S-100 se obtendrá la siguiente clasificación:

a) bus de datos, con 16 líneas (terminales 35, 36, 38 a 43 y 88 a 95)

b) bus de direcciones con 16 líneas (terminal 29 a 34, 37 y 79 a 87)

c) bus de control, con un total de 27 líneas.

c1) estado, con 8 líneas (terminal 44 a 48, 58, 96 y 97)

c2) salida de controles, con 5 líneas (terminal 25, 26, y 76 a 78)

c3) entrada de controles, con 6 líneas (terminal 3, 12, 60 y 72 a 74)

c4) cotrol D.M.A. con 8 líneas (terminal 14, 18, 19, 22, 23, 55 a 57)

d) bus de interrupciones vectorizadas, con 8 líneas (terminal 4 a 11)

e) bus de servicio, con un total de 18 líneas.

e1) alimentación con 9 líneas (terminal 1, 2, 20, 50, a 53, 70 y 100)

e2) reloj, con 2 líneas (terminal 24 y 49)

e3) reposición (ó reset), con tres líneas (terminales 54, 75 y 99)

e4) validación de escritura en memoria, con 1 línea (terminal 68)

e5) "esclavo fantasma", con 1 línea (terminal 67)

e6) condiciones especiales, con 2 líneas (terminal 13 y 98)

Quedan las terminales 27, 28, 69 y 71 como reserva

para uso posterior y las líneas 21, 65 y 66 quedan sin definir y quedan ser utilizadas opcionalmente por cada fabricante individual con el requisito de que se suministren con cables punteadores para evitar conflictos y que las señales se limiten a niveles lógicos de 5V. En el apéndice "C" se muestra las especificaciones de cada una de las terminales del bus S-88.

11.5 SISTEMA DE ENTRADA/SALIDA

El sistema de entrada/salida permite la comunicación del microcomputador con el mundo exterior. En general la capacidad y eficiencia en el tratamiento del sistema de E/S es una de las características principales de un sistema de microcomputador.

Se denomina interfaz al sistema hardware-software que permite la comunicación con un periférico determinado, es decir, el conjunto de circuitos y programas que se utilizan para establecer la comunicación.

La forma concreta de realizar una interfaz dependerá de las alternativas que se consideren. En principio, dentro del balance hardware-software se deberá potenciar el software ya que, en general, el incremento en costo de memoria es inferior al incremento en hardware preciso para realizar la

misma función, además de la mayor flexibilidad qe permite el software. En otros casos esta afirmación no estará justificada ya que el costo de programación puede ser decisivo. Por otra parte las entradas y salidas pueden consumir excesivo tiempo de máquina, por lo que puede ser precisa la utilización de circuitería externa, compleja y especializada, a fin de posibilitar la ejecución de estas tareas u otras de proceso que no podrán ejecutarse si el microprocesador tuviese que hacerse cargo de toda la gestión y control de las operaciones de transferencia.

Existen dos tipos de información en la comunicación microprocesador-periférico:

1) datos: entrada de información para proceso y salida de resultados
2) Control: salida de señales para el gobierno de los periféricos y entrada de información del estado de los mismos.

La flexibilidad en el tratamiento de esta información dependerá de varios factores:

a) Instrucciones del microprocesador aplicables al sistema de E/S.
b) Posibilidad de ceder las tareas de control de la

transferencia a circuitos de adaptación externos al microprocesador y controladores de periférico.
c) Técnicas de transferencia utilizadas.

Las funciones que deberá realizar el sistema de interfaz ó adaptación son:
-Identificación de direcciones, a fin de establecer la conexión con el bus de datos y control cuando un acceso concreto de E/S es seleccionado.
- Interpretación de órdenes. En general las órdenes enviadas directamente al sistema de E/S por el microprocesador se reducen a señales de escritura y lectura, ya decodificadas ó que precisan un pretratamiento sencillo. En otros casos, la estructura de E/S es más compleja y se envían a los periféricos señales de control a partir de las cuales se debe interpretar la operación a realizar (lectura, escritura, borrado, prueba, activación, etc.)

- Adaptación física entre los dos sistemas, microcomputadora y periféricos. Esta función es realizada por los transmisores y receptores de líneas, convertidos tensión-corriente y viceversa, optoaisladores, etc. Si la transmisión de los datos se realiza en serie, será preciso considerar la conversión de formato de la información en ambos sentidos.
-Temporización de la transferencia a fin de controlar el

flujo de información entre sistemas de forma ordenada y eficaz. En esta función intervienen los protocolos de comunicación y las técnicas de transferencia.

En la fig. 11.5 se muestra las diferentes estructuras de E/S mediante la cual pueden estar enlazados los periféricos a la CPU.

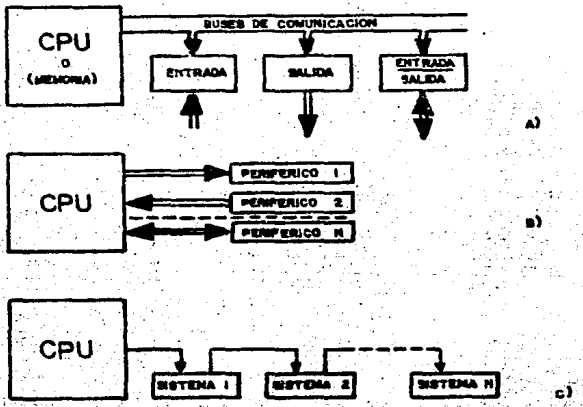


FIGURA 11.5 ESTRUCTURAS DE E/S

11.5.1 ENTRADAS/SALIDAS EN PARALELO

El concepto de "entradas/salidas" engloba toda comunicación ó intercambio de información entre la Unidad Central de Proceso y el exterior, esta comunicación puede efectuarse por dos métodos fundamentales: serie y paralelo.

La entrada/salida en paralelo se basa en la transmisión simutánea de un bits, siendo éstos los componentes de la unidad básica de información codificada ó palabra. Las distintas palabras que componen el bloque de información son transmitidos secuencialmente por las mismas vías

La entrada/salida serie exige una serialización previa y paralelización posterior de la información, que no es preciso efectuar en la entrada/salida en paralelo; como contrapartida en la comunicación en paralelo se precisan las vías de comunicación más las que se requieran para protección y control, implicando por tanto un mayor número de hilos de conexión y circuitos terminales (lo que consituirá la interfaz).

Como consecuencia, cuando la distancia entre conjuntos que deban comunicarse sea reducida, ó bien cuando la velocidad de transmisión deba ser elevada será preferida la transmisión en paralelo.

En el estudio de un sistema de entrada/salida en paralelo deben observarse una serie de aspectos que condicionan el tipo de línea ó circuito terminales que deben utilizarse, tales como: velocidad de transmisión: distancia entre equipos; sentido de la comunicación; interferencia por

ruidos externos ó diafonía entre bits; número de bits componentes de la información, más el correspondiente a las señales de control, más el que se requiera para protección de la información el cual dependerá del grado de seguridad preciso y las características inherentes al periférico y a la información que éste suministre ó recibe.

11.5.2 CIRCUITOS DE SOPORTE PARA MANEJO DE INFORMACION EN PARALELO

AMPLIFICADORES DE BUS:

Son circuitos que permiten expandir la carga admisible (fan-out) de los buses, respetando la polaridad de las señales ó invirtiendola. Las salidas son generalmente de tres estados, Ejemplos pueden ser los 74LS240, 241, y 244 (fig.11.6), amplificadores de bus de 8 bits de 3 estados que requieren una corriente de entrada máxima de Ø.2 mA presentando características de histéresis para mejorar la susceptibilidad a ruidos. Su salida puede entregar hasta 24 mA (aptos para mandar hsta 15 entradas TTL ó 6Ø TTL LS). La diferencia entre los tres circuitos mencionados está en el caracter inversor ó no de sus entradas

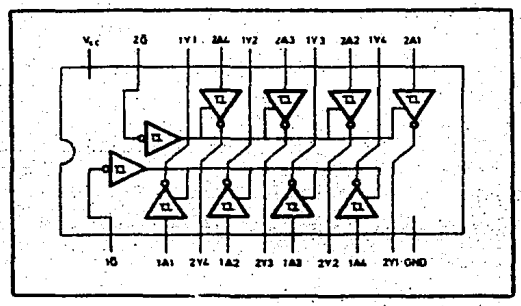


FIGURA 11.6 AMPLIFICADOR DE BUS 74LS240

TRANSCEPTORES

Los circuitos transceptores tienen una cierta similitud con los amplificadores del bus, de los que se distinguen por el hecho de poder amplificar las señales bidireccionalmente, lo que simplifica la conexión entre la mayoría de los microprocesadores cuyo bus de datos es bidireccional y periféricos con características de diálogo.

Estos circuitos presentan generalmente las salidas con características de tres estados y disponen de señales de control que permiten habilitar la amplificación en uno u otro sentido, ó bien sea para señales de control de salida, independientes para cada sentido, o bien por una señal que determina el sentido y otra que autoriza la salida, independientes para cada sentido y otra que autoriza las salidas.

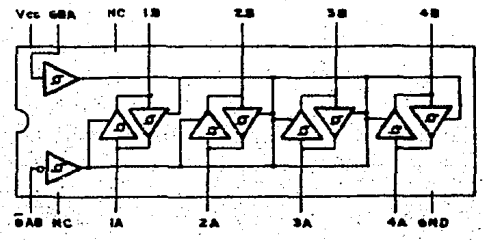


FIGURA 11.7 TRANSCEPTOR DE BUS 74S243

Ejemplos típicos son el 74LS242 y 243 (fig. 11.7) transceptores de 4 bits con autorización de salida en un sentido por la señal GBA en Ø. La diferencia entre uno y otro es que el 242 invierte el bus, mientras que el 243 respeta la polaridad.

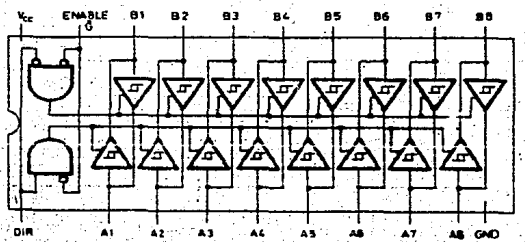


FIGURA 11.8 TRANSCEPTOR DE 8 BITS 74LS245

Como muestra de transceptor de bus de 8 bits se puede tomar el 74LS245 en el que el control de salida se obtiene por la combinación de una señal de autorización y otra de determinación de la dirección de la señal en el bus.

REGISTROS

Un elemento de circuito muy utilizado para

sincronización y memorización de datos de entrada/salida son los registros de 8 bits formado por 8 biestables tipo R-S ó tipo D, con una entrada común de precarga y otra también común de autorización de salidas de 3 estados.

Existe una gran cantidad de modelos distintos en el mercado entre los cuales podemos mencionar los siguientes: 74273, 74363, 74364, 74373, 74374, y 74377 (fig. 11.9)

TRANSCEPTORES-REGISTROS

Un tipo particular de los antes mencionados son los transceptores-registros consistentes en un doble conjunto de 4 biestables bidireccionales con salidas de tres estados.

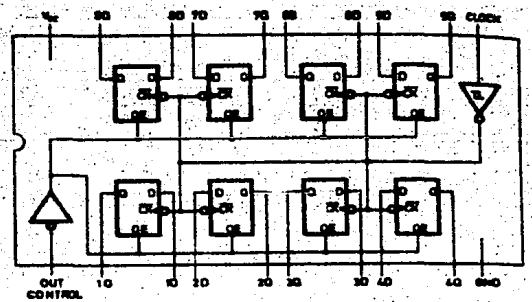


FIGURA 11.9 REGISTRO OCTAL 74LS 373.

El circuito 74S226 nos permite la transferencia bidireccional desde y hacia dos buses; asimismo la doble

memorización y salida pueden ser utilizadas para realizar el intercambio de datos entre ambos buses en el tiempo equivalente de un único impulso de reloj. El almacenamiento de datos se obtiene seleccionando la función de memorización, entrando los datos y poniendo la señal adecuada de validación a nivel Ø. Mientras la validación se mantenga a nivel de cero, los datos son almacenados para la función seleccionada.

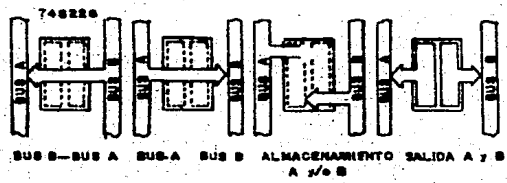
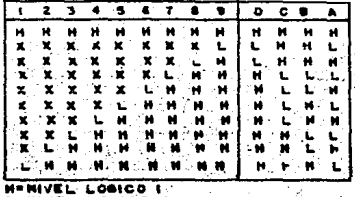


FIGURA 11.1Ø POSIBILIDADES DEL TRANSCEPTOR-REGISTRO 74S226.

Se ofrecen otros modos de atenuación por medio de la posibilidad de utilizar controles de salida independientes, que pueden usarse para autorizar ó desconectar las salidas independientemente de la función de memorización que se haya seleccionado. Las funciones de memorización pueden realizarse con las salidas seleccionadas al estado de alta impedancia. En dicho estado de alta impedancia, las entradas/salidas no ofrecen cargo, ni entregan señal a los buses de modo significativo. Asimismo dispone de entradas P-N-P lo que permite una histéresis típica de 4ØØ milivoltios para mejorar la inmunidad al ruido.

PRIORIZADORES DE INTERRUPCIONES

En los sistemas de microprocesadores en los que se utilizan interrupciones vectorizadas ó jerarquizadas, estan diseñados generalmente para recibir información binaria respecto al elemento periférico que genera la interrupción. Como los circuitos periféricos no estan relacionados entre sí, se requiere intercalar los circuitos apropiados que realicen la codificación idónea al caso. En primera instancia, cualquier circuito codificador podría ser apto, pero se presentan dos tipos de problemas si simultáneamente dos ó más periféricos hacen una solicitud de interrupción. En primer lugar debe de impedirse que, ante varias solicitudes se realice una mezcla de bits que genere una codificación que no sea correspondiente a ninguno de los periféricos solicitantes; por otro lado puede ser necesario el establecimiento de prioridades, de tal modo que ante una solicitud multiple simultánea se establezca un arbitraje determinante de quien se le concede la interrupción en primera instancia.



1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H=NIVEL LOGICO 1
L=NIVEL LOGICO 0
X=INDIFERENTE

FIGURA 11.11 TABLA DE VERDAD DEL PRIORIZADOR 74147

Las dos funciones anteriormente mencionadas pueden ser resueltas por los codificadores prioritarios integrados, ejemplos de los cuales son los circuitos 74147, 74148, 74278, y 74LS348.

COLAS DE ESPERA

Las distintas velocidades de transmisión y aceptación tanto en entradas como en salidas generan una serie de problemas de sincronización. En dichas situaciones se simplifica grandemente la problemática mediante registros de desplazamiento con secuenciamiento de colas de espera (en ingles FIFO, First-In;Firt-Out = primero que entra, primero que sale); estos circuitos consisten en un símil de registros de desplazamiento (shift-registers) con la particularidad de disponer de dos relojes distintos para la carga y la descarga de la información almacenada en forma temporal, con lo que pueden acondicionarse ritmos variados de transferecia entre unidades centrales y unidades periféricas.

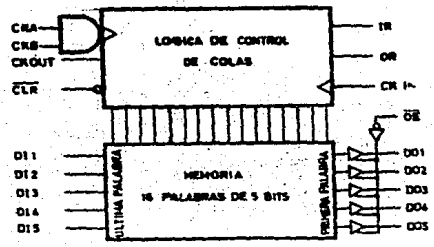


FIGURA 11.12 FIFO 74S225

Un ejemplo de este tipo de circuito es el 74S225 (fig. 11.12). Este circuito es una memoria de colas de 16 palabras de 5 bits. Dispone de tres señales de control. La señal " entrada disponible " indica que la última palabra esta en estado de ocupado, cuando esta llena la memoria. Esta salida esta a nivel 1 cuando la memoria esta disponible para aceptar nuevos datos. La salida de reloj emite un impulso en lógica negativa, sincronizado con el reloj interno, mientras la última posición está libre, facilitando el encadenamiento de etapas. La terminal de "salida disponible", se mantiene a nivel lógico 1 mientras la primera posición de memoria contenga datos válidos y la entrada de reloj de descarga esté a nivel lógico 1. Cuando la entrada de reloj de descarga pasa a nivel Ø, la señal de disponibilidad de salida también pasa a Ø. La primera posición de memoria es aquella desde la cual se dispone de datos a la salida.

La salida de datos (de tres estados) respetan la polaridad de las entradas, con una entrada común de control. Cuando dicha señal de control está a nivel lógico Ø, las salidas de datos son permitidas funcionando como salidas

activas de baja impedancia. Un nivel lógico 1 pone a las salidas en estado de alta impedancia, mientras el resto de entradas y salidas continua siendo activas. También dispone de una entrada de restauración que inválida todos los datos almacenados en la memoria, poniendo a cero la lógica de control.

PUERTOS DE ENTRADA/SALIDA

Estos circuitos son en caso particular de registros de 8 bits y amplificadores de bus, con la adición de un circuito auxiliar para el tratamiento de interrupciones. Ejemplo clásico es el 8212 ó 74S412 (ambos equivalentes entre sí).

Dispone de 8 entradas DI-1 a DI-8 que cargan los ocho biestables cuando las entradas de selección DS1 y DS2 adoptan valores activos y la señal de modalidad de entrada/salida está a nivel 1, ó bien cuando dicha señal es Ø y se activa la validación STB. Los estados de los 8 biestables estarán disponibles en las salidas cuando esten activas simultáneamente las señales de selección DS1 y DS2, ó bien cuando la señal MD (modalidad de entrada/salida) esté a nivel lógico 1. Dispone de un biestable tipo D que entrega señal de Ø a la salida INT cuando las entradas de selección son activas y se recibe una señal de validación STB. Esta señal de estado puede utilizarse para indicar que el registro esta

cargado ó bien para iniciar una secuencia de interrupción. Una entrada auxiliar CTR restaura a Ø todos los bits simultáneamente al ser activada.

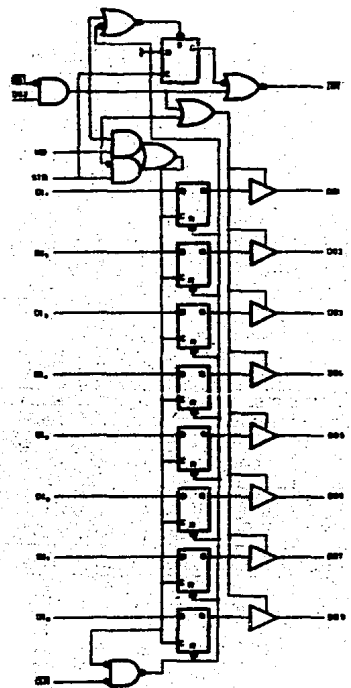


FIGURA 11.13 PUERTO DE 8 BITS DE ENTRADA/SALIDA.

11.6 CONTROLADORES DE ENTRADA/SALIDA EN PARALELO

Los fabricantes de microprocesadores han desarrollado múltiples circuitos especificamente desarrollados para el control de entradas/salidas en parale o dependientes de sus propias unidades centrales, en seguida se enumeran los circuitos más representativos.

- Interfaz programable 8155
- Interfaz universal 8041/8741.
- Controlador de interrupciones programable 8259.
- Controlador de acceso directo a memoria programable 8257
- Controlador de interfaz paralelo Z-80 PIO.
- Adaptador de bus standard 9914.
- Entrada/salida programable 3861.
- Entrada/salida universal 10696.
- Controlador de entrada/salida mN603.
- Controlador de bus de entrada/salida 9442.
- Adaptador de interfaz periférica 6821

11.7 TIPOS DE PERIFERICOS

Los periféricos que pueden conectarse a una microcomputadora pueden dividirse en dos grupos fundamentales: periféricos locales, intimamente ligados al proceso en sí (tales como las memorias RAM, ROM, interruptores, pilotos, visualizadores numéricos y alfanuméricos, temporizadores, relojes de tiempo real, procesdores aritméticos auxiliares, etc.) y periféricos remotos que requieren circuitos específicos de control en ocaciones tanto ó más complejos que la unidad central (tales como: teclados, teletipos, impresoras de alta velocidad, visualizadores en TRC, cintas magnéticas, discos flexibles, discos magnéticos, otros microprocesadores, instrumentos de medida, generadores de señales, sensores analógicos, actuadores electromecánicos, etc).

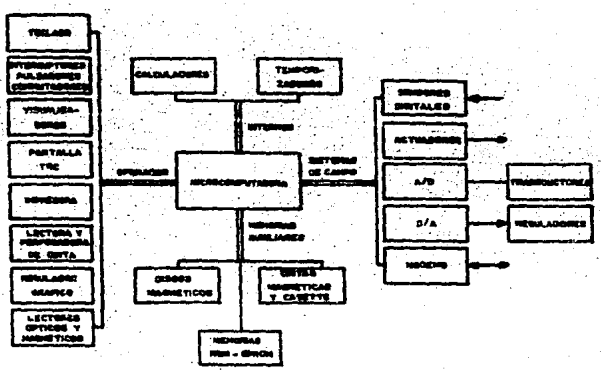


FIGURA 11.14 TIPOS DE PERIFERICOS DE UN SISTEMA MICROCOMPUTADOR

11.7.1 SALIDAS POR IMPRESORA

Las impresoras son los dispositivos primarios de salida útiles para preparar los documentos que serán usados por personas. Las que se fabrican actualmente pueden clasificarse de acuerdo con la forma y velocidad que imprimen.

Impresoras de caracter (ó en serie). Las impresoras de caracteres son dispositivos que imprimen un caracter a la vez. Son utilizadas en mirocomputadoras, minicomputdoras y terminales de teleimpresora para trabajos de impresión de bajo volumen. La técnica empleada para imprimir los caracteres pueden variar de un sistema a otro. Los sistemas de impacto emplean el método ya conocido de la máquina de escribir, en la cual golpea la cara de un tipo contra una cinta entintada que toca el papel. A menudo las impresoras de impacto en serie emplean un mecanismo de impresión de rueda de margarita ó una de matriz de punto. La rueda de margarita para impresión puede quitarse facilmente y reemplazarse con una rueda que tenga distintos tipos. La velocidad de una impresora de rueda de margarita oscila entre 25 y 60 cps. Las velocidades de la impresora por matriz de punto oscilan entre 30 y 350 cps.

En el método de la rueda de margarita cada "petalo"

de la rueda tiene un relieve. Un motor hace girar rápidamente la rueda hasta colocar el caracter deseado en la posición correcta, un martinete de impresión lo golpea para producir la salida. En el método de matriz de puntos, un conjunto de pequeños punzones golpea para producir los caracteres deseados. Cada punzón imprime un pequeño punto.

Generalmente las impresoras de matriz de puntos son más veloces que los dispositivos de rueda de margarita y a menudo resultan menos costosas, pero su calidad de impresión no resulta tan buena.

11.8 SISTEMA DE COMPUTO

Un sistema de cómputo esta formado por: la unidad central de proceso; todos los dispositivos periféricos unidos a ella, y el sistema operativo. Los sitemas de cómputo pueden clasificarse en microcomputadoras, minicomputadoras y computadoras (que corresponde a pequeñas medianas y grandes).

El tamaño adecuado de un sistema de cómputo de determina de acuerdo a la carga total de trabajo proporcionada por el usuario, basandose sobre todo en:

1) el número de terminales de usuarios necesarios.
2) La cantidad y la naturaleza del trabajo que será realizada al mismo tiempo por los usuarios desde cada

terminal (trabajo ligeros=interactivo, trabajo pesado=por lotes).

Y la cantidad necesaria de almacenamiento en disco de línea para contener la información. Por lo general un sistema de cómputo permite la conexión física de una cantidad máxima de memoria y de periféricos.

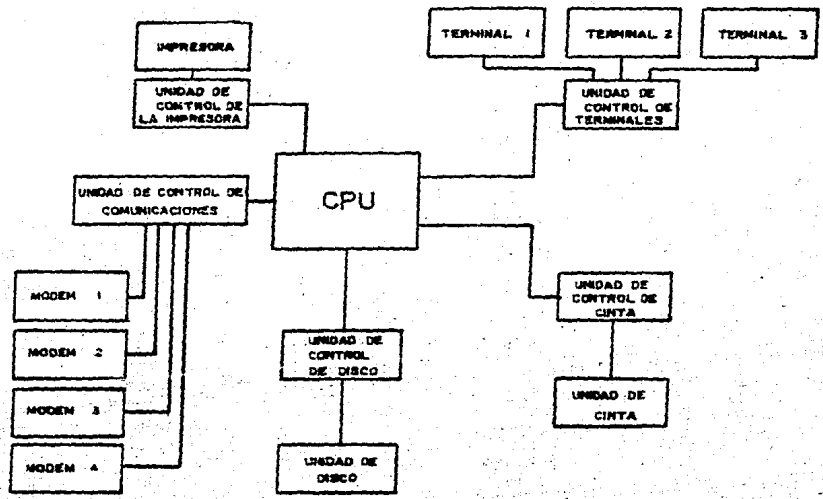


FIGURA 11.15 SISTEMA DE COMPUTO.

11.9 TARJETA PRI

La tarjeta Interface Paralela para Impresora (Printer Interface Board: PRI), nos permite la operación simultanea de la impresora de matriz Cromenco 3703 ó 3779 y de la impresora

de margarita 3355. La tarjeta PRI esta diseñada para trabajar con el bus S-100 y con las características de control de cinta automática para la impresora de forma de margarita.

Aunque los puertos de entrada/salida son diseñados por el fabricante, estos pueden ser cambiados mediante una pequeña modificación por el usuario. La tarjeta PRI viene ensamblada de fábrica (no hay kit de esta para armar).

ESPECIFICACIONES TECNICAS

- Puertos de entrada	54, 5A
- puertos de salida	54, 5A, 5B, 5C
- nota	los 4 bits más significativos- del puerto de direcciones pueden ser cambiados por el usuario mediante una mínima modificación del hardware
- Bus	S-100
- Potencia Requerida	+ 8 Volts; 0.7 amperes
- Medio Ambiente de Operación.	0-55 grados C

11.9.1 ASIGNACION DE PUERTOS

Impresora de matriz modelo 3779 ó 3703

	pata	puerto de salida 54H	ata	puerto de salida 54H
Bit Ø	13	Dato Ø		
Bit 1	25	Dato 1		
Bit 2	12	Dato 2		
Bit 3	24	Dato 3		
Bit 4	11	Dato 4		
Bit 5	23	Dato 5	17	Ocupado
Bit 6	10	Dato 6		
Bit 7	22	Dato de prueba		
Señal de tierra	14		14	

Notas:

a) Los numeros de pata especificados son válidos tanto para el conector EIA DB25 como el final y empiezo del cable Crommenco TU-ART (número de parte TRT-CBL) y el conector J1 dentro de la tarjeta PRI (ver el diagrama)

b) Todas las lineas son de nivel TTL activas en alto excepto DATO DE PRUEBA

c) El DATO DE PRUEBA indica que la información de datos en ASCII en las lineas Ø-6 están listas (este no es un bit de paridad)

d) OCUPADO indica que la impresora no esta lista para aceptar entradas.

11.9.2 TEORIA DE OPERACION

La tarjeta interface de impresora es accesada por la CPU como un dispositivo de entrada/salida standard. Cuando el procesador hace una salida a la tarjeta, el número del puerto de E/S es puesto en los 8 bits más bajos del bus de dirección S-100 de A0 a A7. La señal SOUT (pata 45 del bus S-100) está saliendo al mismo tiempo.

La señal OUT en la tarjeta PRI estará activa (bajo). Los 4 bits menos significativos (A0 hasta A3) del bus de dirección son decodificados por IC11 e IC18, los cuales sólo reconocen las direcciones 04H, 0AH, 08H y 0CH, las demás direcciones no las reconoce.

Con lo que respecta a los 4 bits de dirección más significativos (A4 hasta A7) son decodificacados por IC6 de la tarjeta PRI (el valor faltante es 05H;no obstante esta puede cambiarse contando 2 pistas y añadiendo un interruptor DIP, ver la sección de cambio de asignamiento de puerto).

Cuando un número correcto del puerto de salida es decodificado (uno de 54H, 5AH, 5BH ó 5CH), en la pata 3 ó pata 11 de IC10 tendrá un nivel bajo.

Con esta conjunción de la señal OUT causa que una de

las 4 salidas de IC5 se vaya a un nivel bajo y habilita uno de los 8 bits del registro IC1 hasta IC4. El dato esta disponible para la impresora en cualquiera de los 2 conectores J1y J2.

Cuando el procesador hace una entrada desde la tarjeta PRI, tambien coloca 8 bits del puerto de dirección dentro del bus de dirección junto con la señal SINP (pata 46 del bus S-100). Esto da lugar a que la señal IN de la tarjeta PRI habilita las salidas de IC8 ó IC9 respectivamente. La tarjeta PRI coloca entonces el dato de estado de entrada proveniente de la correspondiente impresora sobre el bus de datos (de entrada) S-100, DI0 a DI7.

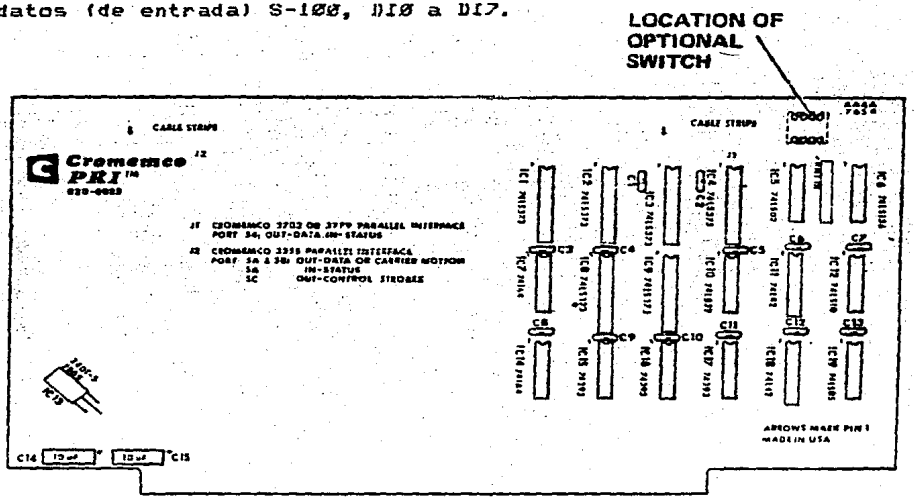


FIGURA 11.16 LOCALIZACION DE LAS PARTES DE LA TARJETA PRI.

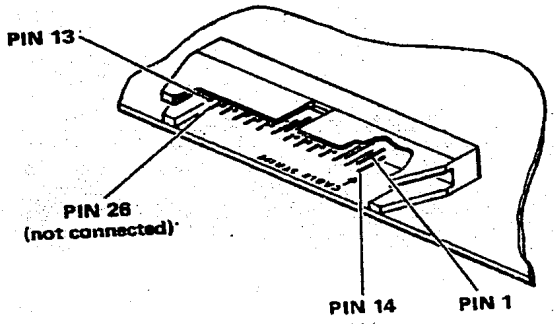


FIGURA 11.17 CONEXION DE PATAS

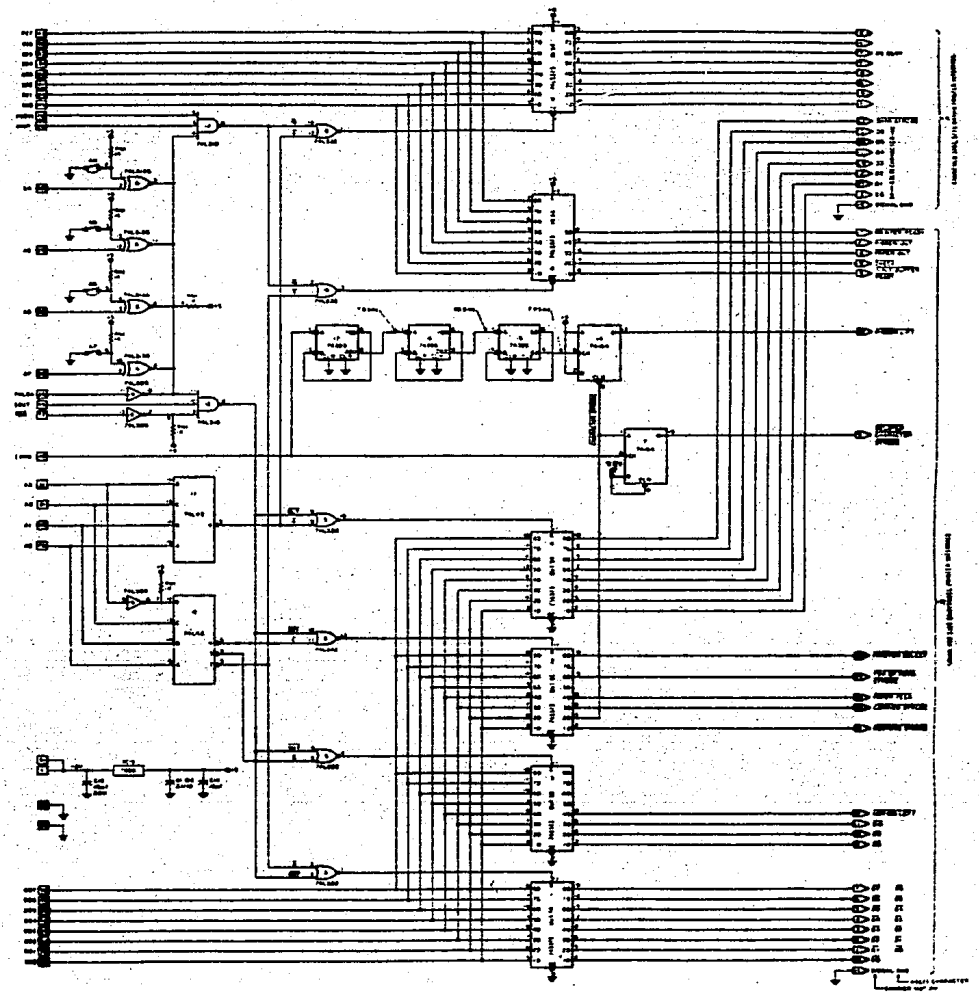


FIGURA 11.18 DIAGRAMA DE LA TARJETA MR1

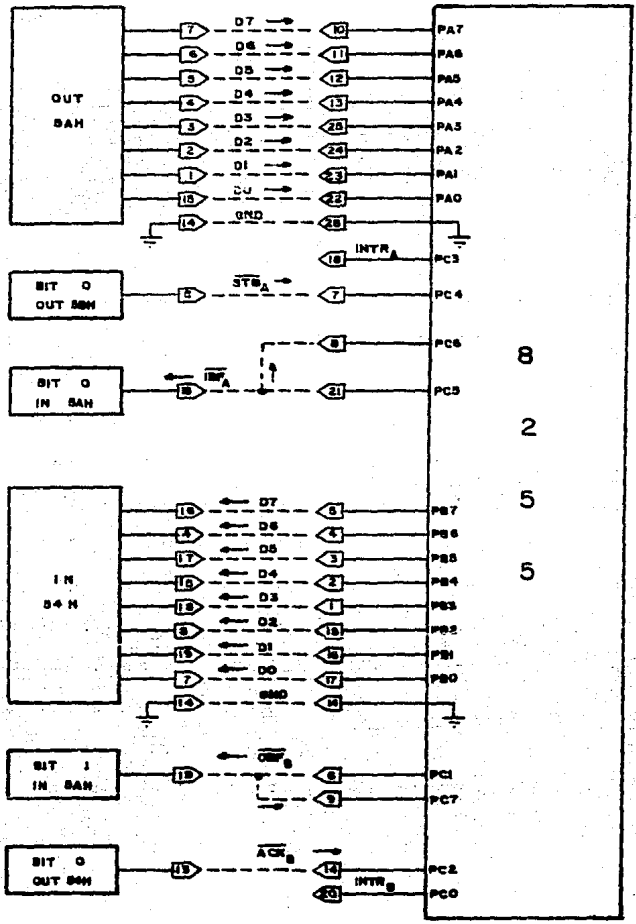


FIGURA 11.19 CONEXION ENTRE LA MICROCOMPUTADORA CROMENCO Y EL MICRO-KIT MK-Z80

11.10 HARDWARE DE COMUNICACION DE DATOS

El que una computadora sirva como una computadora central para un sistema de comunicación de datos en tiempo real y en línea depende tanto de su propia capacidad y características como de la capacidad y características de otras unidades conectadas a ella. Muchas computadoras que actualmente están en el mercado pueden emplearse para redes de comunicación de datos en línea y en tiempo real, condicionado a que el equipo conectado pueda manejar las tareas para las que la computadora central no es eficiente. En otras palabras las características que hacen adecuada a una computadora para la comunicación de datos no necesariamente la hacen buena para el procesamiento de la información que se tenga. En especial, el trabajo de comunicación de datos comprende muchos periodos breves de actividad para dar servicio a un solo caracter de llegada ó salida. Una computadora cuyo hardware ó software entorpezca ó demore éste tipo de operación no se desempeñará bien en el medio de comuniciaciones de datos. Para que esa máquina sea efectiva se necesitará de hardware auxiliar.

Generalmente se pueden clasificar en tres categorías las configuraciones de comunicaciones de datos, que dependen de la interfase entre la red de comunicaciones de datos y las funciones de procesamiento de la computadora central. La

primera de estas categorías es una configuración de computadora aislada, en que se diseña a la computadora para que maneje un conjunto específico de instalaciones y terminales de comunicaciones. Los circuitos para manejar esto se incorporan directamente a la computadora, es decir, la arquitectura del equipo se diseña de tal manera que pueda interactuar en un modo de tiempo real. En la fig. 11.20 muestra una configuración aislada para comunicaciones en que la computadora central pueda manejar todos los protocolos de la comunicación este tipo de computadora es de programa almacenado con posibilidad de comunicacoines y de cómputo. Se emplea típicamente en un modo donde se da mayor énfasis a las comunicaciones que al procesamiento de datos. Con frecuencia se le encuentra en el medio ambiente de la fabricación para el control de procesos, y en áreas donde el usuario consulta una base de datos acerca del estado de determinado producto, nivel de inventario u otras consultas parecidas. Este campo esta dominado por minicomputadoras que se han desarrollado y programado para el procesamiento de propósito especial y funciones de comunicaciones.

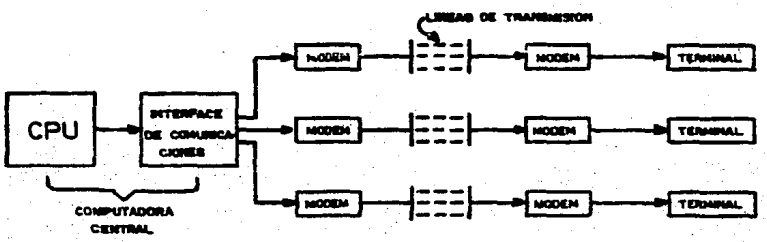


FIGURA 11.20 CONFIGURACION DE COMUNICACIONES AISLADAS.

La fig. 11.21 muestra una configuración de computadora delantera/central.

Este tipo de configuración se emplea primordialmente cuando los requerimientos de entrada y salida y de procesamiento de cómputo son muy grandes además de necesitarse tiempos rápidos de respuesta. Este es el tipo de configuración usado en todas las grandes redes de comunicación de datos.

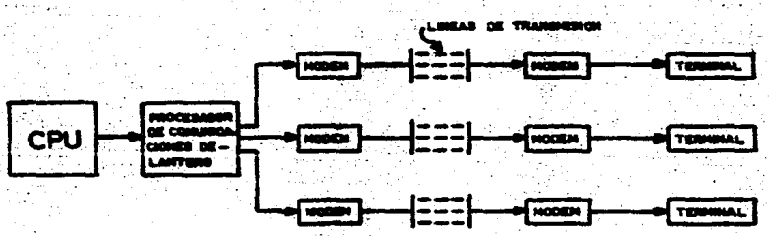


FIGURA 11.21 CONFIGURACION DE COMPUTADORA DELANTERA/CENTRAL

A veces, la distinción entre las tres categorías de

comunicaciones de datos anteriores, no son claras debido al traslape que hay en estas categorías en el médio actual de las redes. Por ejemplo, ocasionalmente se puede utilizar la configuración de comunicaciones aislada (primera categoría) como el procesador de comunicaciones delantero. La configurción de la computadora de propósito general (segunda categoría) es por lo general la computadora principal de procesador de comunicaciones delantero (tercera categoría). En la fig. 11.22 se muestra una red punto a punto junto con una computadora central, procesador de comunicaciones delantero modems y terminales.

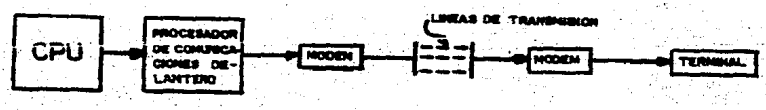


FIGURA 11.22 RED DE PUNTO A PUNTO.

La segunda categoría es la de las computadoras de propósito general, las que generalmente no incorporan hardware de interface para comunicaciones. Este tipo de computadoras puede manejar una red pequeña de comunicaciones de datos. Sin embargo a medida que aumenta el número de terminales, la computadora de propósito general llega al

punto en que se degrada mucho el rendimiento por su incapacidad para manejar la parte de trabajo que se refiere a comunicaciones. Un ejemplo de esta configuración lo tienen las organizaciones que tienen una computadora grande de propósito general empleada tanto para el procesamiento de lotes como para ciertas funciones de tiempo compartido dentro de la organización.

La tercera categoría, que se está popularizando más en la actualidad es la configuración delantera, en la que se emplea una computadora grande de propósito general tanto para la comunicación de datos como para cierto procesamiento en lotes, pero dando mayor énfasis a la porción del sistema encargada de la comuniciación de datos en línea y tiempo real. En esta configuración, hay una división clara de trabajo entre el módulo delantero y la computadora de propósito general.

El módulo delantero puede tener dos formas: la primera es una unidad de control de comunicaciones no programable, con alambrado permanente diseñada por el fabricante de la computadora para adaptar las características de la línea y terminal específicas a la computadora. La segunda forma es la de un procesador de comunicaciones delantero programable que pueda manejar algunas ó todas las actividades de entrada/salida y además cierto procesamiento.

11.10.1 PROCESADORES DE COMUNICACIONES DELANTEROS

Estos dispositivos pueden ser no programables ó completamente programables. A los no programables también se les conoce como controladores de comunicaciones ó unidades de control de transmisión. Estas unidades son dispositivos de manejo de datos que controlan la transmisión de datos entre una computadora central y terminales remotas. Esta unidades de control de comunicaciones de alambrado permanente realizan el máximo número posible de funciones relacionadas con las comunicaciones para aliviar la carga de la computadora central. La tendencia actual esta dirigida hacia las unidades programables, pues los no programables no tienen la suficiente flexibilidad para satisfacer las necesidades actuales. La única ventaja que presentan los dispositivos no programables es la velocidad al que pueden trabajar.

Los procesadores de comunicaciones delanteros programables son dispositivos complejos, parecidos a computadoras. De hecho, muchos de estos procesadores son minicomputadoras complejas que se pueden comprar con diversas opciones, funciones y lógica incorporada. Las unidades programables son mucho más poderosas que una unidad de control de comunicaciones alambrada y ofrecen una extensa variedad de características y funciones que ofrece un procesador de comunicaciones delantero programable.

11.10.2 DISPOSITIVOS PARA COMPARTIR UN PUERTO

Todos los procesadores de comunicaciones delanteros tienen una capacidad máxima de circuitos. Si se diseña un procesador de comunicaciones delantero para que maneje hasta 100 puertos, ello significa que se le pueden conectar hasta 100 circuitos. Siempre que se excede la capacidad diseñada de un procesador de comunicaciones delantero, se puede utilizar un dispositivo para compartir puertos.

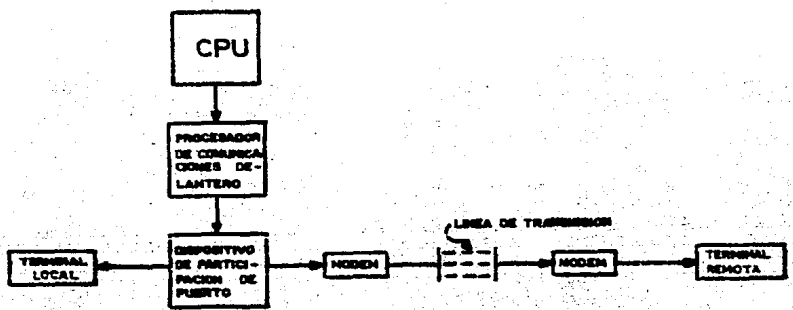


FIGURA 11.23 DISPOSITIVO PARA COMPARTIR UN PUERTO

Un dispositivo para compartir puertos, ó "puente", trata a varias líneas de punto a punto como si fuera una sola línea de puntos múltiples, conservando de esta manera los puertos acosta de permitir que sólo una terminal transmite a la computadora a la vez. La fig. 11.23 muestra un dispositivo para compartir puertos con una terminal local (la

terminal local no necesita modem) y una terminal remota conectadas. Extisten dispcsitivos para compartir puertos que permiten conectar varias terminales de baja velocidad (locales ó remotas). Este dispositivo se puede utilizar para no tener que instalar un segundo procesador de comunicaciones delantero cuando estan ocupados todos los puertos del primero. Esta solución puede no ser a largo plazo, permitiendo solucionar el programa hasta que pueda configurarse una nueva red ó se compre un equipo nuevo. Los dispositivos para compartir puertos son puentes digitales porque se instalan del lado digital del módem. Por otra parte, los puentes analógicos se utilizan del lado analógio del módem siempre que se desee que dos ó más líneas de punto a punto se comporten como una línea de puntos múltiples. En la práctica, la portadora común (compañia de comunicaciones) sirve como puente entre líneas de punto a punto, siempre que un cliente ordene una línea de puntos múltiples.

11.10.3 DIVISOR DE LINEA

Un divisor de línea es parecido a un dispositivo para compartir puertos excepto porque se localiza en el extremo remoto de la línea (fig. 11.24). Es un "interruptor" que trata a varias terminales como si fueran una línea de puntos múltiples. Las tres terminales de la fig. 11.24 comparten la capacidad total (velocidad de transmisión en bits por

segundo) del enlace, es decir que el divisor de lineas comparte el enlace entre las cuatro terminales.

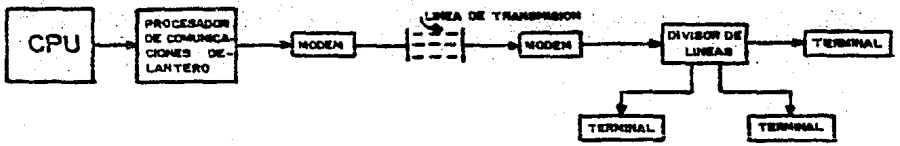


FIGURA 11.24 DIVISOR DE LINEAS

CONCLUSIONES

El presente trabajo, aparte de dar solución a un problema (comunicación entre el KIT y la microcomputadora CROMENCO), proporcionó las bases que nos permiten comprender la forma en que se puede entablar comunicación entre dos sistemas de cómputo.

Como resultado del estudio anterior se obtuvo un programa que nos permite entablar comunicación en paralelo, entre la microcomputadora CROMENCO y el KIT MKE-Z80. Esta comunicación se realizó a través de los puertos de Entrada/Salida de la tarjeta PRI de la microcomputadora y el PPI 8255 del KIT.

Inicialmente se tenían tres posibilidades: una de ellas consistía en fabricar una tarjeta, con un PPI 8255, que se conectaría directamente al bus del sistema CROMENCO. Esta idea se desechó ante la posibilidad que existía de dañar el sistema cuando se hicieran los primeras pruebas.

Otra opción era conectarse con la CROMENCO a través de la tarjeta de interface D+7A E/S. Esta tenía el incoveniente de que emplean algunas señales analógicas lo que complicaba el trabajo, entre otras cosas porque se requería de refresco de datos, ó de que los datos se tenían que

manejar por nibles lo que haría más lento el funcionamiento del sistema, y más complicada la programación del mismo.

Por último el dispositivo que proporcionaba las mejores características, para ser conectado con el PPI 8255 del KIT, era la tarjeta de interface de impresora PRI, del sistema CROMENCO.

Resulta importante mencionar que en la etapa de análisis se consideró que los subprogramas que maneja el KIT eran estructurados, lo cual no resulto cierto. Esto constituyó el principal problema en la elaboración del programa que finalmente resulto ser no estructurado.

En la tésis se hizo un especial énfasis en la comunicación de datos tomando en cuenta la importancia que tendrán las comunicaciones en el futuro. Se pronostica que para fines de este siglo, cuando los sistemas conocidos con el nombre genérico de VIDEOTEX, cobren popularidad, las ciudades interconectadas, donde las redes de comunicación uniran cada hogar con los bancos de información de las computadoras centrales, ya no serán temas de ciencia ficción. Los avances en el diseño de computadoras, las notables reduccoiones en el costo por operación junto con las ideas creativas en las aplicaciones de una computadora, han incrementado el uso de los sistemas de comunicación de datos

para transmitir información entre sitios comerciales muy separados y las computadoras y equipo terminal instalado en estos sitios.

En la actualidad están en servicio redes de computacion/comunicación muy grandes y es compleja la coordinación requerida para el uso eficiente de esas redes, tales redes tienen cientos de terminales y muchos procesadores pequeños localizados en sitios dispersos. A través de diferentes canales de transmisión, estos sitios se ligan a su vez a los computadores anfitriones de mayor tamaño. La tarea de los diseñadores de las redes consisten en seleccionar y coordinar los componentes de la red de tal manera que los datos necesarios sean transladados al lugar y en el momento adecuado, con el menor número de errores y al menor costo posible.

La interconexión entre la CROMEMCO y el KIT representa el primer paso dentro de un cúmulo de posibilidades, entre las que se encuentra la creación de una red en la que la microcomputadora CROMEMCO tomaría el papel de computadora anfitrion (HOST), el KIT conectado en paralelo fungiría como un procesador de comunicaciones delantero y los restantes KITS como terminales. En el apéndice "D" se muestran las caracteristicas y funciones de un procesador de comunicaciones delantero. Esta información

puede ser tomada como punto de partida para la creación de dicha red.

BIBLIOGRAFIA

- CDOS DISK OPERATING SYSTEM MANUAL
  Cromemco

- PRI INTERFACE
  Manual Cromemco

- MKE Z80
  Manual del Usuario Microkits

- MACRO ASSEMBLER
  Manual Cromemco

- MICROELECTRONICS DIGITAL AND ANALOG CIRCUITS AND SYSTEMS
  Jacob Millman
  Edit. McGraw Hill

- HANDBOOK OF ELECTRONICS CALCULATIONS
  Kaufman Milton/Arthur H. Seidman

- LOGIC DATA BOOK
National Semiconductor

- THE TTL DATA BOOK FOR DESINGN ENGINEERS
Texas Instruments

- INFORMATION TRANSMITION MODULATION
Schwartz, Misha
Edit. McGraw Hill

- TECHNICAL ASPECTS OF DATA COMUNICATION
McNamara Jhon
Edit. Digital

- APUNTES PARA EL USO DEL EQUIPO CROMEMCO
Centro de Servicios de Computo
UNAM

- MICROPROCESADORES 8080 E INTERFACES
M.C. Octavio F. Garcia Narcia
ESIME IPN.

- AN INTRODUCCION TO MICROCOMPUTERS
Jerry Kame y Adam Osborne
Edit. McGraw Hill

- DATA COMUNICACION AND TELEPROCESING SYSTEMS
Tevor Housley
Edit. Pretinse Hall

- CIRCUITOS DIGITALES Y MICROPROCESADORES
Herbert Taub
Edit. McGraw Hill

- INTRODUCCION A LA CIENCIA DE LA COMPUTACION
Vladimir Zwass.
Editorial CESA.

- PRINCIPIOS DIGITALES
Roger L. Tokhem
Edit. Serie Schaum

- INFORMATICA PRESENTE Y FUTURO
Donald H. Sanders

- CIENCIA Y DESARROLLO
Revistas Varias
Consejo Nacional de Ciencia y Tecnologia

- INFORMACION CIENTIFICA Y TECNOLOGICA
Revistas Varias
Consejo Nacional de Ciencia y Tecnologia

- ENCICLOPEDIA DE LA INFORMATICA

  Revistas Varias

  Editorial Origen S.A.

- MI COMPUTER

  Revistas Varias

  Editorial Artemisa.

- ENCICLOPEDIA PRACTICA DE LA INFORMATICA

  Editorial Nueva Lente.

  Revistas Varias

APENDICES

APENDICE A

INSTRUCCIONES DEL Z80
Y
ESPECIFICACIONES ELECTRICAS

8-BIT LOAD GROUP

Mnemonic	Symbolic Operation	Flags* S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD r, s	r — s	•	•	X	•	X	•	•	•	01 r s		1	1	4	r, s Reg.
LD r, n	r — n	•	•	X	•	X	•	•	•	00 r 110 — n —		2	2	7	000 B 001 C
LD r, (HL)	r — (HL)	•	•	X	•	X	•	•	•	01 r 110		1	2	7	010 D
LD r, (IX+d)	r — (IX+d)	•	•	X	•	X	•	•	•	11 011 101 01 r 110 — d —	DD	3	5	19	011 E 100 H 101 L
LD r, (IY+d)	r — (IY+d)	•	•	X	•	X	•	•	•	11 111 101 01 r 110 — d —	FD	3	5	19	111 A
LD (HL), r	(HL) — r	•	•	X	•	X	•	•	•	01 110 r		1	2	7	
LD (IX+d), r	(IX+d) — r	•	•	X	•	X	•	•	•	11 011 101 01 110 r — d —	DD	3	5	19	
LD (IY+d), r	(IY+d) — r	•	•	X	•	X	•	•	•	11 111 101 01 110 r — d —	FD	3	5	19	
LD (HL), n	(HL) — n	•	•	X	•	X	•	•	•	00 110 110 — n —	36	2	3	10	
LD (IX+d), n	(IX+d) — n	•	•	X	•	X	•	•	•	11 011 101 00 110 110 — d — — n —	DD 36	4	5	19	
LD (IY+d), n	(IY+d) — n	•	•	X	•	X	•	•	•	11 111 101 00 110 110 — d — — n —	FD 36	4	5	19	
LD A, (BC)	A — (BC)	•	•	X	•	X	•	•	•	00 001 010	0A	1	2	7	
LD A, (DE)	A — (DE)	•	•	X	•	X	•	•	•	00 011 010	1A	1	2	7	
LD A, (nn)	A — (nn)	•	•	X	•	X	•	•	•	00 111 010 — n — — n —	3A	3	4	13	
LD (BC), A	(BC) — A	•	•	X	•	X	•	•	•	00 000 010	02	1	2	7	
LD (DE), A	(DE) — A	•	•	X	•	X	•	•	•	00 010 010	12	1	2	7	
LD (nn), A	(nn) — A	•	•	X	•	X	•	•	•	00 110 010 — n — — n —	32	3	4	13	
LD A, I	A — I	↕	↕	X	0	X	IFF	0	•	11 101 101 01 010 111	ED 57	2	2	9	
LD A, R	A — R	↕	↕	X	0	X	IFF	0	•	11 101 101 01 011 111	ED 5F	2	2	9	
LD I, A	I — A	•	•	X	•	X	•	•	•	11 101 101 01 000 111	ED 47	2	2	9	
LD R, A	R — A	•	•	X	•	X	•	•	•	11 101 101 01 001 111	ED 4F	2	2	9	

Notes: r, s means any of the registers A, B, C, D, E, H, L
IFF the content of the interrupt enable flip-flop (IFF) is copied into the P/V flag

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

EXCHANGE GROUP AND BLOCK TRANSFER AND SEARCH GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE, HL	DE↔HL	•	•	X	•	X	•	•	•	11 101 011	EB	1	1	4	
EX AF, AF'	AF↔AF'	•	•	X	•	X	•	•	•	00 001 000	08	1	1	4	
EXX	BC↔BC' DE↔DE' HL↔HL'	•	•	X	•	X	•	•	•	11 011 001	D9	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔(SP+1) L ↔(SP)	•	•	X	•	X	•	•	•	11 100 011	E3	1	5	19	
EX (SP), IX	IX_H ↔(SP+1) IX_L ↔(SP)	•	•	X	•	X	•	•	•	11 011 101 11 100 011	DD E3	2	6	23	
EX (SP), IY	IY_H ↔(SP+1) IY_L ↔(SP)	•	•	X	•	X	•	•	•	11 111 101 11 100 011	FD E3	2	6	23	
LDI	(DE)←(HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	•	•	X	0	X	① ↕	0	•	11 101 101 10 100 000	ED A0	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE)←(HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 110 000	ED B0	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
LDD	(DE)←(HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	•	•	X	0	X	① ↕	0	•	11 101 101 10 101 000	ED A8	2	4	16	
LDDR	(DE)←(HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 111 000	ED B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A − (HL) HL ← HL+1 BC ← BC-1	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 100 001	ED A1	2	4	16	
CPIR	A − (HL) HL ← HL+1 BC ← BC-1 Repeat until A = (HL) or BC = 0	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 110 001	ED B1	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A − (HL) HL ← HL-1 BC ← BC-1	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 101 001	ED A9	2	4	16	
CPDR	A − (HL) HL ← HL-1 BC ← BC-1 Repeat until A = (HL) or BC = 0	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 111 001	ED B9	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)

Notes: ① P/V flag is 0 if the result of BC-1 = 0, otherwise P/V = 1
② Z flag is 1 if A = (HL), otherwise Z = 0.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
↕ = flag is affected according to the result of the operation.

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00	dd0	001		3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
										—	n	—					
										—	n	—					
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11	011	101	DD	4	4	14	
										00	100	001	21				
										—	n	—					
										—	n	—					
LD IY, nn	IY ← nn'	•	•	X	•	X	•	•	•	11	111	101	FD	4	4	14	
										00	100	001	21				
										—	n	—					
										—	n	—					
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	X	•	X	•	•	•	00	101	010	2A	3	5	16	
										—	n	—					
										—	n	—					
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
										01	dd1	011					
										—	n	—					
										—	n	—					
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20	
										00	101	010	2A				
										—	n	—					
										—	n	—					
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20	
										00	101	010	2A				
										—	n	—					
										—	n	—					
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	X	•	X	•	•	•	00	100	010	22	3	5	16	
										—	n	—					
										—	n	—					
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
										01	dd0	011					
										—	n	—					
										—	n	—					
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20	
										00	100	010	22				
										—	n	—					
										—	n	—					
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20	
										00	100	010	22				
										—	n	—					
										—	n	—					
LD SP, HL	SP ← HL	•	•	X	•	X	•	•	•	11	111	001	F9	1	1	6	
LD SP, IX	SP ← IX	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	10	
										11	111	001	F9				
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10	
										11	111	001	F9				qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH qq	(SP-2) ← qqL (SP-1) ← qqH	•	•	X	•	X	•	•	•	11	qq0	101		1	3	11	
PUSH IX	(SP-2) ← IXL (SP-1) ← IX_H	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	15	
										11	100	101	E5				
PUSH IY	(SP-2) ← IY_L (SP-1) ← IY_H	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	15	
										11	100	101	E5				
POP qq	qq_H ← (SP+1) qq_L ← (SP)	•	•	X	•	X	•	•	•	11	qq0	001		1	3	10	
POP IX	IX_H ← (SP+1) IXL ← (SP)	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	14	
										11	100	001	E1				
POP IY	IY_H ← (SP+1) IY_L ← (SP)	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	14	
										11	100	001	E1				

Notes: dd is any of the register pairs BC, DE, HL, SP
qq is any of the register pairs AF, BC, DE, HL
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively.
e.g. BC_L = C, AF_H = A

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ flag is affected according to the result of the operation.

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD A, r	A ← A + r	‡	‡	X	‡		X	V	0	‡	10 000 r		1	1	4	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD A, n	A ← A + n	‡	‡	X	‡		X	V	0	‡	11 000 110 – n –		2	2	7	
ADD A, (HL)	A ← A + (HL)	‡	‡	X	‡		X	V	0	‡	10 000 110		1	2	7	
ADD A, (IX+d)	A ← A + (IX+d)	‡	‡	X	‡		X	V	0	‡	11 011 101 10 000 110 – d –	DD	3	5	19	
ADD A, (IY+d)	A ← A + (IY+d)	‡	‡	X	‡		X	V	0	‡	11 111 101 10 000 110 – d –	FD	3	5	19	
ADC A, s	A ← A + s + CY	‡	‡	X	‡		X	V	0	‡	001					s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the 000 in the ADD set above.
SUB s	A ← A – s	‡	‡	X	‡		X	V	1	‡	010					
SBC A, s	A ← A – s – CY	‡	‡	X	‡		X	V	1	‡	011					
AND s	A ← A ∧ s	‡	‡	X	1		X	P	0	0	100					
OR s	A ← A ∨ s	‡	‡	X	0		X	P	0	0	110					
XOR s	A ← A ⊻ s	‡	‡	X	0		X	P	0	0	101					
CP s	A – s	‡	‡	X	‡		X	V	1	‡	111					
INC r	r ← r + 1	‡	‡	X	‡		X	V	0	•	00 r 100		1	1	4	
INC (HL)	(HL) ← (HL) + 1	‡	‡	X	‡		X	V	0	•	00 110 100		1	3	11	
INC (IX+d)	(IX+d) ← (IX+d) + 1	‡	‡	X	‡		X	V	0	•	11 011 101 00 110 100 – d –	DD	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d) + 1	‡	‡	X	‡		X	V	0	•	11 111 101 00 110 100 – d –	FD	3	6	23	
DEC s	s ← s – 1	‡	‡	X	‡		X	V	1	•	101					s is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in OP Code.

Notes: The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity. V = 1 means overflow, V = 0 means not overflow, P = 1 means parity of the result is even, P = 0 means parity of the result is odd.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
‡ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands	‡	‡	X	‡	X	P	•	‡	00 100 111	27	1	1	4	Decimal adjust accumulator
CPL	$A - \overline{A}$	•	•	X	1	X	•	1	•	00 101 111	2F	1	1	4	Complement accumulator (One's complement)
NEG	$A - \overline{A}+1$	‡	‡	X	‡	X	V	1	‡	11 101 101 01 000 100	ED 44	2	2	8	Negate acc, (two's complement)
CCF	$CY - \overline{CY}$	•	•	X	X	X	•	0	‡	00 111 111	3F	1	1	4	Complement carry flag
SCF	CY – 1	•	•	X	0	X	•	0	1	00 110 111	37	1	1	4	Set carry flag
NOP	No operation	•	•	X	•	X	•	•	•	00 000 000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01 110 110	76	1	1	4	
DI*	IFF – 0	•	•	X	•	X	•	•	•	1: 110 011	F3	1	1	4	
EI*	IFF – 1	•	•	X	•	X	•	•	•	11 111 011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 101 101 01 000 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 101 101 01 010 110	ED 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 01 011 110	ED 5E	2	2	8	

Notes: IFF indicates the interrupt enable flip-flop
CY indicates the carry flip-flop.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
‡ = flag is affected according to the result of the operation.

*Interrupts are not sampled at the end of EI or DI

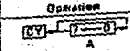
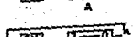
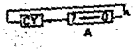
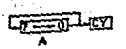
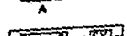
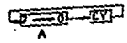
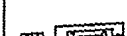
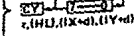
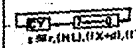
16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL,ss	HL ← HL+ss	•	•	X	X	X	•	0	↕	00 ss1 001		1	3	11	ss Reg.
															00 BC
ADC HL,ss	HL ← HL+ss+CY	↕	↕	X	X	X	V	0	↕	11 101 101	ED	2	4	15	01 DE
										01 ss1 010					10 HL
															11 SP
SBC HL,ss	HL ← HL-ss-CY	↕	↕	X	X	X	V	1	↕	11 101 101	ED	2	4	15	
										01 ss0 010					
ADD IX,pp	IX ← IX + pp	•	•	X	X	X	•	0	↕	11 011 101	DD	2	4	15	pp Reg.
										00 pp1 001					00 BC
															01 DE
															10 IX
															11 SP
ADD IY,rr	IY ← IY + rr	•	•	X	X	X	•	0	↕	11 111 101	FD	2	4	15	rr Reg.
										00 rr1 001					00 BC
															01 DE
															10 IY
															11 SP
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	•	00 ss0 011		1	1	6	
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	•	11 011 101	DD	2	2	10	
										00 100 011	23				
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11 111 101	FD	2	2	10	
										00 100 011	23				
DEC ss	ss ← ss - 1	•	•	X	•	X	•	•	•	00 ss1 011		1	1	6	
DEC IX	IX ← IX - 1	•	•	X	•	X	•	•	•	11 011 101	DD	2	2	10	
										00 101 011	2B				
DEC IY	IY ← IY - 1	•	•	X	•	X	•	•	•	11 111 101	FD	2	2	10	
										00 101 011	2B				

Notes: ss is any of the register pairs BC, DE, HL, SP
pp is any of the register pairs BC, DE, IX, SP
rr is any of the register pairs BC, DE, IY, SP.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
↕ = flag is affected according to the result of the operation.

ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		•	•	X	0	X	•	0	↕	00 000 111	07	1	1	4	Rotate left circular accumulator
RLA		•	•	X	0	X	•	0	↕	00 010 111	17	1	1	4	Rotate left accumulator
RRCA		•	•	X	0	X	•	0	↕	00 001 111	0F	1	1	4	Rotate right circular accumulator
RRA		•	•	X	0	X	•	0	↕	00 011 111	1F	1	1	4	Rotate right accumulator
RLC r		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 r	CB	2	2	8	Rotate left circular register r
RLC (HL)		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 110	CB 06	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	 r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	11 011 101 11 001 011 – d – 00 000 110	DD CB 06	4	6	23	
RLC (IY+d)		↕	↕	X	0	X	P	0	↕	11 111 101 11 001 011 – d – 00 000 110	FD CB 06	4	6	23	
RL s	 s ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	00 010					Instruction format and states are as shown for RLC's. To form new Op Code replace 000 of RLC's with shown code
RRC s	 s ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	00 001					
RR s	 s ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	00 011					
SLA s	 s ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	00 100					
SRA s	 s ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	00 101					
SRL s	 s ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	00 111					
RLD		↕	↕	X	0	X	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected
RRD		↕	↕	X	0	X	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b, r	$Z \leftarrow \overline{r_b}$	X	↕	X	1	X	X	0	•	11	001	011	CB	2	2	8	r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A; b: Bit Tested 000 0, 001 1, 010 2, 011 3, 100 4, 101 5, 110 6, 111 7
										01	b	r					
BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	X	↕	X	1	X	X	0	•	11	001	011	CB	2	3	12	
										01	b	110					
BIT b, (IX+d)$_b$	$Z \leftarrow \overline{(IX+d)_b}$	X	↕	X	1	X	X	0	•	11	011	101	DD	4	5	20	
										11	001	011	CB				
										–	d	–					
										01	b	110					
BIT b, (IY+d)$_b$	$Z \leftarrow \overline{(IY+d)_b}$	X	↕	X	1	X	X	0	•	11	111	101	FD	4	5	20	
										11	001	011	CB				
										–	d	–					
										01	b	110					
SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	001	011	CB	2	2	8	
										11	b	r					
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	001	011	CB	2	4	15	
										11	b	110					
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	23	
										11	001	011	CB				
										–	d	–					
										11	b	110					
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	23	
										11	001	011	CB				
										–	d	–					
										11	b	110					
RES b, s	$s_b \leftarrow 0$ (s ≡ r, (HL), (IX+d), (IY+d))	•	•	X	•	X	•	•	•	10							To form new Op-Code replace 11 of SET b, s with 10. Flags and time states for SET instruction

Notes: The notation s_b indicates bit b (0 to 7) of location s.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

JUMP GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11 000 011 ← n → ← n →	C3	3	3	10	
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	X	•	X	•	•	•	11 cc 010 ← n → ← n →		3	3	10	cc | Condition 000 | NZ non zero 001 | Z zero 010 | NC non carry 011 | C carry 100 | PO parity odd 101 | PE parity even 110 | P sign positive 111 | M sign negative
JR e	PC ← PC + e	•	•	X	•	X	•	•	•	00 011 000 ← e-2 →	18	2	3	12	
JR C, e	If C = 0, continue	•	•	X	•	X	•	•	•	00 111 000 ← e-2 →	38	2	2	7	If condition not met
	If C = 1, PC ← PC+e											2	3	12	If condition is met
JR NC, e	If C = 1, continue	•	•	X	•	X	•	•	•	00 110 000 ← e-2 →	30	2	2	7	If condition not met
	If C = 0, PC ← PC+e											2	3	12	If condition is met
JR Z, e	If Z = 0 continue	•	•	X	•	X	•	•	•	00 101 000 ← e-2 →	28	2	2	7	If condition not met
	If Z = 1, PC ← PC+e											2	3	12	If condition is met
JR NZ, e	If Z = 1, continue	•	•	X	•	X	•	•	•	00 100 000 ← e-2 →	20	2	2	7	If condition not met
	If Z = 0, PC ← PC+e											2	3	12	If condition is met
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11 101 001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11 011 101 11 101 001	DD E9	2	2	8	
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	11 111 101 11 101 001	FD E9	2	2	8	
DJNZ, e	B ← B-1 If B = 0, continue	•	•	X	•	X	•	•	•	00 010 000 ← e-2 →	10	2	2	8	If B = 0
	If B ≠ 0, PC ← PC+e											2	3	13	If B ≠ 0

Notes: e represents the extension in the relative addressing mode.

e is a signed two's complement number in the range <-126, 129>

e-2 in the op-code provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

CALL AND RETURN GROUP

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
CALL nn	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L$ $PC \leftarrow nn$	•	•	X	•	X	•	•	•	11 – –	001 n n	101 – –	CD	3	5	17	
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	X	•	X	•	•	•	11 – –	cc n n	100 – –		3 3	3 5	10 17	If cc is false If cc is true
RET	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$	•	•	X	•	X	•	•	•	11	001	001	C9	1	3	10	
RET cc	If condition cc is false continue, otherwise same as RET	•	•	X	•	X	•	•	•	11	cc	000		1 1	1 3	5 11	If cc is false If cc is true
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 01	101 001	101 101	ED 4D	2	4	14	
RETN[1]	Return from non maskable interrupt	•	•	X	•	X	•	•	•	11 01	101 000	101 101	ED 45	2	4	14	
RST p	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L$ $PC_H \leftarrow 0$ $PC_L \leftarrow p$	•	•	X	•	X	•	•	•	11	t	111		1	3	11	

cc	Condition	
000	NZ	non zero
001	Z	zero
010	NC	non carry
011	C	carry
100	PO	parity odd
101	PE	parity even
110	P	sign positive
111	M	sign negative

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

[1] RETN loads $IFF_2 \rightarrow IFF_1$

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
↕ = flag is affected according to the result of the operation.

INPUT AND OUTPUT GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11 011 011 n	DB	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	‡	‡	X	‡	X	P	0	•	11 101 101 01 r 000	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INI	(HL) ← (C) B ← B - 1 HL ← HL + 1	X	‡ ①	X	X	X	X	1	•	11 101 101 10 100 010	ED A2	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INIR	(HL) ← (C) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1 ①	X	X	X	X	1	•	11 101 101 10 110 010	ED B2	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
IND	(HL) ← (C) B ← B - 1 HL ← HL - 1	X	‡ ①	X	X	X	X	1	•	11 101 101 10 101 010	ED AA	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INDR	(HL) ← (C) B ← B - 1 HL ← HL - 1 Repeat until B = 0	X	‡ ①	X	X	X	X	1	•	11 101 101 10 111 010	ED BA	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11 010 011	D3	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11 101 101 01 r 001	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTI	(C) ← (HL) B ← B - 1 HL ← HL + 1	X	‡ ①	X	X	X	X	1	•	11 101 101 10 100 011	ED A3	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTIR	(C) ← (HL) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1 ①	X	X	X	X	1	•	11 101 101 10 110 011	ED B3	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTD	(C) ← (HL) B ← B - 1 HL ← HL - 1	X	‡ ①	X	X	X	X	1	•	11 101 101 10 101 011	ED AB	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTDR	(C) ← (HL) B ← B - 1 HL ← HL - 1 Repeat until B = 0	X	1 ①	X	X	X	X	1	•	11 101 101 10 111 011	ED BB	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$

Notes: ① If the result of B - 1 is zero the Z flag is set, otherwise it is reset.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
‡ = flag is affected according to the result of the operation.

ESPECIFIACIONES ELECTRICAS

11.0 ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... Specified Operating Range

Storage Temperature ..... −65°C to +150°C

Voltage on Any Pin with Respect to Ground ..... −0.3V to +7V

Power Dissipation 1.5W

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ± 5% unless otherwise specified

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITION
V_{ILC}	Clock Input Low Voltage	−0.3		0.8	V	
V_{IHC}	Clock Input High Voltage	Vcc-.6		Vcc+.3	V	
V_{IL}	Input Low Voltage	−0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 1.8mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −250 μA
I_{CC}	Power Supply Current			150*	mA	
I_{LI}	Input Leakage Current			10	μA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			−10	μA	V_{OUT} = 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	$0 \leq V_{IN} \leq V_{CC}$

*200mA for -4, -10 or -20 devices

CAPACITANCE

T_A = 25°C, f = 1MHz unmeasured pins returned to ground

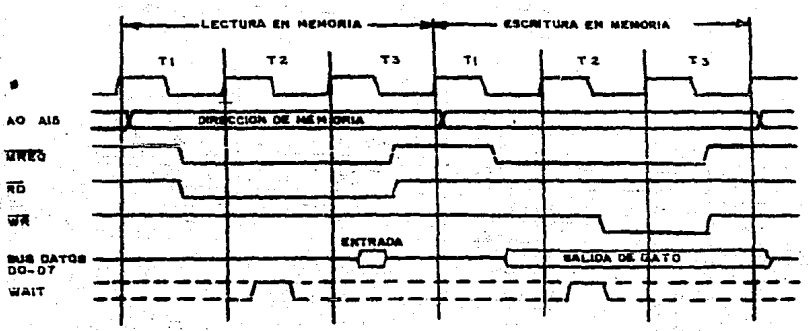
SYMBOL	PARAMETER	MAX.	UNIT
C_Φ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

*Comment

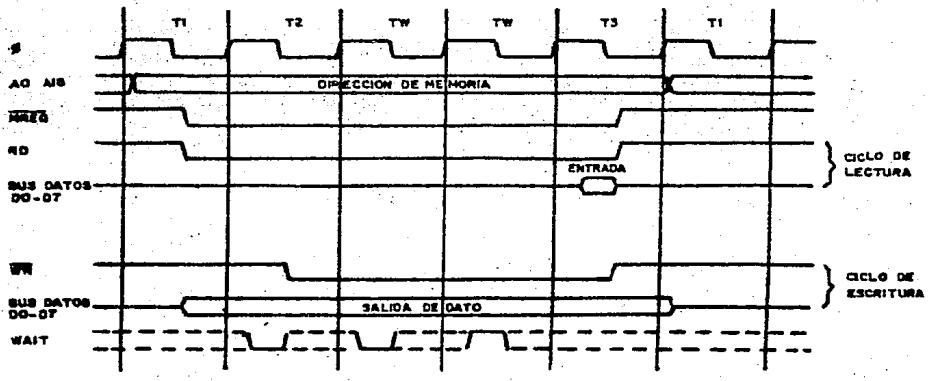
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

APENDICE B

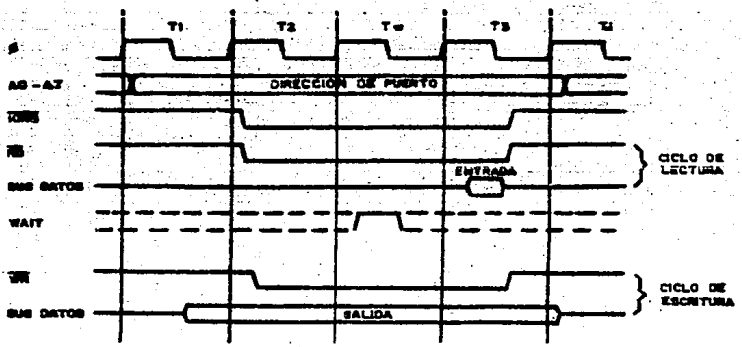
DIAGRAMAS DE TIEMPOS DEL Z-80



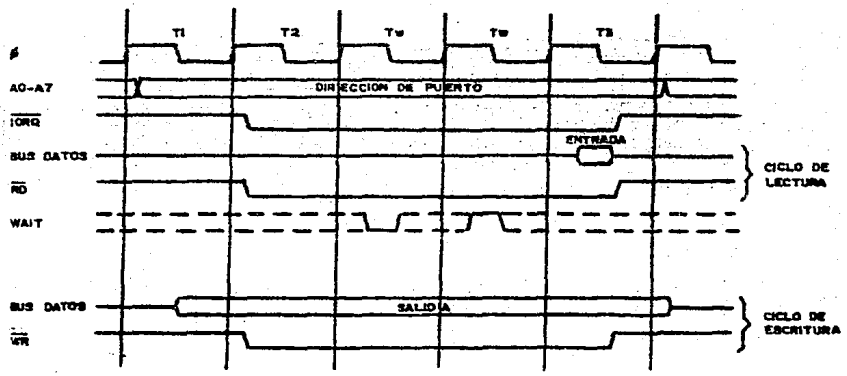
LECTURA O ESCRITURA EN MEMORIA



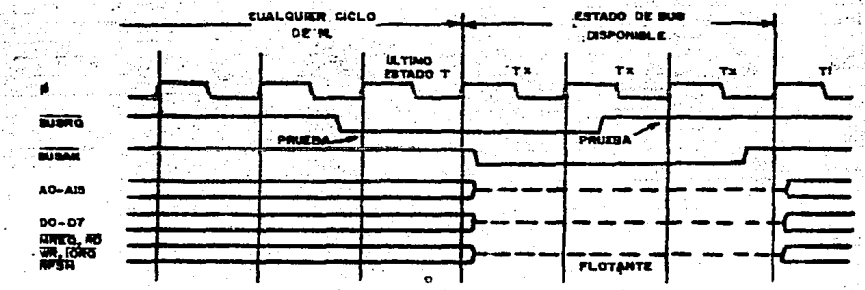
LECTURA O ESCRITURA CON ESTADOS DE ESPERA



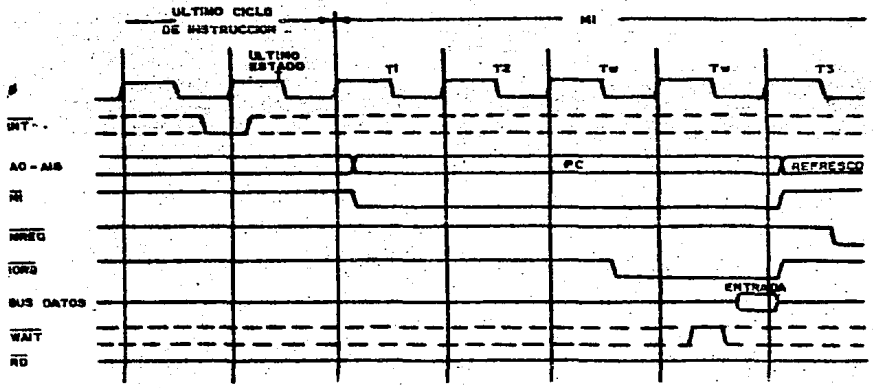
CICLO DE ENTRADA O SALIDA



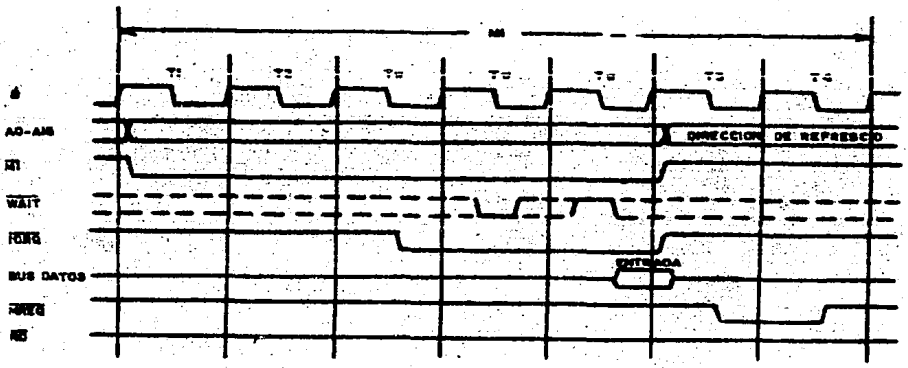
ENTRADA O SALIDA CON ESTADOS DE ESPERA



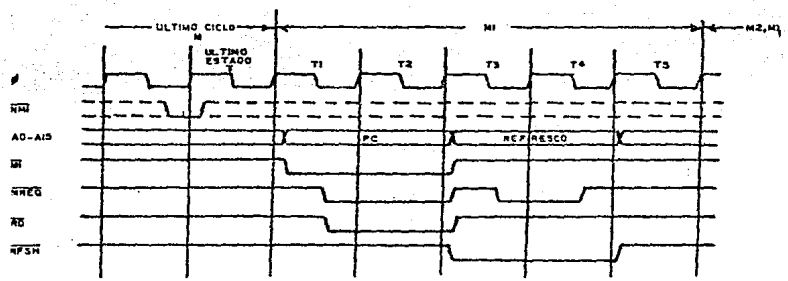
SOLICITUD/RECONOCIMIENTO DE BUS



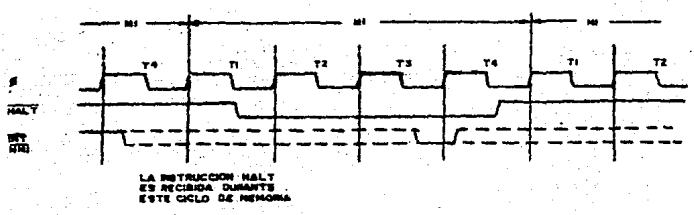
SOLICITUD/RECONOCIMIENTO DE INTERRUPCIONES



SOLICITUD/RECONOCIMIENTO DE INTERRUPCIONES CON ESTADOS DE ESPERA



SOLICITUD DE INTERRUPCION NO MASCARABLES



TIEMPOS DE ESPERA

APENDICE C

BUS NORMALIZADO S-100

nº del terminal	Nombre de la señal (y bus)		Nivel activo/ dirección	Función y características de la señal
1	+8 V	(B)		Mínimo instantáneo (MINI) superior a 7 V. Máximo instantáneo (MAXI) inferior a 25 V. Máximo promedio (MAXP) inferior a 11 V
2	+16 V	(B)		MINI > 14,5 V. MAXI < 35 V (ver nº 1) MAXP < 21,5 V.
3	XRDY	(S)	H/no	Una entrada de «preparado» al amo actual del bus. Activada junto con la nº 72 pasa el bus a «preparado».
4	VI0	(S)	L/si	Señal 0 del vector de interrupciones.
5	VI1	(S)	L/si	Señal 1 del vector de interrupciones.
6	VI2	(S)	L/si	Señal 2 del vector de interrupciones.
7	VI3	(S)	L/si	Señal 3 del vector de interrupciones.
8	VI4	(S)	L/si	Señal 4 del vector de interrupciones.
9	VI5	(S)	L/si	Señal 5 del vector de interrupciones.
10	VI6	(S)	L/si	Señal 6 del vector de interrupciones.
11	VI7	(S)	L/si	Señal 7 del vector de interrupciones.
12	NMI	(S)	L/si	Interrupción no enmascarable.
13	PWRFAIL	(B)	L/no	Señal de fallo de alimentación al bus.
14	QMA3	(M)	L/si	Bit 3 de prioridad para amo temporal del bus (el bit más significativo).
15	A18	(M)	H/no	Bit 18 de dirección ampliada.
16	A16	(M)	H/no	Bit 16 de dirección ampliada.
17	A17	(M)	H/no	Bit 17 de dirección ampliada.
18	SDSB	(M)	L/si	Control para inhabilitar a las 8 señales de estado.
19	CDSB	(M)	L/si	Control para inhabilitar a las 5 señales de salida de controles.
20	GND	(B)		Masa del sistema (común con la nº 100)
21	NDEF			Sin especificar (no definida)
22	ADSB	(M)	L/si	Control para inhabilitar a las 16 señales de dirección.
23	DODSB	(M)	L/si	Control para inhabilitar a las 8 señales de salida de datos.
24	Φ	(B)	H/no	Señal de origen de temporización para el bus.
25	sSTVAL	(M)	L/no	Validación («strobe») de estado.
26	sHLDA	(M)	H/no	Control que junto con HOLD (nº 74) coordina las operaciones de transferencia de los amos de bus.
27	RFU			Sin especificar (reservada para usos posteriores).
28	RFU			Sin especificar (reservada para usos posteriores)
29	A5	(M)	H/no	Bit 5 de dirección.
30	A4	(M)	H/no	Bit 4 de dirección.
31	A3	(M)	H/no	Bit 3 de dirección.
32	A15	(M)	H/no	Bit 15 de dirección.
33	A12	(M)	H/no	Bit 12 de dirección.
34	A9	(M)	H/no	Bit 9 de dirección.
35	DO1 DATA1	(M) (M-S)	H/no	Bit 1 de dato de salida/Bit 1 de dato bidireccional.
36	DO0 DATA0	(M) (M-S)	H/no	Bit 0 de dato de salida/Bit 0 de dato bidireccional
37	A10	(M)	H/no	Bit 10 de dirección.
38	DO4 DATA4	(M) (M-S)	H/no	Bit 4 de dato de salida/Bit 4 de dato bidireccional.
39	DO5 DATA5	(M) (M-S)	H/no	Bit 5 de dato de salida/Bit 5 de dato bidireccional.
40	DO6 DATA6	(M) (M-S)	H/no	Bit 6 de dato de salida/Bit 6 de dato bidireccional.
41	DI2 DATA10	(S) (M-S)	H/no	Bit 2 de dato de entrada/Bit 10 de dato bidireccional.
42	DI3 DATA11	(S) (M-S)	H/no	Bit 3 de dato de entrada/Bit 11 de dato bidireccional.
43	DI7 DATA15	(S) (M-S)	H/no	Bit 7 de dato de entrada/Bit 15 de dato bidireccional.
44	sM1	(M)	H/no	Indica que el ciclo en curso es de extracción del código de operación.
45	sOUT	(M)	H/no	Identifica el ciclo de transferencia de datos hacia un dispositivo de salida.
46	sINP	(M)	H/no	Identifica el ciclo de transferencia de datos desde un dispositivo de entrada.
47	sMEMR	(M)	H/no	Identifica los ciclos de bus en que se transfieren datos desde memoria hacia un amo de bus y que no sean ciclos de extracción de instrucción de reconocimiento de interrupción.
48	sHLTA	(M)	H/no	Señal de reconocimiento de la ejecución de HLT.
49	CLOCK	(B)		De 2 MHz (± 10 kHz) y ciclo de trabajo del 40 al 60%. No es preciso que se sincronice con ninguna otra señal del bus.
50	GND	(B)		Masa del sistema (común con la nº 100)
51	+8 V	(B)		(ver nº 1)
52	−16 V	(B)		MAXI<-14,5 V; MINI>-35 V MINP>21,5 V. (ver nº 1)
53	GND	(B)		Masa del sistema (común con la nº 100).
54	SLAVE CLR	(S)	L/si	Señal de reinicialización para los esclavos del bus. Puede activarse externamente.
55	DMA0	(M)	L/si	Bit 0 de prioridad para amo temporal del bus.
56	DMA1	(M)	L/si	Bit 1 de prioridad para amo temporal del bus.

Nº del terminal	nombre de la señal (y tipo)		[illegible]	*Función y características de la señal*
57	DMA2	(M)	L/si	Bit 2 de prioridad para amo temporal bus.
58	sXRQ	(M)	L/no	Solicitud de transferencia sobre 16 bits (ver n.º 60).
59	A19	(M)	H/no	Bit 19 de dirección ampliada.
60	SIXTN	(S)	L/si	Respuesta afirmativa de los esclavos del bus a la señal de petición sXTRQ (n.º 58).
61	A20	(M)	H/no	Bit 20 de dirección ampliada.
62	A21	(M)	H/no	Bit 21 de dirección ampliada
63	A22	(M)	H/no	Bit 22 de dirección ampliada.
64	A23	(M)	H/no	Bit 23 de dirección ampliada.
65	NDEF			Sin especificar (no definida).
66	NDEF			Sin especificar (no definida).
67	PHANTOM	(M-S)	L/si	Habilitar a los dispositivos «esclavos fantasmas» (útil en principio para sistemas «bootstrapping» sin circuitos de panel frontal)
68	MWRT	(B)	H/no	Escritura en memoria: debe seguir a la pWR con un máximo de 30 ns. de retraso (ver n.º 77).
69	RFU			Sin especificar (reservada para usos posteriores)
70	GND	(B)		Masa del sistema (común con la n.º 100)
71	RFU			Sin especificar (reservada para usos posteriores).
72	RDY	(S)	H/si	Ver señal n.º 3.
73	INT	(S)	L/si	Señal principal de petición de interrupción
74	HOLD	(M)	L/si	Control para coordinar operaciones de transferencia de amo del bus. (ver n.º 26).
75	RESET	(B)	L/si	Reinicialización de los amos del bus. Puede generarse externamente. Precisa la activación de la POC (n.º 99)
76	pSYNC	(M)	H/no	Control que identifica el estado inicial, BS_1, de un ciclo de bus.
77	pWR	(M)	L/no	Control que indica la presencia de datos válidos en el bus de datos de salida
78	pDBIN	(M)	H/no	Control que solicita datos al bus de entrada de datos
79	A0	(M)	H/no	Bit 0 de dirección (el menos significativo).
80	A1	(M)	H/no	Bit 1 de dirección.
81	A2	(M)	H/no	Bit 2 de dirección.
82	A6	(M)	H/no	Bit 6 de dirección.
83	A7	(M)	H/no	Bit 7 de dirección.
84	A8	(M)	H/no	Bit 8 de dirección.
85	A13	(M)	H/no	Bit 13 de dirección.
86	A14	(M)	H/no	Bit 14 de dirección.
87	A11	(M)	H/no	Bit 11 de dirección.
88	DO2 DATA2	(M) (M-S)	H/no	Bit 2 de datos de salida/Bit 2 de dato bidireccional
89	DO3 DATA3	(M) (M-S)	H/no	Bit 3 de dato de salida/Bit 3 de dato bidireccional
90	DO7 DATA7	(M) (M-S)	H/no	Bit 7 de dato de salida/Bit 7 de dato bidireccional
91	DI4 DATA12	(S) (M-S)	H/no	Bit 4 de dato de entrada/Bit 12 de dato bidireccional
92	DI5 DATA13	(S) (M-S)	H/no	Bit 5 de datos de entrada/Bit 13 de dato bidireccional
93	DI6 DATA14	(S) (M-S)	H/no	Bit 6 de dato de entrada/Bit 14 de dato bidireccional
94	DI1 DATA9	(S) (M-S)	H/no	Bit 1 de dato de entrada/Bit 9 de dato bidireccional
95	DI0 DATA8	(S) (M-S)	H/no	Bit 0 (el menos significativo) de dato de entrada/Bit 8 de dato bidireccional
96	sINTA	(M)	H/no	Identifica el ciclo (s) de entrada que puede(n) seguir a la aceptación de una petición de interrupción
97	sWO	(M)	L/no	Identifica un ciclo de bus en el que se transfieren datos desde un amo de bus hacia un esclavo
98	ERROR	(S)	L/si	Significa que hay alguna condición de error durante el ciclo de bus en curso
99	POC	(B)	L/no	Señal de puesta a cero al conectar alimentación para todos los dispositivos, cuando pasa al nivel bajo debe permanecer así al menos 10 ms
100	GND	(B)		Masa del sistema

NOTAS

- H significa señal en estado alto
- L significa señal en estado bajo
- M significa señal que debe generar el amo del bus cuando tiene el control del bus
- S significa señal que debe generar el esclavo del bus (las necesarias para la comunicación en curso)
- B las demás señales (algunas de ellas puede generarlas un amo)

Por tratarse de especificaciones no definitivas, estas señales podrían sufrir alguna pequeña modificación en el futuro

APENDICE D

EL PROCESADOR DE COMUNICACIONES DELANTERO

CARACTERISTICAS Y FUNCIONES DEL PROCESADOR DE COMUNICACIONES DELANTERO

1.- Conecta desde una a varios cientos de líneas de comunicaciones a la computadora principal. De esta manera es la interface entre las líneas de comunicaciones y la computadora. Para redes muy grandes puede haber varios procesadores de comunicaciones delanteros conectados a una misma computadora.

2.- Acepta la transmisión síncrona, asíncrona ó isócrona y en serie ó paralelo para los datos que vienen desde terminales remotas. Esto significa que el procesador delantero debe convertir cualquier forma de transmisión recibida a la que acepte la computadora central, según su diseño arquitectónico. La computadora central ó el procesador delantero puede hacer la conversión del formato de caracteres en serie a paralelo. Dicha conversión se conoce como serialización ó deserialización de los bits que representan cada caracter. La mayoría de las computadoras modernas son

máquinas orientadas a palabras ó a caracteres, es decir que dentro de la computadora se transmite un caracter completo (6 a 9 bits) ó una palabra de 2 a 4 caracteres en cada ciclo de la computadora. En la comunicación de datos no se transmite todo un caracter de inmediato por la línea (excepto en la transmisión en paralelo, de uso ocacional), sino que el procesador delantero serializa los caracteres que ha recibido de la computadora anfitriona.

3.- Sondea las terminales para ver si tienen que enviar un mensaje ó si esta en estado de recibir un mensaje, el sondeo lo realiza el dispositivo delantero para evitar a la computadora central la pérdida de tiempo que implica dicho sondeo. También puede darse instrucciones al procesador para que cambie la secuencia de sondeo siempre que haya picos de tráfico durante el día laboral.

4.- Maneja las respuestas ó llamadas hacia el exterior en forma automática en la red telefónica pública para conectar diversas terminales al sistema.

5.- Utiliza diversas claves de comunicaciones de datos cuando la red tiene diversos tipos de terminales conectadas. El procesador delantero puede realizar la traducción de códigos requerida, aunque para tener eficiencia probablemente debiera hacerse esto por medio de un

dispositivo específico para reducir el alto del procesador delantero.

6.- Realiza la conmutación de circuito ó líneas y da la posibilidad de almacenar y retransmitir. Siempre que una terminal está transmitiendo un mensaje a otra terminal, el procesador delantero puede almacenar ese mensaje al retransmitirlo luego a la segunda terminal cuando no este ocupada. De la misma manera se retienen los mensajes a una terminal descompuesta hasta que vuelva a la línea.

7.- Maneja las diferencias de velocidad de transmisión. Las terminales conectadas al sistema no necesitan transmitir a la computadora central a la misma velocidad.

8.- Lleva la bitácora de todos los mensajes hacia y desde el exterior de una bitácora especial, lo que proporciona la protección y la contabilidad de datos para el caso de la pérdida de un mensaje, junto con la capacidad de restaurar el sistema si ocurre un "desplome".

9.- Proporciona la detección y la corrección de errores. Cuando se encuentra un error en un mensaje el procesador delantero corrije u ordena la retransmisión del mensaje desde la terminal emisora. También puede interpretar

los mensajes con errores lógicos como son: dirección de terminales no existentes, información de encabezados incorrecta, etc. Cuando estos errores son detectados se avisa a la estación emisora o se rutea el mensaje a una terminal de corrección central.

10.- Agrega claves de control de linea de comunicaciones a los mensajes de salida y las quita de los mensajes de entrada Ejemplos de estas claves son el caracter de fin de bloque, caracter de fin de transmisión, caracter de principio de mensaje y otros.

11.- Puede direccionar un grupo especial de terminales (dirección de grupo), varias terminales a la vez (direcciones múltiples), una sola terminal (dirección simple) ó envia un mensaje simultáneamente a todas las terminales en el sistema.

12.- Asigna número de serie, asi como registro de hora y fecha a todos los mensajes que maneja.

13.- Maneja el tiempo de llamada que corresponde a cada terminal. Si una terminal ó estación no responde con rapidez, el sistema agota el tiempo, y se salta a esa estación pasando a la siguiente estación ó actividad a realizar para que se pueda dar servicio a otros usuarios aún

cuando no esté respondiendo correctamente.

14.- Realiza dos niveles de edición. En el primero el procesador delantero puede agregar partes a un mensaje, rerutearlo o arreglar los datos para continuar la transmisión. También puede determinar si la direccion del mensaje es exacta, además de realizar las pruebas de paridad, en el segundo nivel de edición, el procesador delantero se programa para realizar ediciones específicas de los distintos mensajes que entran al sistema. Esta edición se refiere al contenido del mensaje en lugar de la forma y es específica a los programas de aplicación que se ejecuten.

15.- Maneja el sistema de prioridad de mensaje, si existe. Se establece un plan de prioridades para que determinadas áreas de la red puedan utilizar más la linea ó para asegurar que se maneje determinadas operaciones antes que otras, de menor importancia.

16.- Realiza la correlación de la densidad de tráfico y disponibilidad de circuitos. Estos análisis son obligatorios para administrar de manera eficiente una red grande de comunicación de datos. Algunos de los articulos incluídos en un informe de densidad de tráfico son (1) el número de mensajes manejados por hora o día en cada eslabón de la red, (2) el número de errores que se encuentran por

hora ó día, (3) el número de errores encontrados por programa ó módulo de programa, (4) las terminales ó estaciones que parecen tener mayor índice de errores que el promedio.

17.- Determinar los caminos alternos sobre los que se transmitirán los datos, cuando la red tiene multiples caminos que pueden escojerse para transmitir un mensaje de una estación a otra. Se puede escoger un camino alterno si el procesador delantero detecta un índice excesivo de errores de línea ó tráfico intenso en un eslabón.

18.- Realiza varias funciones, como: (1) enciende alarmas remotas si se exceden determinados parámetros; (2) ejecutan las funciones de multiplicación involucradas cuando se emplean circuitos en formato multiplicado (lo que por lo general se efectúa mediante dispositivos específicos de multiplicación, ó multiplex; (3) avisa de las anomalías a la computadora central y (4) frena la entrada y salida de mensajes cuando la computadora central está sobrecargada y hay la posibilidad de que se desplome el sistema debido al intenso tráfico.

APENDICE E

SUMARIO DE LAS CARACTERISTICAS DE PROGRAMACION PARA LA TRANSMISION DE DATOS

La lógica para llevar a cabo las funciones que se mencionan a continuación puede estar en:

1.- Equipo
2.- Microprogramación de "empresa"
3.- Programación de paquete
4.- Programas escritos para un usuario

A. Requerimientos lógicos para las funciones básicas de transmisión:

En la recepción:

1.- Iniciar y controlar la recepción de datos.
2.- Ensamblar los bits para formar caracteres.
3.- Ensamblar los caracteres para formar mensajes.
4.- Reconocer los caracteres de fin de registro y de fin de transmisión.
5.- Convertir la puesta en clave de los caracteres.
6.- Comprobar errores.
7.- Corregir errores ó iniciar la retransmisión

de mensajes erróneos.

8.- Redactar los mensajes que llegan removiendo los caracteres de retroceso y otros de esa misma índole.

9.- Entregar los mensajes al programa principal.

Durante la transmisión:

10.- Acepta mensajes de los programas principales.

11.- Prepararlos para su transmisión, añadiendoles los caracteres apropiados de control y dirección.

12.- Iniciar la transmisión.

13.- Terminar la transmisión, cuando se envia el caracter de fin de transmisión.

14.- Vigilar el proceso de transmisión.

15.- Aceptar el acuse de recibo de la terminal receptora.

16.- Retransmitir los mensajes que se recibieron con errores ó que no se recibieron.

B. Para dirigir la transmisión del mecanismo pertinente:

Para sistemas de disco:

17.- Llamar con el disco a un mecanismo remoto.

18.- Rastreo de terminales de comunicación con disco.

19.- Contestación automática cuando se llama con el disco a la computadora.

20.- Iniciar la transmisión después de una conexión con disco.

21.- Determinar el tipo de sistema que ha llamado con el disco.

Para sistemas de matrícula:

22.- Matriculación de las terminales para averiguar cuando alguna de ellas tiene algo que transmitir.

23.- Recibir la contestación de las terminales e iniciar la recepción, cuando así se especifique.

24.- Enviar un mensaje a una terminal, pidiendo permiso para transmitir.

25.- Recibir la contestación de la terminal e iniciar la transmisión.

26.- Reajustar la línea después de la transmisión.

27.- Función de espera si un mecanismo no contesta.

28.- Modificar un mecanismo de matrícula cuando falla o se quita un mecanismo.

29.- Cambiar la dirección de envío de un mecanismo cuando se usa matriculacón central, caso dado si falla la terminal de esa dirección.

30.- Recuperación de direcciones de errores.

Para sistemas de lazada:

31.- Establecer y mantener la sincronización de lazada.

32.- Colocar los caracteres en ranuras apropiadas de transmisión.

33.- Recibir los caracteres de las ranuras apropiadas de transmisión.

34.- Descubrir errores en la recepción y pedi la retransmisión al mecanismo.

35.- Aceptar notificaciones de errores de salida y retransmitir.

C.- Bloqueo, funcionamiento múltiplex y compresión:

36.- Cambiar en un solo mensaje "físico" de transmición, varios mensajes lógicos.

37.- Extraer los mensajes lógicos durante la recepción.

38.- Cuando se usa un multiplexor remoto la programación puede llevar a cabo las operaciones de multiplexar y desmultiplexar en el centro de computadoras.

39.- Los datos transmitidos pueden comprimirse con varias técnicas posibles para ensancharse en otro mecanismo remoto.

40.- pueden enviarse referencias a mensajes

compuestos previamente y esos mensajes pueden generarse en un mecanismo remoto.

41.- El ensanchamiento puede llevarse a cabo en la entrada.

D. Planeamiento y asignación de recursos:

42.- Asignación de amortiguadores. Se emplea la asignación dinámica de los mismos para utilizar más eficientemente el almacenamiento.

43.- Manejo de filas de esperas de lineas.

44.- Manejo de las filas de espera para los programas.

45.- Manejo de partidas para destinos multiples.

46.- Funcionamiento de almacenamiento y envío.

47.- Interconexión de mensajes

E. Registro y obtención de estadisticas:

48.- Registro de mensajes (para referencia posterior y para procedimientos de recuperación).

49.- Mantenimiento de estadisticas sobre volumenes de tráfico.

50.- Mantenimiento de estadisticas sobre fallas de línea.

F. Control de una unidad externa de contestación:

52.- Control de una unidad externa que genera respuestas.

53.- Control de una unidad externa que da formatos a documentos y exhibiciones.

54.- Control de una unidad externa de contestación de voz.

G. Manejo de fallas:

55.- Generación de números de secuencia de mensajes para usarlos en procedimientos de recuperación de mensajes.

56.- Rutinas de recuperación asociadas con el registro de mensajes.

57.- Rutinas de recuperación asociadas con numeración en serie.

58.- Iniciación de la comunicación con disco, cuando falla una línea arrendada.

59.- Obtención de totales mezclados de las transacciones que llegan.

60.- Contestación a las terminales cuando hay una falla parcial del sistema central.

H. Funciones de seguridad:

61.- Puesta en clave y conversión de mensajes.

62.- Identificación de las terminales que

envía entradas.

63.- Identificación de las terminales antes de enviar salidas.

64.- Identificación del usuario de la terminal, por ejemplo, con claves de seguridad.

I. Diagnósticos:

65.- Diagnósticos para verificar el funcionamiento correcto de una línea y una terminal, desde otra terminal remota.

66.- Pruebas cruzadas para cerciorarse de que el equipo y la programación de la computadora de control de la línea estan funcionando correctamente.

67.- Programación para enviar y recibir automáticamente una respuesta de diagnóstico de un mecanismo remoto.

68.- Determinación de los mecanismos que no funcionan en líneas con mecanismos múltiples.

69.- Mantenimiento de las tablas de situacio de las cadenas y mecanismos.

J. Funciones de diálogo. (En general, orientadas hacia las aplicaciones):

70.- Verificaciones de la racionalidad de la entrada.

71.- Verificaciones de totalidad de entrada

72.- Redacción del contesto de la entrada.

73.- Vuelta de páginas, enrollado, búsqueda de índices, hojeadas u otras operaciones para revisar datos sin procesarlos.

74.- Diálogo con los operadores antes de emplear programas de aplicación (por ejemplo, para recolección de datos).

APENDICE F

PROGRAMA

PROBLEMA:

Conexión entre el sistema cromemco y el Kit MKE-Z80.

El Kit MKE-Z80 debe ser capaz de leer programas ensamblados en hexadecimal grabados en disco desde el Drive "B".

El sistema Cromemco debe de hacer las veces de una interfase entre el Drive "B" y el Kit MKE-Z80, el cual debe de responder a las ordenes que se le hagan desde el Kit MKE-Z80.

Es responsabilidad del usuario que utilice este programa que el archivo que se quiera leer este en el disco ensamblado en hexadecimal y que este en el Drive "B", ya que si esto no se cumple este programa no funciona correctamente.

Cuando en el Drive "B" no haya disco y se pidio lectura de un archivo el Kit y el sistema esperará hasta que se le introduzca el disco en el Drive "B" pero no se

informará que esto se tiene que hacer.

La información que se lea de algún archivo debe de estar comprendida entre la dirección 2000H y 2A00H ya que si no es esta, en el kit la información se grabará en el area de variables o será una dirección de ROM, aún si es muy cerca de la dirección 2A00H, el programa podría fallar en caso de que se ocupara mucho el STACK ya que el STACK del Kit MKE-Z80 empieza en la dirección 2BA4H cuando esta corriendo el monitor HOLA, y cuando se corre este programa el STACK comienza en la dirección 2AE0H.

LISTADO DE PROGRAMAS

PROGRAMA RESIDENTE EN DISCO

```
TYPE TESIS.Z80
;*****************************************************************************
;*                                                                           *
;*                    PROGRAMA PRINCIPAL                                     *
;*                                                                           *
;*****************************************************************************

        INICIO. CALL CINI          ;LLAMA A CINI
                LD B,4             ;# CARACTERES = 4
                LD HL,NOMC         ;DIR TABLA = NOMC
                CALL INCRO         ;LLAMA A INCRO
                CALL SDB           ;LLAMA A SDB
                LD DE,DBUFER
                CALL BSET          ;LLAMA A BSET
                LD DE,FCB
                CALL ABRE          ;LLAMA A ABRE
        PRORE.  LD DE,FCB
                CALL LEE           ;LLAMA A LEE
                CALL TRAF          ;LLAMA A TRAF
                CP 00H
                JR Z,PRORE
                LD DE,FCB
                CALL CIERRA        ;LLAMA A CIERRA
                CALL CONV          ;LLAMA A CONV
                JP INICIO

;*****************************************************************************
;*                                                                           *
;*                FIN DE PROGRAMA PRINCIPAL                                  *
;*                                                                           *
;*****************************************************************************
```

```
;*****************************************************************
;*                                                               *
;*                    ESPACIO    DE    VARIABLES                 *
;*                                                               *
;*****************************************************************

        FCB.                        ;APUNTADOR DE FCB
        FCBDK.  DB DDB              ;DISCO EN OPERACION
        NOMC.   DB '
        FCBFT.  DB 'HEX'            ;EXTENCION DE ARCHIVO
        FCBEX.  DB 0                ;NUMERO DE SECTORES
        RESER.  DS 2                ;RESERVA
        FCBRC.  DS 1                ;CONTADOR DE RECORD
        FCBMP.  DS 16               ;MAPA DE LOCALIZACION
        FCBNR.  DS 1                ;PROXIMO RECORD
        DBUFER. DS 80H              ;BUFFER DEL DISCO
        OK.     DS 01H

;*****************************************************************
;*                                                               *
;*                         SUBRUTINAS                            *
;*                                                               *
;*****************************************************************



;*****************************************************************
;*                                                               *
;*      CINI.INICIALIZAR EL SISTEMA                              *
;*           ENTRADA.NINGUNA                                     *
;*           SALIDA.STB ALTO, ACK ALTO                           *
;*           MODIFICA.A                                          *
;*           LLAMA.NINGUNO                                       *
;*                                                               *
;*****************************************************************

        CINI.   LD A,STB1           ;STB - 1 (ALTO)
                OUT (SSTB),A        ;ESCRIBE STB EN EL 8255
                LD A,ACK1           ;ACK - 1 (ALTO)
                OUT (SACK),A        ;ESCRIBE ACK EN EL 8255
                LD HL,FTPR
                LD(SIGT),HL
                LD(INT),HL
                LD(INO),HL
                LD HL,FTB
                LD(FTB1),HL
                RET
```

```
;*****************************************************************
;                                                                 *
;       INCRO.RECIBE DEL KIT UNA CANTIDAD DE DATOS, ESTA          *
;            CANTIDAD DE DATOS DEBE DE ESTAR EN EL                *
;            REGISTRO B, Y LOS DEPOSITA EN UN BUFFER QUE          *
;            DEBE DE ESTAR APUNTADO POR HL                        *
;            ENTRADA.B, HL                                        *
;            SALIDA.PUERTO DE ENTRADA DEL SISTEMA                 *
;            LLAMA.NINGUNA                                        *
;                                                                 *
;*****************************************************************

INCRO.  LD C,PSC        ;PUERTO DE ENTRADA=PSC
                        ;PARA CONT=1 HASTA B EN PASO 1 HAGA
NTDAT.  IN A,(SOBF)             ;MIENTRAS OBF <> ALTO HAGA
        BIT 1,A                 ;SENSA OBF
        JR NZ,NTDAT             ;FINMIENTRAS
        INI                     ;RECIBE DATOS (CONT)
        RES 0,A                 ;SACA ACK BAJO
        OUT (SACK),A
        SET 0,A                 ;SACA ACK ALTO
        OUT (SACK),A
        JR NZ,NTDAT     ;FINPARA
        RET             ;REGRESA


;*****************************************************************
;                                                                 *
;               SUBRUTINAS DEL SISTEMA                            *
;                                                                 *
;*****************************************************************

SDB.    LD E,DDB        SELECCIONA EL DRIVE B
        LD C,SDD
        CALL CDOS
        RET
BSET.   LD C,PON        ;PONE EL BUFFER DEL DISCO
        CALL CDOS
        RET
ABRE.   LD C,ABRIR      ;ABRE ARCHIVO DE DISCO
        CALL CDOS
        RET
LEE.    LD C,LEERD      ;LEE PROXIMO RECORD
        CALL CDOS
        RET
CIERRA. LD C,CERRAR     ;CIERRA ARCHIVO DE DISCO
        CALL CDOS
        RET

;*****************************************************************
```

```
;++++++++++++
;+                                                                  +
;+      OUTKI:TRANSMITE A EL KIT UNA CANTIDAD DE DATOS,             +
;+             ESTA CANTIDAD DEBE DE ESTAR EN EL REGISTRO           +
;+             B, Y EL BUFFER DE SALIDA DEBE DE ESTAR               +
;+             APUNTADO POR EL REGISTRO HL                          +
;+             ENTRADA:B, HL                                        +
;+             SALIDA:PUERTO DE SALIDA DE CROMEMCO                  +
;+             MODIFICA:A, B, C, HL                                 +
;+             LLAMA:NINGUNA                                        +
;+                                                                  +
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

OUTKI:  LD C,0AH          ;C - PUERTO SALIDA CARACTERES
NUDAT:  LD A,00000001B    ;STB - 1 (ALTO)
        OUT (0BH),A       ;ESCRIBE STB EN EL 8255
IBF:    IN A,(0AH)        ;LEE IBF DEL 8255
        AND 00000001B     ;SI IBF - 0
        JR NZ,IBF         ;A.-ENTONCES VE A IBF
                          ;B.-DE LO CONTRARIO CONTINUA
        OUTI              ;ESCRIBE (DIR TABLA) EN 8255
                          ;# CARACTERES - # CARACTERES - 1
                          ;DIR TABLA - DIR TABLA + 1
        LD A,00000000B    ;STB   0 (BAJO)
        OUT(0BH),A        ;ESCRIBE STB EN EL 8255
        JR N[illegible]   ;SI # CARACTERES   0
                          ;A.-ENTONCES CONTINUA
                          ;B.-DE LO CONTRARIO VE A NUDAT
        LD A,00000001B    ;STB - 1 (ALTO)
        OUT(0BH),A        ;ESCRIBE EN EL 8255
        RET

;-++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;+                                                                    +
;+      CAH:CONVERTIR UN BYTE DE CODIGO ASCII A HEXADECIMAL,          +
;+             ESTE BYTE DEBE DE ESTAR EN EL REGISTRO A Y LA          +
;+             SALIDA ESTA TAMBIEN EN EL REGISTRO A                   +
;+             ENTRADA:A                                              +
;+             SALIDA:A                                               +
;+             MODIFICA:A, D                                          +
;+             LLAMA:NINGUNA                                          +
;+                                                                    +
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


CAH:    LD D,30H          ;A- HEXA(A)
        LD A,(HL)
        SUB D
        CP 10H
        JR C, SALTE
        LD D,7
        SUB D
SALTE:  RET               ;REGRESA
```

```
;*****************************************************************
;*                                                               *
;*      TRAF TRANSFIERE EL BUFFER DEL DRIVE, A EL BUFFER DEL     *
;*           ARCHIVO                                             *
;*           ENTRADA.NINGUNA                                     *
;*           SALIDA.BUFFER DE ARCHIVO                            *
;*           MODIFICA, HL, DE, <SIGT>, BC,                       *
;*           LLAMA.NINGUNA                                       *
;*                                                               *
;*****************************************************************

TRAF    LD HL,DBUFER        ;APUNTA BUFFER DRIVE
        LD DE,(SIGT)        ;APUNTA BUFFER ARCHIVO
        LD BC,80H
        LDIR                ;ARCHIVO - BUFFER DRIVE
        LD(SIGT),DE         ;DIRECCION NUEVA DE BUFFER ARCHIVO
        RET                 ;REGRESA

;*****************************************************************
;*                                                               *
;*      CONV CONVIERTE EL ARCHIVO QUE ESTA EN HEXADECIMAL EN     *
;*           BINARIO Y LO TRANSMITE AL KIT                       *
;*           ENTRADA NINGUNA                                     *
;*           SALIDA.ARCHIVO HACIA EL KIT                         *
;*           MODIFICA.HL, BC, A, DE                              *
;*           LLAMA  OUTKI                                        *
;*                                                               *
;*****************************************************************

CONV    LD HL,(INC)         ;APUNTA INICIO DE ARCHIVO
OTS     INC HL              ;DESHECHA UN CARACTER
        CALL CO1            ;CONVIERTE CARACTER
        LD (CYO),A          ;CYO-CANTIDAD
        CALL CO1            ;CONVIERTE CARACTER
        LD (CYO+2),A        ;CYO+2-PRIMER BYTE DE ORIGEN

        CALL CO1            ;CONVIERTE DIGITO
        LD (CYO+1),A        ;CYO+1-SEGUNDO BYTE DE ORIGEN
        INC HL              ;DESHECHA 2 ARCHIVOS
        INC HL
        LD A,(CYO)          ;SI CANTIDAD DIFERENTE DE 0 ENTONCES
        LD B,A
        CP 00
        JR Z,FIN
        LD A,(CYO)          ;B-CANTIDAD
        LD B,A
OT      CALL CO1                    ;REPITE HASTA QUE B=0
        PUSH HL                     ;CONVIERTE UN DATO A TRANSFERIR
        LD HL,(FTB1)
        LD (HL),A
        INC HL                      ;APUNTA SIGUIENTE DATO
        LD (FTB1),HL
        LD HL,(INT)
        INC HL
        LD(INT),HL
        POP HL
        DEC B                       ;B=B-1
        JR NZ,OT                    ;FINREPITE
```

```
        CALL FIN          ;TRANSFIERE CANTIDAD Y ORIGEN
        LD A,(CYO)        ;CYO-CANTIDAD
        LD B,A
        PUSH HL
        LD HL,FTB         ;FTB-ORIGEN
        LD(FTB1),HL
        CALL OUTKI        ;TRANSFIERE DATOS
        POP HL            ;DESHECHA CARACTERES
        INC HL
        INC HL
        INC HL
        INC HL
        JP GTS            ;FINSI
;*****************************************************************
;*                                                               *
;*      FIN.TRANSFIERE CANTIDAD Y ORIGEN DE UN BLOQUE DE DATOS   *
;*              ENTRADA. CYO                                     *
;*              SALIDA CANTIDAD Y ORIGEN AL KIT                  *
;*              MODIFICA.B, BC, DE, A                            *
;*              LLAMA OUTKI                                      *
;*                                                               *
;*****************************************************************

FIN.    PUSH HL
        LD B,03           ;CANTIDAD 3
        LD HL,CYO         ;CYO-ORIGEN
        CALL OUTKI        ;TRANSFIERE DATOS
        POP HL
        RET               ;REGRESA

;*****************************************************************
;*                                                               *
;*      C21 CONVIERTE DOS LOCALIDADES CONTIGUAS DE CODIGO        *
;*              ASCII A UN DATO HEXADECIMAL APUNTADO POR EL      *
;*              REGISTRO HL, Y LO DEPOSITA EN EL REGISTRO A      *
;*              ENTRADA.HL                                       *
;*              SALIDA.A                                         *
;*              MODIFICA.A, E, HL, (PE)                          *
;*              LLAMA CAH                                        *
;*                                                               *
;*****************************************************************

C21     CALL CAH          ;A-HEXA(<HL>)
        LD E,A            ;E-A
        INC HL
        CALL CAH          ;A-HEXA(<HL>)
        INC HL
        PUSH HL
        LD HL,PE          ;A-"E#" + "A#"
        LD (PE),A
        LD A,E
        RLCA
        RLCA
        RLCA
        RLCA
        RRD
        POP HL
        RET               ;REGRESA
```

```
;*****************************************************************
;*                                                               *
;*                  ASIGNACION DE VARIABLES                      *
;*                                                               *
;*****************************************************************

        STB1.    EQU 0000000001B  ;STB = 1
        SSTB.    EQU 0BH
        ACK1.    EQU 0000000001B  ;ACK = 1
        SACK.    EQU 04H
        PSC.     EQU 03H
        SOBF.    EQU 0AH
        CDOS.    EQU 05H

;*****************************************************************
;*                                                               *
;*                  SUBRUTINAS DEL SISTEMA                       *
;*                                                               *
;*****************************************************************

        SDD.     EQU 0EH
        DDB.     EQU 02H
        PON.     EQU 13H
        ABRIR.   EQU 0FH
        LEERD.   EQU 11H
        CERRAR.  EQU 10H
        IND.     DS 2
        CYO.     DS 2
        INT.     DS 2
        SIGT.    DS 2
        PE.      DS 1
        FTB1.    DS 2
        FTB.     DS 1CH
        FTPR.    DS 2000H

        END      INICIO
```

PROGRAMA RESIDENTE EN KIT MKE-Z80

```
A>TYPE PRUE2.Z80
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;+                                                                     +
;+                    PROGRAMA PRINCIPAL                               +
;+                                                                     +
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

DESP2.   EQU     0D82H   ;DESPLEGA EL MENSAJE DE 8
                         ;DIGITOS APUNTADOS POR HL
TADIR    EQU     07D8H   ;DIRECCION DE LA TABLA DIR
BUFF9.   EQU     2BD1H   ;BUFFER # 9
BFF90.   EQU     0700H   ;COLOCA LOS BUFFER 9 Y 8 EN
                         ;CERO
ALD82    EQU     066AH   ;TOMA EL ULTIMO DATO TECLEADO,
                         ;LO COLOCA EN C, LO CONVIERTE
                         ;AL C7S Y DEJA EL NIBLE MENOS
                         ;SIGNIFICATIVO (CTE) EN A
LIMPIA.  EQU     03BFH   ;LIMPIA EL AREA DE MEMORIA DEL
                         ;DESPLEGADO
DESP1.   EQU     0074H   ;RECIBE EN HL LA DIRECCION DEL
                         ;MENSAJE A DESPLEGAR SOLAMENTE
                         ;EN EL CAMPO DE DIRECCIONES DEL
                         ;DESPLEGADO Y LO DESPLEGA
LLAC1.   EQU     0681H   ;DESPLEGA LO QUE ENCUENTRA EN
                         ;EL AREA DE MEMORIA DEL DESPLEGADO,
                         ;SENSA EL TECLADO Y NO REGRESA
                         ;HASTA QUE UNA TECLA SEA PULSADA
ERROR.   EQU     0084H   ;DESPLEGA MENSAJE DE ERROR Y
                         ;REGRESA CON A=0
BUFF2.   EQU     2BD8H   ;BUFFER # 2
BUFF1.   EQU     2BDCH   ;BUFFER # 1
PILA1.   EQU     2AE0H   ;STACK POINTER
BFF98.   EQU     070DH   ;EL CONTENIDO DE LOS BUFFER 9 Y 8
                         ;ES COLOCADO EN EL CAMPO DE DIRECCIONES
                         ;DEL AREA DE MEMORIA DEL DESPLEGADO

         ORG 2AE1H
SUFF6.   DS 4            ;DIRECCION SALTO DE TEC
NOM.     DS 2            ;NOMBRE EN BINARIO
NOMA.    DS 4            ;NOMBRE EN ASCII
CANT.    DS 2
TABLA.   DS 6            ;DIRECCION DE TABLA DE LETRERO
CONF.    DS 2
KOK.     DS 1
CYC.     DS 3

         ORG 1000H
TEC.     MACRO #TACO,#EJECO
                         ;DEFINE FUNCION TECLA (DIRECCION
```

```
                                    ;TABLA, SALTO>
        LD HL,#TACO
        LD (TABLA),HL               ;(TABLA)=DIRECCION TABLA
        LD HL,#EJECO
        LD (BUFF6),HL               ;(BUFF6)=DIRECCION SALTO
        CALL TEC1                   ;LLAMA A TEC1
        JP SAL                      ;VE A SAL
        MEND                        ;FIN DE FUNCION
START   LD A,10111100B              ;PALABRA DE CONTROL
                                    ;MODO 1
                                    ;PUERTO A ENTRADA
                                    ;PUERTO B SALIDA
                                    ;PUERTO C ALTO ENTRADA
                                    ;PUERTO C BAJO SALIDA
        OUT(07H),A                  ;ESCRIBE PALABRA DE CONTROL
        LD SP,PILA1                 ;SP=STACK POINTER
        CALL LIMPIA                 ;LLAMA A LIMPIA
        LD HL,TAPESO                ;HL= DIR TABLA PESO
        CALL DESP3                  ;LLAMA A DESP3
COMA.   CALL LLAC1                  ;LLAMA A LLAC1
        CALL Z,ERROR                ;SI TECLA = 'C' COMANDO
        JR Z,COMA                   ;A =ENTONCES
                                    ;       A =LLAMA A ERROR
                                    ;       B =VRA A COMA
                                    ;B =DE LO CONTRARIO CONTINUA
        LD HL,DIR1-1                ;VER (DIR1,DIR2,DIR3,DIR4)
        LD A,(BUFF2)                ;SI TECLA = '1'(GO TO)1, COM2, OTRO)
        ADD A,L
        LD L,A
        JP(HL)
DIR1.   JP REH                      ;VE A REH
        NOP
DIR2.   JP DIR4                     ;VE A DIR4
        NOP
DIR3.   JP DIR4                     ;VE A DIR4
        NOP
DIR4.   CALL ERROR                  ;LLAMA A ERROR
        LD SP,PILA1                 ;SP=STACK POINTER
        JR COMA                     ;VE A COMA
SAL.    CALL ARLD1                  ;LLAMA A ARLD1
        CALL 5FF98                  ;LLAMA A 5FF98
        CALL LLAC1                  ;LLAMA A LLAC1
        JR SAL                      ;VE A SAL
REH.    CALL REH1
        JP START
REH1.   TEC TAREH,CONREH,FUNCION TEC (DIRECCION TABLA REH,
                                    ;SALTO CON REH)
```

```
CONFRM. CALL BUFN          ;LLAMA A BUFN
ENTRA1. CALL LL8A          ;LLAMA A LL8A
ENTRA2. CALL OSO           ;LLAMA A OSO
ENTRA3. LD B,04
        LD HL,NOMA
        CALL OUTCRG
        CALL OT
        RET
```

```
;*****************************************************************
;*                                                               *
;*                                                               *
;*                  FIN DE PROGRAMA PRINCIPAL                    *
;*                                                               *
;*                                                               *
;*****************************************************************
```

```
;*****************************************************************
;*                                                               *
;*                                                               *
;*                       SUBRUTINAS                              *
;*                                                               *
;*                                                               *
;*****************************************************************
```

```
;*****************************************************************
;*                                                               *
;*  ARLD1       TOMA LA ULTIMA TECLA PULSADA Y                   *
;*              -SI ES DATO, ROTA A LA IZQUIERDA LOS CUATRO NIBLES*
;*              EN LAS DOS LOCALIDADES DE MEMORIAS CONTIGUAS, DE LAS*
;*              CUALES LA MAS ALTA ES DIRECCIONADA POR HL. COLOCA *
;*              ESA ULTIMA TECLA COMO EL NIBLE MENOS SIGNIFICATIVO*
;*              DE (HL-1) Y REGRESA A QUIEN LO LLAMO.            *
;*              -SI ES LA TECLA 'OTRO' ENTONCES EJECUTA INTRUCCION*
;*              -SI LA TECLA ES CUALQUIER OTRO COMANDO DESPLEGA   *
;*              MENSAJE DE ERROR Y BRINCA A DIR1                 *
;*              ENTRADA HL-PONTERO DE LOCALIDADES A ROTAR        *
;*                     BUFFER0-ULTIMA TECLA PULSADA              *
;*              SALIDA.(HL) Y (HL-1)                             *
;*              MODIFICA A,F,C,DE                                *
;*              LLAMA ALDSQ                                      *
;*                                                               *
;*****************************************************************
```

```
            ARLD1.  CALL ALDS2      ;SENSA TECLADO
                    LD A,(BUFF1)    ;SI TECLA <> DATO ENTONCES
                    BIT 4,A
                    JR Z,AARLD
                    LD A,(BUFF2)            ;SI TECLA <> "OTRO" ENTONCES
                    CP 10H                          ;INICIO
                    JP NZ,DIR4                      ;VE A DIR4
                                                    ;DE LO CONTRARIO
                    POP DE                  ;FINSI
                    LD HL,(BUFF6)
                    JP(HL)              ;FINSI
            AARLD.  LD A,C              ;CONVIERTE LA TECLA PULSADA A
                    DEC HL              ;HEXADECIMAL
                    RLD
                    INC HL
                    RLD
                    RET                 ;REGRESA
;*****************************************************
;*                                                   *
;*      BUFN.COLOCA (BUFF9) EN (NOM)                 *
;*              ENTRADA.NINGUNA                      *
;*              SALIDA..(NOM)                        *
;*              MODIFICA.HL                          *
;*              LLAMA.NINGUNA                        *
;*                                                   *
;*****************************************************
            BUFN.   LD HL,(BUFF9)   ;NOMBRE = DESPLEGADO
                    LD (NOM),HL
                    RET             ;REGRESA
;*****************************************************
;*                                                   *
;*      BUFC.COLOCA (BUFF9) EN (CANT)                *
;*              ENTRADA.NINGUNA                      *
;*              SALIDA.(CANT)                        *
;*              MODIFICA.HL                          *
;*              LLAMA.NINGUNA                        *
;*                                                   *
;*****************************************************
            BUFC.   LD HL,(BUFF9)   ;CANTIDAD = DESPLEGADO
                    LD (CANT),HL
                    RET             ;REGRESA
```

```
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;+                                                                           +
;+       TEC1.SENSA EL TECLADO Y ESPERA EL PRIMER DIGITO DE LA               +
;+              DIRECCION O NOMBRE DE LA INSTRUCCION. ESTE DIGITO            +
;+              DEBE SER HEXADECIMAL SI ESTO OCURRE REGRESA A                +
;+              QUIEN LO LLAMO. DE LO CONTRARIO (TECLA FUE COMAN             +
;+              DO) DESPLEGA ERROR Y SALTA A DIR 4.                          +
;+              ENTRADA.PRIMER DIGITO HEXADECIMAL                            +
;+              SALIDA.DESPLEGADO                                            +
;+              MODIFICA.HL                                                  +
;+              LLAMA.LIMPIA, DESPD, LLAC1, BFF9B                            +
;+                                                                           +
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

TEC1.   CALL LIMPIA      ;LIMPIA ZONA DE DESPLEGADO
        LD HL,(TABLA)    ;DESPLEGA LETRERO QUIEN TE LLAMO
        CALL DESPD
        CALL LLAC1       ;SENSA TECLADO
        CP 10H           ;SI TECLA < > DATO ENTONCES
        JP C,DIR4                ;VE A DIR4
                         ;FINSI
        CALL BFF9B       ;LIMPIA ZONA TECLADO Y DESPLEGA CEROS
        RET              ;REGRESA
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;-                                                                           +
;+       LLBA EL NOMBRE DEL PROGRAMA QUE ESTA EN DOS LOCALIDADES             +
;+              EN DECIMAL LOS COLOCA EN CUATRO LOCALIDADES TAMBIEN          +
;+              EN DECIMAL SEPARANDOLOS EN NIBLES                            +
;+              ENTRADA.NINGUNA                                              +
;+              SALIDA.(NOMA) CUATRO LOCALIDADES                             +
;+              MODIFICA.HL, DE, A                                           +
;+              LLAMA.NINGUNA                                                +
;+                                                                           +
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

LLBA.   LD HL,NOMA1      ;APUNTA A LOS 2 DIGITOS MAS SIGNIFICATIVOS
        LD DE,NOMA       ;APUNTA A LA TABLA DEL NOMBRE EN 4 DIGITOS
        CALL BA          ;SEPARA 1ER DIGITO
        CALL BA          ;SEPARA 2DO DIGITO
        DEC HL
        CALL BA          ;SEPARA 3ER DIGITO
BF      XOR A            ;SEPARA 4TO DIGITO
        RLD
        LD(DE),A
        INC DE
        RET              ;REGRESA
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;+                                                                           +
;+       OSO.CONVIERTE EL CONTENIDO DE NOMA EN DECIMAL A CODIGO              +
;+              ASCII Y LO COLOCA EN EL MISMO LUGAR.                         +
;+              ENTRADA.(NOMA) CUATRO LOCALIDADES                            +
;+              SALIDA.(NOMA) CUATRO LOCALIDADES                             +
;+              MODIFICA.A, B, HL                                            +
;+              LLAMA.NINGUNA                                                +
;+                                                                           +
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

OSO.    LD B,4           ;PARA CONT-1 HASTA 4 EN PASO HAGA
        LD HL,NOMA
OS.     LD A,(HL)                ;A-NOMA (CONT)
        CALL CDAS                ;A-ASCII (A)
        LD(HL),A                 ;NOMA (CONT)-A
        INC HL
        DJNZ OS          ;FINPARA
        RET              ;REGRESA
```

```
;*****************************************************************
;*                                                               *
;*      CDAS.CONVIERTE LO QUE ESTA EN DECIMAL EN "A" EL NIBLE    *
;*           MENOS SIGNIFICATIVO EN CODIGO ASCII                 *
;*           ENTRADA.A                                           *
;*           SALIDA.A                                            *
;*           MODIFICA.A                                          *
;*           LLAMA.NINGUNA                                       *
;*                                                               *
;*****************************************************************

            CDAS.   CP  0AH             ;A- ASCII (A)
                    JR  C,MAS
                    ADD A,7
            MAS.    ADD A,30H
                    RET                 ;REGRESA

;*****************************************************************
;*                                                               *
;*      OUTCRO.TRANSMITE AL SISTEMA CROMEMCO UNA CANTIDAD DE     *
;*           DATOS, ESTA CANTIDAD DEBE DE ESTAR EN EL            *
;*           REGISTRO B, Y EL BUFFER A TRANSMITIR DEBE DE        *
;*           ESTAR APUNTADO POR EL REGISTRO HL                   *
;*           ENTRADA.B, HL                                       *
;*           SALIDA.DATOS POR EL PUERTO DE SALIDA                *
;*           MODIFICA.C, A, HL, B                                *
;*           LLAMA.NINGUNA                                       *
;*                                                               *
;*****************************************************************

    OUTCRO. LD  C,PUERTB        ;LEE IBF DE CROMENCO
            JR  PRIME
    NOLIST. IN  A,(PUERTC)
            BIT 7,A             ;SI IBF=0
            JR  Z,NOLIST        ;A.=ENTONCES VE A NOLIST
                                ;B.=DE LO CONTRARIO CONTINUA
    PRIME.  OUTI                ;ESCRIBE A CROMENCO (DIR TABLA)
                                ;# DE CARACTERES = # DE CARACTERES - 1
                                ;DIR TABLA = DIR TABLA + 1
            JR  NZ,NOLIST       ;SI # DE CARACTERES <> 0
                                ;A. = ENTONCES VE A NOLIST
                                ;B. = DE LO CONTRARIO CONTINUA
            RET
;*****************************************************************
```

```
;*****************************************************************
;*                                                               *
;*      INCRO.RECIBE DEL SISTEMA CROMENCO UNA CANTIDAD DE DATOS  *
;*            ESTA CANTIDAD DEBE DE ESTAR EN EL REGISTRO B, Y    *
;*            LOS DEPOSITA EN UN BUFER QUE DEBE DE ESTAR         *
;*            APUNTADO POR EL REGISTRO HL                        *
;*            ENTRADA.B, HL                                      *
;*            SALIDA.PUERTO DE ENTRADA DEL KIT                   *
;*            MODIFICA.HL, B, C                                  *
;*            LLAMA.NINGUNA                                      *
;*                                                               *
;*****************************************************************

INCRO.  LD C,04H        ;C = PUERTO DE ENTRADA
NOHAY.  IN A,(PUERTC)   ;LEE STB DEL 8255
        AND 00100000B   ;SI STB = 0
        JR Z,NOHAY      ;A.=ENTONCES VE A NOHAY
                        ;B.=DE LO CONTRARIO CONTINUA
        INI             ;LEE (DIR TABLA) DE CROMENCO
                        ;# CARACTERES = # CARACTERES - 1
                        ;DIR TABLA = DIR TABLA + 1
        JR NZ,NOHAY     ;SI # CARACTERES <> 0
                        ;A.=ENTONCES VE A NOHAY
                        ;B.=DE LO CONTRARIO CONTINUA
        RET


OT      LD B,3          ;REPITE HASTA QUE CANTIDAD <> 0
        LD HL,CYO       ;INICIO
        CALL INCRO      ;TRANSMITE CANTIDAD
        LD A,(CYO)
        CP 00
        RET Z
        LD A,(CYO)
        LD B,A
        LD HL,(CYO+1)
        CALL INCRO      ;TRANSMITE ORIGEN
        JR OT           ;FIN
                        ;FIN REPITE

PUERTC. EQU     06H     ;PUERTO C
CONPPI. EQU     07H     ;PUERTO CONTROL
PUERTB. EQU     05
BIT61   EQU     01000000B
        TAREH.  DB      50H,79H,00,76H,54H,5CH
        TAFESC  DB      00,71H,79H,6DH,33H,00
        TARE.   DB      00,00,50H,79H
        TATR.   DB      00,00,78H,30H
                END START
```