

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL BUENOS AIRES

Departamento de Electrónica

Cátedra: Técnicas Digitales II - Plan 1995 - Versión 2007

CAPITULO I: Introducción a los Microprocesadores.

Autor: Ing. Marcelo E. Romeo

Objetivo: En este capítulo se introducirán los fundamentos de la lógica programada que derivarán en la arquitectura elemental de una computadora así como en la forma de conexión de sus componentes constitutivos y su vinculación con el mundo exterior.

Por otro lado se introducirán elementos de vocabulario específico tanto para el posterior desarrollo de este curso como para las asignaturas sucesoras.

1. Planteo General.

A principios de la década del 70, quien pensaba en una computadora imaginaba un sistema voluminoso, muy costoso, en ambientes refrigerados, solamente operable por especialistas y aplicable exclusivamente a tareas muy específicas ya que su costo por unidad de tiempo hacia prohibitivo su empleo en tareas cotidianas¹.

Han bastado tres décadas para que la situación cambiara radicalmente. Las computadoras han invadido todos los ámbitos del hombre contemporáneo y se las encuentra en la electrónica de entretenimiento (juegos electrónicos), en los artículos del hogar (lavarropas, televisores), en el automóvil (sistema de control, taxímetros) en la investigación científica (sistemas de adquisición de datos, equipos de diagnóstico médico), en la industria (sistemas de control automático), en las comunicaciones (modems, sistemas de conmutación) y en la mas variada gama de aplicaciones imaginables.

Si bien los factores han sido múltiples, el mas importante fue la integración de gran cantidad de componentes electrónicos en una pequeña pastilla de silicio de altísima pureza, lo que permitió aumentar la densidad de funciones y operaciones por unidad de volumen, a la vez que disminuir increíblemente los costos (los primeros ejemplares del microprocesador 8080 se comercializaban en 1974 en 420 dólares, mientras que actualmente se lo consigue por menos de 1 dólar)².

¹ Como dato ejemplificativo, podemos recordar que la computadora central del INTI (Instituto Nacional de Tecnología Industrial), instalada en 1980, atendía 20 terminales y requería una superficie de aproximadamente 120 m2 con doble sistema de aire acondicionado, con piso flotante ("hueco", para pasar el cableado por debajo), con discos rígidos de 250 Mbytes del tamaño de un lavarropas. Actualmente sus funciones podrían ser reemplazadas por una PC Pentium.

² Gordon Moore, Uno de los fundadores de Intel Corporation, determinó empíricamente lo que hoy se conoce como Ley de Moore. Ella dice que cada 18 meses se duplica la complejidad (cantidad de transistores por integrado), manteniendo el costo.

Estos dos factores han permitido a las microcomputadoras desplazar componentes y equipos electromecánicos, aumentando el repertorio de funciones, la eficiencia y la confiabilidad mientras que se disminuyó el costo, el consumo y las dimensiones físicas.

2. Introducción.

El objeto de este capítulo es la introducción al estudio de los microprocesadores que serán utilizados como "cerebros" de las microcomputadoras.

2.1 Lógica cableada vs. Lógica programada.

Durante la segunda mitad de la década del '60 aparecieron los circuitos integrados, en los que se agrupaban dentro de un único encapsulado más de 25 transistores. La reducción de tamaño resultante permitió encarar nuevas aplicaciones, las que a su vez exigían mayor grado de complejidad en los dispositivos, obligando a los fabricantes de circuitos a incrementar la cantidad de componentes por unidad y a la vez que generar nuevos tipos de circuitos que cumplieran diversas funciones.

En esta concepción circuital, llamada lógica cableada, la personalidad del sistema se hallaba en los componentes circuitales (hardware) utilizados y en la forma de conexión. Un error en el diseño o un cambio en las especificaciones del proyecto, llevaba normalmente a tener que rediseñar completamente el circuito, con el consiguiente incremento de costo y tiempo de desarrollo.

Al comenzar la década del 70, con la experiencia adquirida en el desarrollo de computadoras y analizando la creciente miniaturización de los componentes, algunos fabricantes de dispositivos semiconductores³, decidieron encarar el diseño de circuitos secuenciales con características de pequeñas computadoras (micro-computadoras) con los siguientes elementos constitutivos básicos:

a) Una unidad central de proceso o microprocesador (MPU, Microprocessor Unit, CPU, Central Processing Unit), formada por una unidad aritmético-lógica, encargada del procesamiento de la información y de la toma de decisiones, y por una unidad de control, con la función del guiado de las señales binarias desde la fuente hasta el destino de las mismas, por medio de multiplexores y del control del flujo y la adecuada generación y temporización de las señales. Esta CPU se constituye por un reducido número de circuitos integrados.

b) Circuitos de comunicación con el mundo exterior o interfaces por el que ingresarán las excitaciones digitales de entrada y partirán las respuestas digitales de salida.

c) Una memoria semiconductor en la que se encontrarán las instrucciones (programa) a ejecutar y los datos (variables y resultados) utilizadas por el proceso en ejecución.

La personalidad de un sistema así constituido, se halla tanto en los elementos circuitales como en el programa. En efecto, cambiando el programa, una misma plaqueta de circuito impreso podrá ser utilizada para diferentes aplicaciones. Esta facilidad permitió aumentar considerablemente la utilización de un mismo conjunto de componentes, reduciendo drásticamente los costos ya que no se requería el rediseño del circuito para cada aplicación. Solo había que rescribir el programa (quizás solo partes del mismo, pues se podía mantener el núcleo), contando además con bibliotecas facilitadas por los fabricantes para simplificar la introducción del nuevo método de diseño.

Esta concepción de sistemas recibió el nombre de **lógica programada** y su gran aceptación se basó en:

a) Menor cantidad de componentes, ya que las funciones complejas las realizaba el programa y no los elementos circuitales.

³ Intel recibió un pedido de desarrollo de una calculadora. En lugar de implementarla con la metodología estándar del diseño de un circuito combinacional-secuencial, decidió encararla con una idea de versatilidad de un producto que sirviera para más de una aplicación. Fue el 4004 un fallido microprocesador de 8 bits en tecnología PMOS. Su experiencia sirvió para implementar una terminal de video de 8 bits con el 8008. Éste a su vez dio lugar al 8080, primer microprocesador comercial de 8 bits en tecnología NMOS.

b) Facilidad de corrección de errores, pues normalmente basta con cambiar una parte del programa de gestión.

c) Sencilla adaptación a los cambios de las especificaciones, también basta (en general) con cambiar el programa.

d) Poder realizar el diseño de un único circuito (y un único impreso) para múltiples aplicaciones y que fuera el programa el que cumplía las funciones específicas solicitadas.

Dicha aceptación llevó a popularizar rápidamente el uso del microprocesador, permitiendo disminuir rápidamente los costos.

3. Sistemas de Propósito general y dedicados

3.1 Descripción de ambos

Una computadora puede utilizarse para múltiples propósitos, como por ejemplo una computadora personal que se puede utilizar para ejecutar diversos programas sin modificaciones en su circuitería.

Por otro lado, una computadora dedicada (o de propósito específico) fue concebida para actuar en una sola aplicación y son incapaces de desarrollar otras actividades. Ejemplos de estas computadoras, son los controles de lavarropas, del sistema de encendido de un televisor, el sistema de control de un tren de alta velocidad, de control de un sistema de DVD, etc.

3.2 Ejemplo

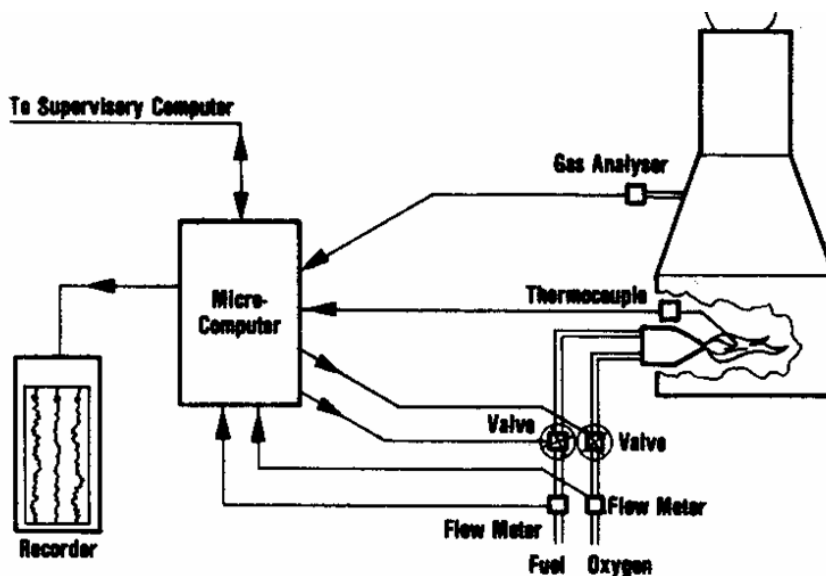


Fig. 1: Sistema de control de llama de un horno

Analicemos el sistema de la Fig. 1. En él vemos un sistema de control de la llama de un horno. Tenemos dos válvulas que controlan el paso de combustible y de aire. Una termocupla mide la temperatura del horno y a su vez un analizador de gases determina si la combustión es adecuada o hay exceso o falta de oxígeno en el proceso. El sistema de control realimentado regulará las válvulas con el objetivo de optimizar la combustión.

Este sistema fue diseñado y optimizado para esta actividad específica y no podrá utilizarse (salvo modificaciones drásticas) para otra aplicación.

3.3 Diseño de un sistema dedicado.

El objetivo de esta asignatura es tratar con sistemas dedicados, fundamentalmente en lo que sea su diseño y optimización. Por ello analizaremos las etapas de un diseño bajo una perspectiva empírica.

3.3.1 Requerimientos

Se nos acercan las especificaciones del producto. Es decir lo que el cliente necesita, según su interpretación que no tiene porqué ser técnica.

3.3.2 Especificación

El diseñador escribe un documento con su interpretación (lo más detallada posible) de los requerimientos del cliente. Se la entrega al mismo para su análisis. El cliente las corrige y la entrega al diseñador que a su vez hace una contrapropuesta técnica de diseño.

Luego de repetir varias veces este proceso, queda lista una especificación, es decir los requerimientos técnicos del producto a desarrollar acordados por ambas partes.

3.3.3 Arquitectura.

Se busca la arquitectura circuital que mejor satisfaga la especificación (utilizar uno o varios microprocesadores, microcontroladores, etc). Incluye el análisis de la vigencia de dicha arquitectura, es decir si se entiende que algún componente que pueda afectar a la arquitectura (los trascendentes) pueda dejar de fabricarse en un lapso tal que condicione al proyecto.

3.3.4 Diseño del componente

Ahora se diseñará en detalle el circuito, se seleccionarán los circuitos integrados, los transistores, se le darán valores a resistores y capacitores y se escribirá el programa. El mismo se depurará y se optimizará.

Se realizarán diagramas temporales con las tareas, sus interrelaciones (que tareas deben terminarse para poder empezar la siguiente), quien será el responsable del diseño y quien de la verificación (distintas personas).

De ser necesario se rediseñará el circuito y se reescribirá el programa a fin de optimizarlo.

3.3.5 Integración al sistema

Se integrará el sistema con microprocesador/microcontrolador al sistema completo y de ser necesario se repetirán los puntos 3.3.3 y 3.3.4 tantas veces como sea necesario. Esto quiere decir, si en el sistema completo había motores y debían ser controlados por nuestro diseño, se verán si trabajan adecuadamente, si había un gabinete prediseñado, si el circuito cabe en el mismo y su operación manual es la adecuada, etc.

4. Clasificación de las computadoras

A fin de poder encarar adecuadamente nuestro estudio, subclasificaremos las máquinas utilizadas en el ámbito de la informática según diversos criterios.

Esta clasificación no es excluyente pues una misma computadora se encontrará en cada uno de los rubros de clasificación (es decir que una computadora podrá ser un mainframe RISC, multiusuario y en red).

4.1 Tamaño y prestaciones

4.1.1 Grandes computadoras.

Propias de los grandes centros de cómputos. Pueden atender una gran cantidad de usuarios y/o procesos simultáneamente. Son caras, de grandes prestaciones y se emplean para aplicaciones de gran magnitud, como la predicción meteorológica, el manejo de grandes bases de datos (AFIP, dirección de Rentas, juegos de azar, obras sociales, etc.). Ejemplos de esta clase son la CRAY-2, la CDC 4000.

Una aplicación original de este tipo de computadoras, se presentó en la necesidad que tuvieron en la década del 70, Holanda e Inglaterra de elaborar un pronóstico meteorológico muy confiable para el Mar del Norte.

En ese mar, con tormentas aterradoras, se había encontrado petróleo en la primera época de alto precio del crudo. Si no se avisaba con tiempo de la posibilidad de tormentas, los enormes buques petroleros corrían el riesgo de destruirse (con daños económicos y ecológicos) y destruir las plataformas de extracción. Tampoco podía dejarse de operar las plataformas con la aparición de la primera nube en el horizonte.

El problema se resolvió instalando varios centenares de estaciones automáticas de toma de datos (presión y temperatura atmosférica, humedad, viento, corrientes marinas, salinidad y temperatura del agua, etc.) y transmitir las periódicamente a una central que se encargaba de elaborar un primer pronóstico y comparar las condiciones con un historial que los británicos tenía con más de 100 años de archivo.

Se necesitaba, por un lado una enorme capacidad de almacenamiento de historia (y generación de nueva historia) y procesamiento de datos llegados desde las estaciones de adquisición. Todo esto se implementó por medio de una CRAY-1. El último grito de la tecnología en ese momento. Basado en la familia de lógica no saturada ECL, requería montar las plaquetas de circuito impreso sobre bastidores de aluminio hueco por el que circulaba freón para refrigerarla y en forma cilíndrica para que el tiempo de propagación de todas las señales hacia los buses centrales fueran idénticos.

Como detalle constructivo, en la Fig. 2 vemos que la base saliente de la computadora ¡Es la fuente de alimentación!



Fig. 2. Gran computadora. Cray I

4.1.2 Mini-computadoras.

El escalón inferior a las anteriormente mencionadas. Su velocidad es inferior a las anteriores, puede atender menos usuarios, pero su costo hace que sean accesibles a las medianas empresas. Ejemplos son la VAX 8600, VAX 11/780, IBM 4341, 4381 sistema AS400 y la familia Microvax.

En un supermercado, cuando pasamos por una caja registradora, en realidad estamos activando una computadora personal, que se comunica con una minicomputadora que se encarga de enviar a la caja el precio del producto desde una base de datos centralizada, descontar ese producto del stock, determinar si se llegó al stock mínimo de ese producto y ordenar a los repositorios que coloquen nuevas unidades en las góndolas y, de ser necesario, enviar órdenes de compra a los proveedores estipulando lugar y fecha de entrega, según la velocidad de rotación del producto previamente calculada. También se descontarán los montos de IVA que afectan al producto comprado y lo depositará en la cuenta de IVA.

Éste conjunto de operaciones, además de las comunicaciones con las tarjetas de crédito esperando las autorizaciones, son desarrolladas por una minicomputadora.

4.1.3 Microcomputadoras Personales.

Son aquellas que utilizan como unidad aritmético-lógica un microprocesador y tienen como salida de video un monitor específico con, por lo menos una definición horizontal de 80 columnas (no admitiendo salida sobre un televisor). Como ejemplo, podemos citar las iMAC de Apple, IBM PC y compatibles⁴.

4.1.4 Microcomputadoras hogareñas.

Son aquellas que han sido diseñadas para que su salida de video se realice sobre un televisor común entrando tanto por video como por antena. Como ejemplos de estas máquinas, mencionaremos la Apple II, Commodore 16, 64 y 128, Sinclair, TI 99, MSX, etc. Debido al muy bajo precio de las microcomputadoras personales, han desaparecido del mercado.

En la actualidad, este rubro se halla cubierto por los video juegos (Nintendo, Play Station, etc.) que emplean un televisor como salida de imagen.

4.2 Conectividad y usuarios

Por otro lado, es posible efectuar otra clasificación de las computadoras, según la cantidad de usuarios que puedan operar sobre ella en:

⁴ Aprovechamos para introducir el concepto de compatibilidad. Quiere significar que se comporta de manera similar a una IBM PC, a pesar de ser fabricada por otra empresa y con componentes distintos.

4.2.1 Computadoras Monousuario.

Son aquellas que, como su nombre lo indica, permiten la operación de un solo usuario.

4.2.2 Computadoras Multiusuario.

Permiten la conexión de más de un usuario. Cada uno de ellos, tiene la impresión que la computadora esta a su entera disposición. Ello se debe a que la computadora se encuentra ejecutando algún programa que fracciona el uso de la misma entre todos los usuarios, de forma que (al menos teóricamente) ningún usuario se entere de la existencia de los demás.

Esta arquitectura se da habitualmente en los grandes organismos públicos (ANSES, Direcciones Provinciales de Rentas, AFIP, etc.) o empresas de aviación, bancos, etc. en las que se debe manejar una única y enorme base de datos como recurso principal y todos los usuarios deberán consultar e interactuar centralizadamente con la misma.

4.2.3 Redes de computadoras.

Es una situación intermedia entre las dos opciones anteriores. Un conjunto de computadoras monousuario son interconectadas (por medio de cable coaxial o cable telefónico, llamado UTP - **U**nshielded **T**wisted **P**airs, pares trenzados no blindados-) entre sí y eventualmente con computadoras de mayor magnitud, compartiendo recursos como ser impresoras, discos, unidades de cinta magnética, lectores de CD-ROM, etc.

Las redes se pueden presentar en dos configuraciones:

4.2.3.1 Servidor centralizado

Existe una computadora central (file-server), en la que un programa asigna recursos y fija derechos a cada usuario (por ejemplo en que horarios puede conectarse, sobre que impresoras y en que PCs puede trabajar).

Este file server suele tener un disco de tamaño muy importante que permite que todos las demás estaciones de trabajo compartan ese disco tanto para consultar la información almacenada en él como para actualizar su información.

4.2.3.2 Vinculación horizontal (peer - to - peer)

La segunda es aquella en la que cada computadora pone de un conjunto de recursos (seleccionables por el usuario) a disposición de los demás usuarios del sistema, por ejemplo disqueteras, parte de sus discos, impresoras, etc. En esta segunda configuración, todas las computadoras están al mismo nivel jerárquico y se denomina *peer to peer*.⁵

4.3 Complejidad de su repertorio de instrucciones

Según la complejidad de sus instrucciones (u órdenes) una computadora puede dividirse en **CISC** o **RISC**.

4.3.1 Computadoras CISC

CISC es la abreviatura de **C**omplex **I**nstruction **S**et **C**omputers o Computadoras de Repertorio de Instrucciones Complejas.

Son aquellas computadoras que disponen de un conjunto de instrucciones complejas y poderosas. Su arquitectura fue diseñada de forma de poder soportar esas instrucciones.

⁵ Debe destacarse que al habilitar algún disco para ser masivamente compartido, todos los usuarios de la red puede acceder al mismo, lo cual es particularmente riesgoso al acceder a Internet. Todos los usuarios de Internet podrían acceder a ese recurso habilitado para compartirse.

4.3.2 Computadoras RISC.

RISC es la abreviatura de **R**educed **I**nstruction **S**et **C**omputers o Computadoras de Repertorio de Instrucciones Reducido. Esta definición no debe interpretarse como despectiva, sino que representa un importante avance en el desarrollo de la arquitectura de las computadoras.

Estadísticamente se demostró que las instrucciones muy poderosas de las computadoras CISC son empleadas sólo por algunos usuarios y muy esporádicamente. Por tal motivo se decidió diseñar una computadora optimizada para ejecutar a muy alta velocidad un reducido número de instrucciones, pero las más frecuentemente utilizadas. Las instrucciones complejas se obtendrán ejecutando un conjunto de instrucciones disponibles en lugar de ser una única instrucción como en las CISC.

En la actualidad, los microprocesadores de más nueva producción como por ejemplo los de la familia Pentium, presentan una arquitectura RISC particular. Debido al requerimiento de compatibilidad en la ejecución de programas diseñados para sus predecesores de la familia 80x86, se necesitarán varias microinstrucciones RISC elementales de alta velocidad para concretar una instrucción

5. Estructura básica de una microcomputadora.

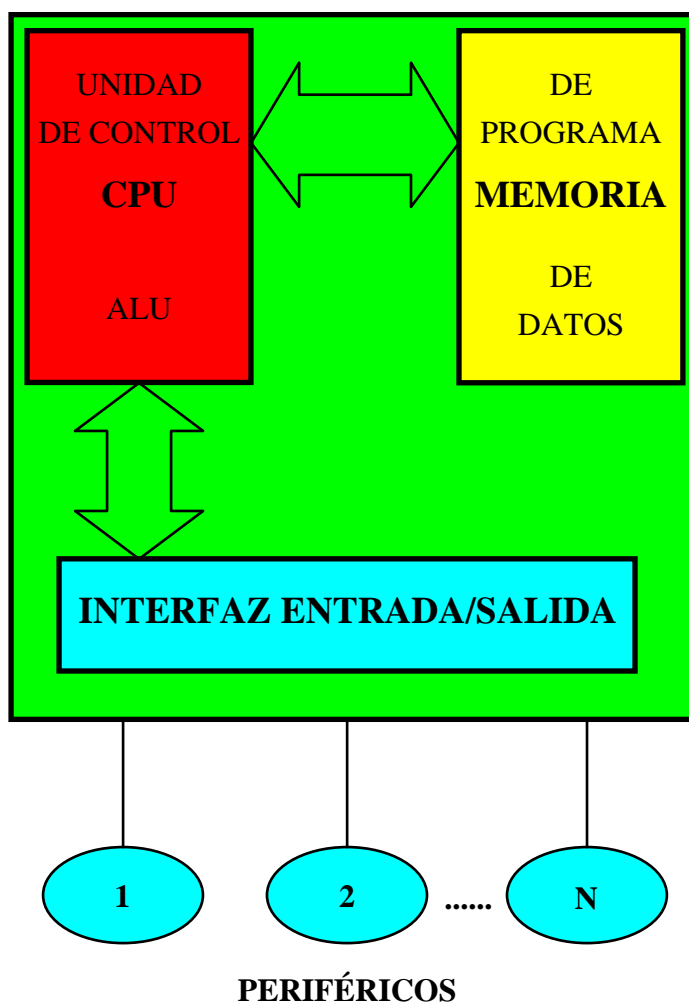


Fig. 3 _ Esquema elemental de una computadora.

En principio una microcomputadora no es esencialmente diferente en estructura a los grandes sistemas de cómputo. Ambas poseen:

a) Unidad Central de Proceso (CPU), formada por una unidad de control y una unidad de cálculo o aritmético-lógica.

- b) Memoria.
- c) Interfaces⁶ a periféricos.

Estas definiciones deben interpretarse conceptualmente y no físicamente. Por ejemplo en el familiar caso de las computadoras personales (PCs), la CPU será el microprocesador (y no todo el gabinete, mal llamado CPU), la memoria es inconfundible, y como periféricos tendremos: indudablemente la impresora, las unidades de disco (a pesar que se hallen dentro del gabinete de la PC), los modems (ya sean internos o externos), el teclado y el monitor.

Los dispositivos de entrada / salida serán, por ejemplo las placas de interfaz serie (COMs) o paralelo para impresora (LPTs), la controladora de video, etc.

Según la constitución de la computadora, la misma podrá contener un microprocesador, un microcontrolador o un DSP.

1. Microprocesador. Cuando la Unidad Central de Proceso esta constituida por un reducido número de integrados, se dice que esta constituida por un microprocesador.
2. Cuando en un solo circuito se encuentra la Unidad Central de Proceso, la memoria de programa, la de datos e interfaces de entrada/salida, diremos que estamos frente a una microcomputadora en un solo circuito. Hoy en día se las llama microcontroladores.
3. Cuando se requiere procesar la información en tiempo real (por ejemplo el filtrado de la voz del cantante en un recital) se utilizará un Procesador Digital de señales (o DSP), que es solamente un microprocesador especializado en realizar algunas operaciones a gran velocidad.

Como las siglas de los DSP se confunden con las de ProcesaMIENTO Digital de Señales, se esta tendiendo a redenominarlos como procesadores a secas.

6. Arquitecturas de las computadoras

6.1 Arquitectura Von Neumann.

Esta arquitectura de computadoras, ofrece una única memoria central en la que se almacenan palabras binarias que pueden representar en forma indistinta instrucciones, datos u operandos. El procesador puede acceder indiscriminadamente a cualquiera de ellos y esta configuración recibe el nombre de Arquitectura Von Neumann.

La mayor parte de los microprocesadores de uso comercial operan con esta arquitectura.

La tradicional PC es un ejemplo de esta arquitectura, pues al cargar una aplicación (programa) una determinada posición de memoria puede contener parte de una instrucción, mientras que al cargar otra aplicación, la misma posición podrá contener una variable y ser accedida como memoria de datos.

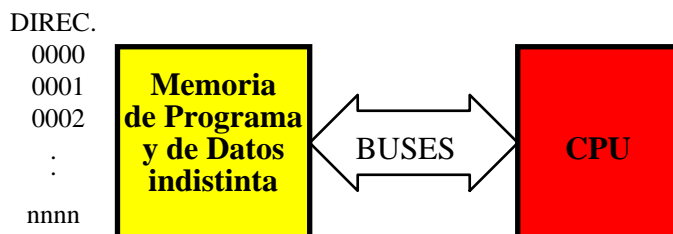


Fig. 4. Computadora bajo Arquitectura Von Neumann

⁶ Utilizaremos el término interfaces como plural de *interfaz* (del árabe faz y no el inglés interface o el químico interfase), pues el concepto que involucra es el de la vinculación entre caras.

6.2 Arquitectura Harvard.

En este tipo de computadoras, existen dos bancos de memoria perfectamente diferenciados. El banco de memoria de programa y el banco de memoria de datos. Las computadoras que responden a esta arquitectura, tienen distintas formas de conexión con la unidad aritmético lógica, y (normalmente) distintas instrucciones para dirigirse a uno u otro banco de memoria. De esta manera pueden coexistir dos posiciones de memoria con igual dirección, pero una posición de memoria en el mapa de programa y la otra en el mapa de datos.

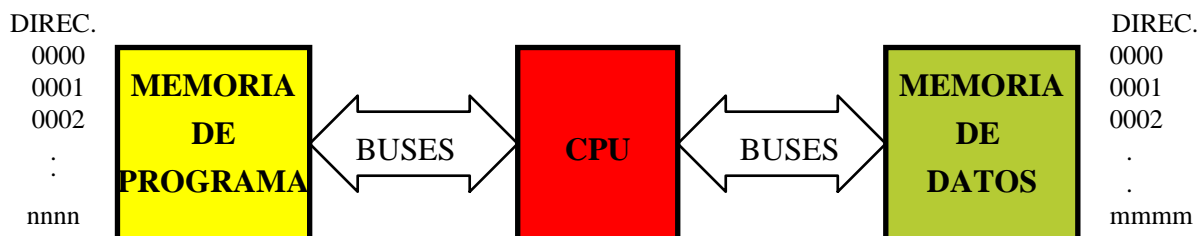


Fig. 5. Computadora bajo Arquitectura Harvard

Esta configuración permite acelerar la operación, ya que se puede acceder simultáneamente a la memoria de programa (para buscar un código de operación) y al mapa de datos (para buscar un operando). Esta operatoria marca una diferencia con las computadoras con una arquitectura Von Neumann, en las que como todo se halla en un único mapa de memoria, se deberá acceder primero a la memoria de programa para lograr el código de operación y luego secuencialmente, para obtener el operando (dato).

Esta arquitectura no es de uso frecuente. Entre los pocos ejemplos comerciales de esta arquitectura se hallan algunas microcomputadoras en un solo circuito (microcontroladores) como el 8048 o el 8051 y algunos dispositivos dedicados al procesamiento digital de señales (DSP), de muy alta velocidad de operación.

En el caso de los DSP, se disponen de dos buses distintos de direcciones y datos, de forma que efectivamente mientras se lee una instrucción de la memoria de programa, se puede estar operando con variables en la memoria de datos.

6.3 Barras ("buses").

Para la transferencia de información dentro de la microcomputadora como con el mundo exterior, se emplean el concepto de barra o bus, estudiado en Técnicas Digitales I.

Aquí simplemente recordaremos que para que puedan concurrir varias salidas a una misma línea de bus (para las entradas solo es necesario respetar la cargabilidad), será necesario que una lógica de control maneje las habilitaciones tri-state de los distintos componentes a fin de que haya un solo dispositivo parlante por vez.

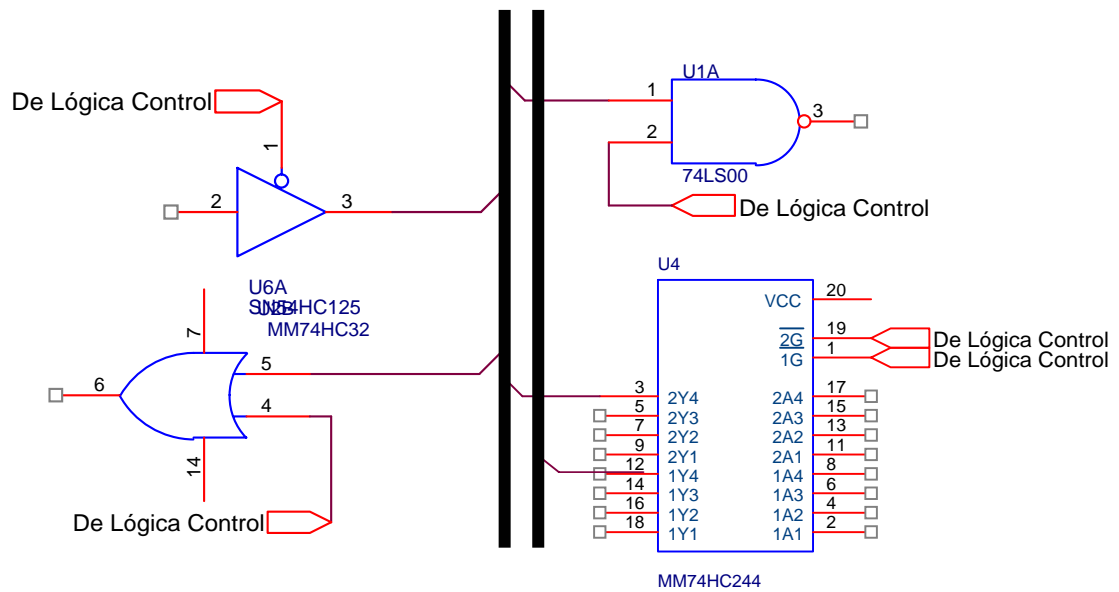


Fig. 6. Conexión a un bus. Múltiples entradas y múltiples salidas comandadas por una lógica de control

En la Fig. 7 se observan tres barras de comunicación entre los componentes de la microcomputadora.

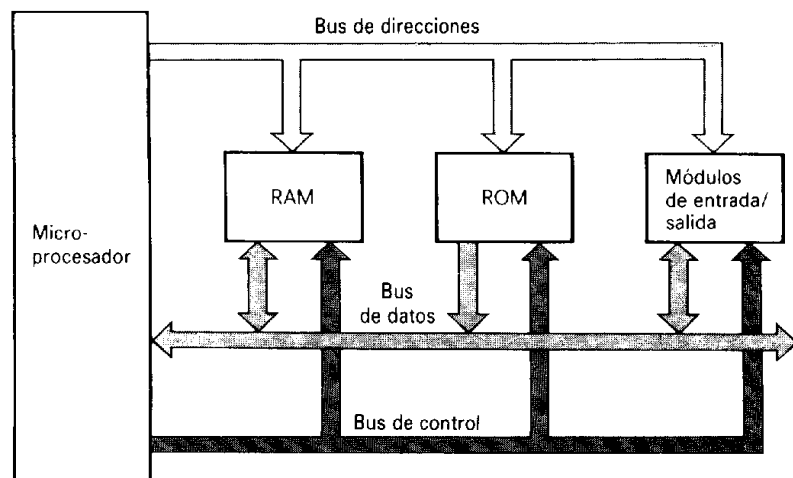


Fig. 7. Comunicación a través de un bus del sistema.

a) BARRA DE DIRECCIONES: permite seleccionar (identificar o direccionar) a la posición de memoria o periférico con el que se desea realizar una acción. Es unidireccional bajo el aspecto de que, normalmente, sólo el procesador puede colocar una palabra binaria sobre el mismo, mientras que los demás componentes de la microcomputadora solo pueden leerla sin escribir sobre esta barra.

b) BARRA DE DATOS: por ella circula la información que se desea leer o escribir. Es bidireccional ya que la información puede salir del procesador (en el caso en que el mismo realice la operación de escritura) o bien puede llegar al mismo (en el caso de la operación de lectura).

c) BARRA DE CONTROL: son las señales de manejo general del sistema de micro cómputo. Es, en general, unidireccional pero tiene señales salientes (ordenes de lectura o escritura) y entrantes (pedidos de interrupción o solicitud de la barra de datos). Groseramente puede decirse que engloba a todo aquello que no es datos o direcciones.

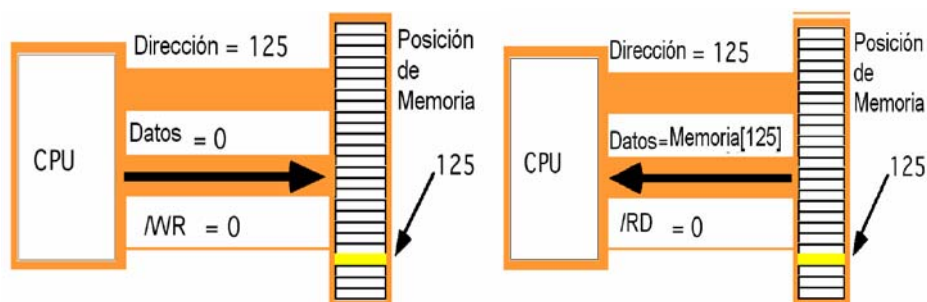


Fig. 8 . Ejemplo del Bus de Control.

Supongamos que deseamos acceder a la posición de memoria 125. Se colocará en el bus de direcciones la palabra binaria 00FFH (125 en decimal). Por el bus de datos viajará la información, pero ¿en qué sentido?, ¿para qué estamos seleccionando esa posición de memoria?. Las señales del bus de control /RD (lectura activa baja) o /WR (escritura) terminarán de proveer la información que necesita la memoria para saber en qué sentido deberá accionar sus buffers (entrante o saliente).

Como resumen y en este caso, podemos decir que el bus de control complementa la información que provee el bus de direcciones y fija el sentido de la información que viaja por el bus de datos.

6.4 Decodificación

Al introducir el concepto de bus, surge la idea de conexión en paralelo de los componentes. De hecho, las memorias y los dispositivos de E/S comparten en paralelo las líneas de dirección, datos y algunas de control.

Si la salida de más de un dispositivo se activa a la vez, nos encontraremos frente a un fenómeno denominado contención del bus que llevará indefectiblemente a la destrucción de los componentes simultáneamente habilitados.

Por tal motivo y como fuera planteado en el Capítulo 6 de Técnicas Digitales I, se deberá implementar una lógica de control que será la encargada de seleccionar un solo dispositivo de salida activado (parlante) por vez.

En el caso de las microcomputadoras, esa selección recibe el nombre de decodificación (pues habitualmente se realiza con circuitos integrados decodificadores tipo 74LS138 ó 74LS139), pudiéndose emplear eventualmente para configuraciones no convencionales memorias Prom. o Pals.

6.4.1 Decodificación completa.

Se dice que la decodificación es completa cuando existe una relación biunívoca entre cada posición de memoria y cada dirección. Esto quiere decir que **a)** al aparecer una dirección sobre el bus correspondiente, se activará una sola celda de memoria (condición indispensable para evitar la contención del bus) y **b)** cada celda de memoria se activará con una única dirección.

Para que exista decodificación completa es necesario que TODAS las líneas del bus de direcciones ingresen en la selección de una celda de memoria (algunas directamente a la memoria y otras a la lógica de decodificación).

En la Fig. 9 se verifica la biunivocidad de forma que cada combinación de líneas del bus de direcciones habilitará un único dispositivo y que las líneas que no llegan al decodificador lo hacen a los componentes del sistema, seleccionando o una posición de memoria o una interfaz de E/S dentro de dicho dispositivo.

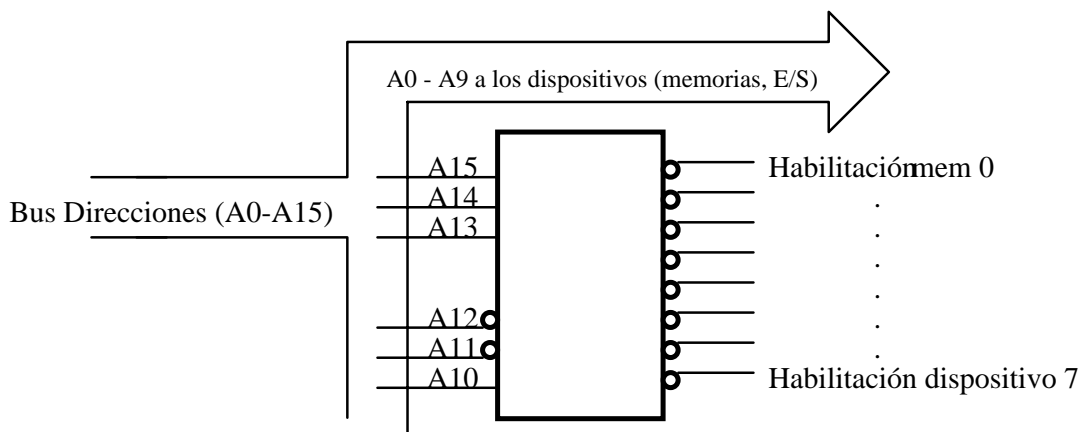


Fig. 9. Decodificación completa

6.4.2 Decodificación incompleta o parcial.

Existen circunstancias en las que por simplicidad o para minimizar la cantidad de componentes no se hacen llegar todas las líneas del bus de direcciones al decodificador o a los dispositivos de memoria o interfaz de entrada / salida.

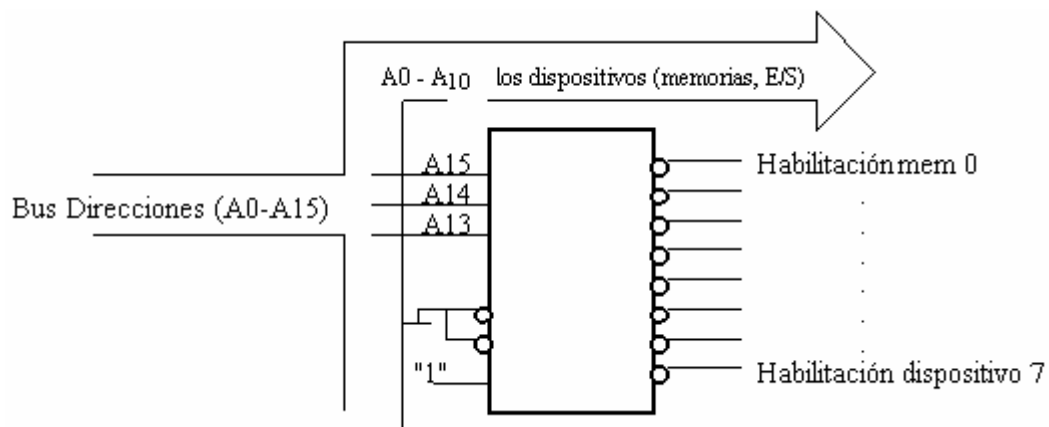


Fig. 10. Ejemplo de decodificación parcial

Supóngase un esquema circuital como el de la Fig. 10. Aquí vemos que las líneas A15, A14 y A13 llegan al decodificador y las líneas A10 hasta A0 llegan a los dispositivos. Nótese que A12 y A11 no están conectadas a ningún componente.

Representemos la combinación de bits del bus de direcciones que permiten seleccionar el circuito integrado memoria 0.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	X	X	0	0	0	0	0	0	0	0	0	0	0
0	0	0	X	X	1	1	1	1	1	1	1	1	1	1	1

Como A12 y A11 no están conectados a ningún dispositivo, cualquiera sea su valor, si se da la combinación restante de bits, se seleccionará la memoria 0.

Esto implica que para el límite inferior anterior, (A15-A13 y A10-A0 = 0) tendremos cuatro combinaciones de los bits no considerados, a saber:

Combinación	Bus de direcciones	Hexa
Combinación 1	0000000000000000	0000H
Combinación 2	0000100000000000	0800H
Combinación 3	0001000000000000	1000H
Combinación 4	0001100000000000	1800H

Esto quiere decir que con cualquiera de esas cuatro combinaciones de bits sobre el bus de direcciones seleccionarán la misma posición de memoria. En otros términos, se dispondrán cuatro imágenes o espejos de memoria.

7. Principales características de las computadoras.

7.1 Longitud de la palabra de instrucción.

La longitud de la palabra de instrucción, da una idea de la dimensión (expresada en bits) de la instrucción (o código de operación) típico de una computadora. En una primera aproximación permite cuantificar la potencia del sistema.

Normalmente la cantidad de bits de la palabra de instrucción coincide con la extensión de la barra de datos, pero no es imprescindible que ello sea así (posteriormente mencionaremos algunas excepciones).

De acuerdo con la cantidad de bits de la palabra de instrucción, podemos realizar una primera clasificación de los procesadores:

7.1.1 Palabra de instrucción de 4 bits:

Algunos pequeños microcontroladores con esta extensión fueron utilizados para sencillas aplicaciones masivas donde importaba de sobremanera el costo para competir con equivalentes electromecánicos.

7.1.2 Palabra de instrucción de 8 bits:

Fue la longitud de la palabra de instrucción de los primeros microprocesadores de uso masivo y actualmente sigue teniendo vigencia. Los principales exponentes de esta clasificación son los microprocesadores:

- Intel 8080.
- Intel 8085.
- Motorola 6800.
- Motorola 6802.
- Motorola 6803.
- Motorola 6809.
- Zilog Z-80.
- Rockwell 6502.

No existen actualmente mini-computadoras con esta extensión de palabra.

7.1.3 Palabra de instrucción de 16 bits:

En 1980 comenzaron a aparecer microprocesadores con esta extensión de instrucción. Son la llamada tercera generación de microprocesadores y además de incrementar la extensión de la palabra de instrucción, han sido provistos de instrucciones más poderosas y mayor velocidad, lo que representa un real salto frente a sus predecesores.

Los microprocesadores más significativos de este grupo son:

- Intel 8086/8088 [7]
- Intel 80188/80186.
- Intel 80286.
- Intel 80386 SX [8]
- Motorola 68000.
- Motorola 68008.
- National 16032.
- Zilog Z8000.

Algunas antiguas mini-computadoras (como la PDP-11) poseyeron esta extensión en su palabra de instrucción.

7.1.4 Palabra de instrucción de 32 bits.

Han desplazado a los microprocesadores de 16 bits casi por completo, ya que presentan las soluciones más populares y (paradójicamente) más económicas pues debido a su configuración interna en la que existen múltiples subdivisiones que trabajan simultáneamente, las memorias externas son más lentas y por ende más económicas.

Algunos de dichos microprocesadores son:

- iAPX 432 (que en 3 circuitos integrados proveía el núcleo del sistema operativo de una computadora IBM 370, actualmente fuera de producción).
- Intel 80386 (DX y SL)
- Intel 80486 (DX, DX2, DX4 y SX).
- Motorola 68020 y 68030.

⁷ Diferencias entre el 8088/188 y el 8086/186: El 8088 y el 8086 son dos procesadores internamente casi idénticos. Comparten el mismo repertorio de intrucciones (es decir que un programa escrito por uno puede ser ejecutado por otro, aunque empleando distinto tiempo de ejecución), pero difieren en su barra de datos externa. El 8088 opera internamente con una barra de datos de 16 bits pero externamente presenta 8 bits, mientras que el 8086 dispone de 16 bits tanto en la barra de datos interna como externa. Análogamente esa diferencia es compartida por los 80188 y 80186.

⁸ El 80386SX es una versión del 80386 que dispone de un bus de datos externo de 16 bits. Permite ejecutar todas las aplicaciones que requieran un 80386 (por ejemplo Windows en el modo mejorado) con el bajo costo de conexionado con el mundo exterior de los microprocesadores de 16 bits.

- National 32032.
- Intel Pentium, Pentium PRO, Pentium MMX y Pentium II
- Motorola Power PC

7.1.5 Palabra de instrucción de 64 bits.

A partir del Pentium Pro de Intel, del K6 de AMD y del G3 de Motorola.

7.1.6 Resumen.

Dentro de la familia de procesadores de Intel, podemos presentar la siguiente Tabla como resumen de las características de acceso del bus de datos.

Microprocesador	Instrucción (bits)
8088	8
80188	8
8086	16
80186	16
80286	16
80386sx	16
80386dx	32
80486	32
80586 class/ Pentium (Pro)	64

Tabla 1. Comparación del ancho del bus de datos de los procesadores Intel

7.2 Extensión de la barra de direcciones.

Originalmente, esta extensión no era un parámetro de significación trascendente en la evaluación del comportamiento de un microprocesador, pues sólo permite conocer la cantidad de posiciones de memoria que puede llegar a direccionar el procesador. La unidad de direccionamiento es el **"K"** (Kilo que aquí representa 2^{10} es decir 1024 posiciones), el **Mega** ($1K * 1K$), el **Giga** ($1K * 1M$) y el **Tera** ($1K * 1G$). En la actualidad, con programas cada vez más extensos debido a los mayores requerimientos que se le exigen al software, la capacidad de direccionamiento de un procesador comienza a tener mayor importancia que la que tenía en los orígenes.

La totalidad de los microprocesadores de 8 bits en su palabra de instrucción, poseen 16 bits en la barra de direcciones, es decir que pueden llegar a direccionar hasta 65536 (64 K) posiciones de memoria.

Otros procesadores (8086, 8088) poseen 20 líneas de direcciones, pudiendo de tal manera direccionar 1 Mega posiciones de memoria.

Los microprocesadores del tipo 80286 tienen 24 líneas de direcciones pudiendo direccionar los 16 Megabytes.

Los microprocesadores de 32 bits de palabra de instrucción disponen también 32 bits de direccionamiento, pudiendo manejar 4 Gigabytes ($2^{32} = 4 \cdot 10^{10} \cdot 10^{10} \cdot 10^{10} = 4$ kilo Mega bytes).

Microprocesador	Líneas en el AB	Capacidad de direc.
8088	20	1,048,576
8086	20	1,048,576
80188	20	1,048,576
80186	20	1,048,576
80286	24	16,777,216
80386sx	24	16,777,216
80386dx	32	4,294,976,296
80486	32	4,294,976,296
80586 / Pentium (Pro)	32	4,294,976,296

Tabla 2. Capacidad de direccionamiento de los procesadores de Intel

7.3 Cantidad de direcciones en la palabra de instrucción.

Debido a la importancia que reviste, procederemos a un análisis más detallado de la palabra de instrucción.

7.3.1 Palabra de 4 direcciones.

La palabra de instrucción genérica consta de cinco partes (o campos) a saber:

- Código de operación, que indica que operación se realizará (suma, resta, comparación, etc.).
- Dirección donde se halla el primer operando.
- Dirección donde se halla el segundo operando.
- Dirección donde se almacenará el resultado de la operación.
- Dirección donde se deberá ir a busca la próxima instrucción (código de operación).
-

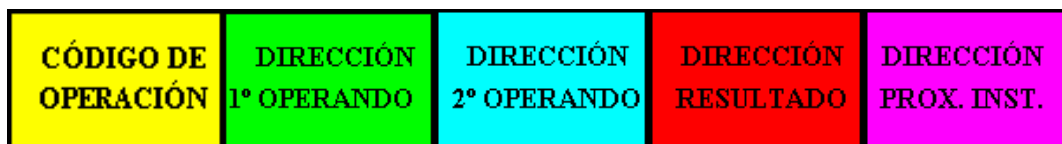


Fig. 11. Formato de Palabra de 4 direcciones.

Este formato es indudablemente el más completo, pues absolutamente toda la información que necesita el sistema de micro cómputo se halla en la palabra de instrucción.

En la mayoría de los microprocesadores usuales, la especificación de una dirección requiere por lo menos 16 bits y el código de operación como mínimo 8 bits, por lo que la palabra de instrucción de 4 direcciones requeriría 72 bits, lo cual es un despropósito para una microcomputadora actual, pues sería costoso disponer de memorias de 72 bits de ancho de palabra o bien lento si se accedieran los 72 bits en fracciones sucesivas.

Como ilustración de este problema, imaginemos una autopista de 9 carriles (tamaño de la instrucción) que se angostan en un puente de un solo carril (bus de datos). De nada han servido los nueve carriles, pues la duración del trayecto dependerá fuertemente del tiempo que se tarde en pasar por ese angostamiento de la carretera.

Por todo ello que deberá buscarse algún método para tratar de disminuir la extensión de la palabra de instrucción.

7.3.2 Palabra de 3 direcciones. Registro Contador de Programa.

Como los programas tienen fundamentalmente una estructura secuencial y ejecutan una instrucción tras otra, es frecuente que dichas instrucciones se encuentren en direcciones consecutivas. Si se acepta esta suposición, el quinto campo de la palabra de cuatro direcciones ya no es necesario y se tiene la palabra de tres direcciones.



Fig. 12. Formato de Palabra de 3 direcciones.

En la Fig. 12 vemos el formato de la palabra de 3 direcciones donde ha desaparecido la dirección de la próxima instrucción. El precio que se debe pagar por esa simplificación es agregar internamente al procesador (normalmente en su unidad de control), un contador incrementable que contenga la dirección de la próxima instrucción a ser ejecutada. Según la arquitectura de que se trate, ese contador puede recibir distintos nombres. El que adoptaremos en este momento será el de Contador De Programa ⁹(PC = Program Counter) y es un registro del procesador auto incrementado o bien modificable por el programa (por medio de saltos o bien de llamados a subrutinas).

7.3.3 Palabra de Dos Direcciones.

La economía de memoria de programa se lleva aún más lejos, si aceptamos que la dirección del resultado coincida con la dirección del primer operando, de modo que al concluir la operación el resultado de la misma se halle en lugar del primer operando. En forma simbólica:

$$(A) \leftarrow (A) + (B)$$

En donde aparecen por primera vez los paréntesis encerrando una variable. Ellos indican "Contenido de:". De esa forma la expresión anterior debe leerse: "El contenido de la posición de memoria (o registro) A sumado al contenido de la posición de memoria (o registro) B es almacenado en la posición de memoria A".

⁹ Debe hacerse una pequeña diferenciación entre el Contador de Programa (PC) y Puntero a Instrucción (IP). El Contador de Programa proviene de los viejos microprocesadores en los que, siguiendo los lineamientos aquí descritos, el registro interno apunta al próximo código de operación u operando a ser **ejecutado**, mientras que el concepto de Puntero a Instrucción se deriva de los nuevos microprocesadores, en los que existe dentro del propio microprocesador una pequeña memoria (habitualmente llamada cola) en la que se ha copiado anticipadamente un tramo de la memoria de programa aprovechando los tiempos ociosos del microprocesador, para disponer inmediatamente de un código de operación sin necesidad de tener que ir a buscarlo a memoria. En estos casos, el IP apunta a la próxima posición de memoria a ser **leída**.

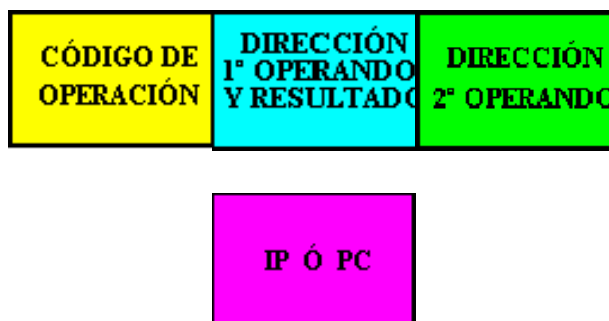


Fig. 13. Formato de Palabra de 2 direcciones.

Por otro lado existen instrucciones en las cuales la tercera dirección es innecesaria. En efecto, si se desea transferir el contenido de A a la posición de memoria B:

$$(B) \leftarrow (A)$$

Una palabra de dos direcciones basta perfectamente para especificar esta instrucción, y una tercera dirección es innecesaria.

7.3.4 Palabra de Una Dirección. Acumulador.

Para simplificar aún más la arquitectura del sistema, podemos retirar otro campo de operando de la palabra de instrucción. En dichas circunstancias el primer operando y el resultado residen en un registro interno de la unidad aritmético-lógica de la CPU que recibe el nombre de **Acumulador**.

Esta computadora de una dirección, requiere solamente la dirección del segundo operando, admitiendo implícitamente, que el primer operando ha sido cargado en el acumulador por medio de una instrucción previa, y que el resultado de la operación quedará en el mismo acumulador.

Si bien se ha reducido la extensión de la palabra de instrucción se hace más lenta la operación pues debe realizarse en varias instrucciones. Primero cargar el acumulador y luego realizar la operación propiamente dicha. La simplificación de la arquitectura, hace más lento el proceso de ejecución del programa.

Estas definiciones deben interpretarse conceptualmente y no físicamente. Por ejemplo en el familiar caso de las computadoras personales (PCs), la CPU será el microprocesador (y no todo el gabinete mal llamado CPU), la memoria es inconfundible y como periféricos tendremos: indudablemente la impresora, las unidades de disco (a pesar que se hallen dentro del gabinete de la PC), los modems (ya sean internos o externos), el teclado y el monitor.



Fig. 14. Formato de Palabra de 1 dirección.

7.3.4.1 Ejemplo para una computadora de una dirección

Se desea sumar el contenido de la posición de memoria **R** con el contenido de la posición de memoria **S** y que el resultado quede en la posición de memoria **T**.

Esta operación requeriría una sola instrucción en una computadora de 3 direcciones, pero con la computadora de una dirección debe fraccionarse la operación como sigue:

- Cargar el operando **R** al acumulador:

$$(\text{Acc}) \leftarrow (\text{R})$$

- Realizar la operación suma.

$$(\text{Acc}) \leftarrow (\text{Acc.}) + (\text{S})$$

- Guardar el resultado en T.

$$(\text{T}) \leftarrow (\text{Acc.})$$

Aparentemente, la computadora de una dirección es complicada de programar y lenta, pero para operaciones reiterativas, como puede ser la suma de los elementos de una tabla de datos, el acumulador debe ser cargado una sola vez y el resultado parcial de una operación sirve como primer operando de la siguiente.

8. Modos de direccionamiento fundamentales

8.1 Planteo

Aún en la máquina de una dirección es necesario proveer, de alguna forma, el segundo operando. Esa forma de entregar esa información recibe el nombre de modo de direccionamiento.

Cada microprocesador tiene sus propias formas de proveer ese segundo operando. Sin embargo existen tres modos de direccionamiento comunes a casi todos los microprocesadores. Son los que llamamos modos de direccionamiento básicos.

8.2 Direccionamiento Directo

En este caso se da la **Dirección** en la que se encuentra el segundo operando, de forma que el microprocesador deberá leer los dos bytes de la dirección y posteriormente colocarlos sobre el bus de direcciones y recién entonces leer de memoria ese segundo operando^[10].

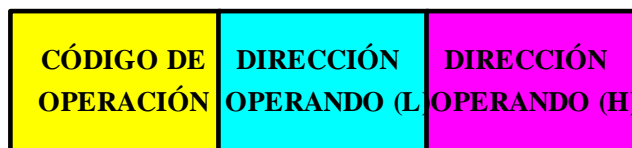


Fig. 15. Ejemplo de instrucción con direccionamiento directo

8.3 Direccionamiento inmediato.

Aquí se entrega directamente el segundo operando (no la dirección del mismo). ¿Cómo hace el microprocesador para saber si lo que se halla después del código de operación es el operando o la dirección en la que el mismo se halla?. Obviamente, aún para la misma instrucción, son distintos códigos de operación para cada modo de direccionamiento.



Fig. 16. Ejemplo de instrucción con direccionamiento inmediato

Aquí el segundo operando se halla en el byte inmediato posterior al código de operación. Una vez que el procesador obtuvo dicho código y lo decodificó, sabe que debe ir a buscar un solo byte y que él es el segundo operando, con el cual debe realizar la operación indicada por el código de operación de la instrucción.

8.4 Direccionamiento inherente (o implícito).

¹⁰ Se presenta aquí por primera vez algo que será común a todos los microprocesadores producidos por Intel, que es que al almacenar en memoria una dirección, primero se guardará el byte menos significativo de la dirección y luego el más significativo. Este mecanismo es totalmente transparente para el usuario, pues el microprocesador guardará automáticamente en memoria las direcciones en este formato y también las rescatará automáticamente cuando se lo solicita.

Existen algunas operaciones en las que no es necesario dar otro operando y toda la información se halla en el código de operación (por ejemplo complementar el acumulador o mover datos entre registros internos al microprocesador). En tales circunstancias el procesador al decodificar la instrucción, sabe que solo debe ejecutarla (sin buscar ninguna información adicional en memoria) y pasar a leer el próximo código de operación.



Fig. 17. Ejemplo de instrucción con direccionamiento implícito

8.5 Otros modos de direccionamiento.

Existen otros modos de direccionamiento, propios de cada microprocesador. En esta sección presentaremos alguno de los modos de direccionamiento que, si bien no están presentes en todos los microprocesadores, son casi universales.

8.5.1 Registro.

Realiza la transferencia de información entre registros internos del microprocesador. No requiere (salvo el código de operación) ninguna información adicional.

Una de las principales innovaciones que se han ido produciendo en las distintas generaciones de microprocesadores ha sido la aparición de numerosos registros dentro de los mismos. El concepto de estos registros se basa en lo visto en el capítulo 10 de Técnicas Digitales I y que se sintetiza en ser posiciones de memoria dentro del microprocesador. La ventaja que conllevan, es disponer de un dato dentro del propio microprocesador, lo cual disminuye el tiempo de ejecución de una instrucción, ya que no es necesario tener que buscar un operando desde la memoria, pues esta acción requiere habilitar los buffers y esperar a que la memoria conteste el requerimiento, lo que implica un tiempo considerable.

8.5.2 Registro Indirecto.

Se emplea algún registro del microprocesador como puntero para mover datos. Por ejemplo supongamos tener un registro R1 dentro del microprocesador, el mismo será empleado para apuntar a memoria y el contenido de la posición de memoria así apuntada será llevado al acumulador. Por medio de una simbología propia de cada microprocesador, ello se indicará como:

(Acc) ← ((R1))

Aquí se presenta por primera vez el doble paréntesis. Debe interpretarse como que el contenido de R1 es usado como puntero a memoria cuyo contenido es transferido al acumulador. Con otra simbología:

MOV A, @R1 o bien

MOV A, [R1]

8.5.3 Relativo.

Consiste en tomar como referencia el estado actual de un registro interno al microprocesador, por ejemplo el contador de programa IP, e indicar en complemento a dos, la distancia respecto de este punto a la que se halla la posición de memoria referenciada. Por ejemplo, la instrucción de un microprocesador hipotético:

SJMP ALLA ; 8 bits

implica saltar a una posición de memoria identificada por la etiqueta ALLA. La instrucción, aparte del código de operación dispondrá de un byte en el que en complemento a dos (-128 a +127) se indicará la distancia a la que se halla el destino, referido al estado actual del contador de programa.

Este modo de direccionamiento se suele emplear para disminuir la extensión del programa, pues la instrucción resultante contiene menos bytes que la que resultaría de indicar la totalidad de la dirección (16 bits por lo menos).

8.5.4 Indexado.

Existirá un registro puntero a memoria, llamado habitualmente índice o index, que se emplea como referencia para direccionar una variable. Por ejemplo supongamos querer acceder a una tabla en memoria para realizar una conversión entre códigos^[1]. El valor del código que se desea convertir se cargará en un registro que llamaremos índice X. Si el origen de la tabla de conversión se halla indicado por la posición de memoria caracterizada con la etiqueta TAB_CONV, la instrucción de un hipotético microprocesador:

```
MOV  A, [TAB_CONV + X ]
```

Traerá al acumulador **A**, el contenido de la posición de memoria que, tomando como base **TAB_CONV**, se halla desplazada el valor **X**. Con ello si en la tabla se encontraban los valores de conversión entre códigos, en el acumulador **A**, quedará el resultado en el código destino.

9. Ciclos de instrucción, máquina y reloj.

9.1 Ciclo de instrucción.

Llamaremos ciclo de instrucción al conjunto de eventos que deben realizarse para ejecutar una instrucción. Desde el punto de vista de la ejecución, la instrucción es la menor unidad indivisible de un programa.. Esto quiere decir que no puede interrumpirse una instrucción luego que comenzó a ejecutarse. Ahora bien, para el análisis del funcionamiento de un microprocesador fraccionaremos la operación del mismo presentando **los ciclos de máquina y reloj**.

9.2 Ciclo de máquina.

Se define así a toda operación perfectamente definible realizada por el microprocesador. Por ejemplo, serán ciclos de máquina la búsqueda de código de operación, la lectura de memoria o dispositivos de entrada, la escritura de memoria o dispositivos de salida y algunas más que veremos en el capítulo 3.

9.3 Ciclo de reloj (o estado).

Las microcomputadoras, como todo sistema secuencial, se rige por un reloj (o clock) obtenido a partir de un oscilador a cristal. Ese reloj maneja todas las transacciones que realiza el microprocesador (y es el que fija los tiempos para activar los multiplexores internos, habilitar buffers, la carga de los registros, etc.).

El período de este reloj es la base de tiempo sobre la cual se medirán todas las operaciones del microprocesador. Un ciclo de máquina durará por lo menos uno y en general varios ciclos de reloj (depende del microprocesador).

En la Fig. 18 vemos el diagrama temporal correspondiente a la escritura de una posición de memoria empleando direccionamiento directo. En el primer ciclo de máquina se buscará el código de operación, luego en dos ciclos de máquina consecutivos se leerán los dos bytes correspondientes a la dirección. Finalmente en el cuarto y último ciclo de máquina se procederá a escribir la posición de memoria.

¹¹ Ver Técnicas Digitales I, Capítulo I.

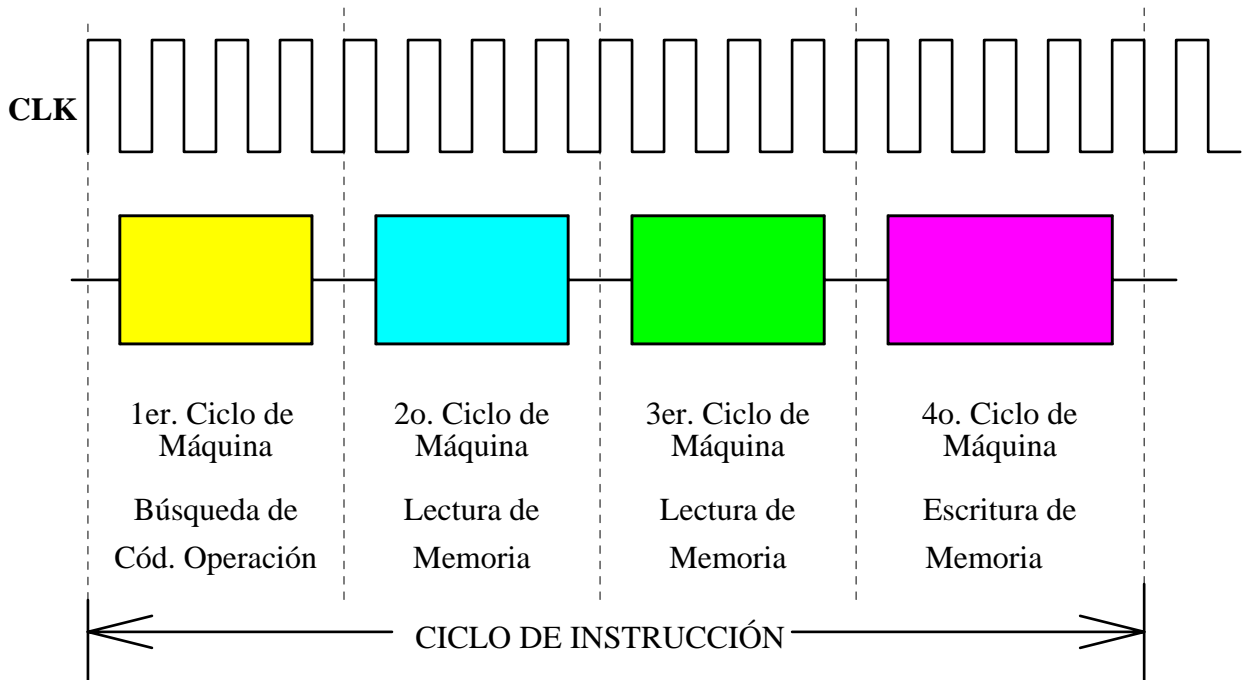


Fig. 18. Ciclos de instrucción, máquina y reloj

10. Análisis de la ejecución de un tramo de programa

10.1 Planteo

Ninguno de los fabricantes de microprocesadores comerciales ha realizado una descripción detallada de la operación interna de un microprocesador. Por tal motivo, en este punto analizaremos las operaciones que realiza un microprocesador hipotético para ejecutar un tramo de programa.

Supongamos disponer de un microprocesador didáctico con la arquitectura de la Fig. 19.

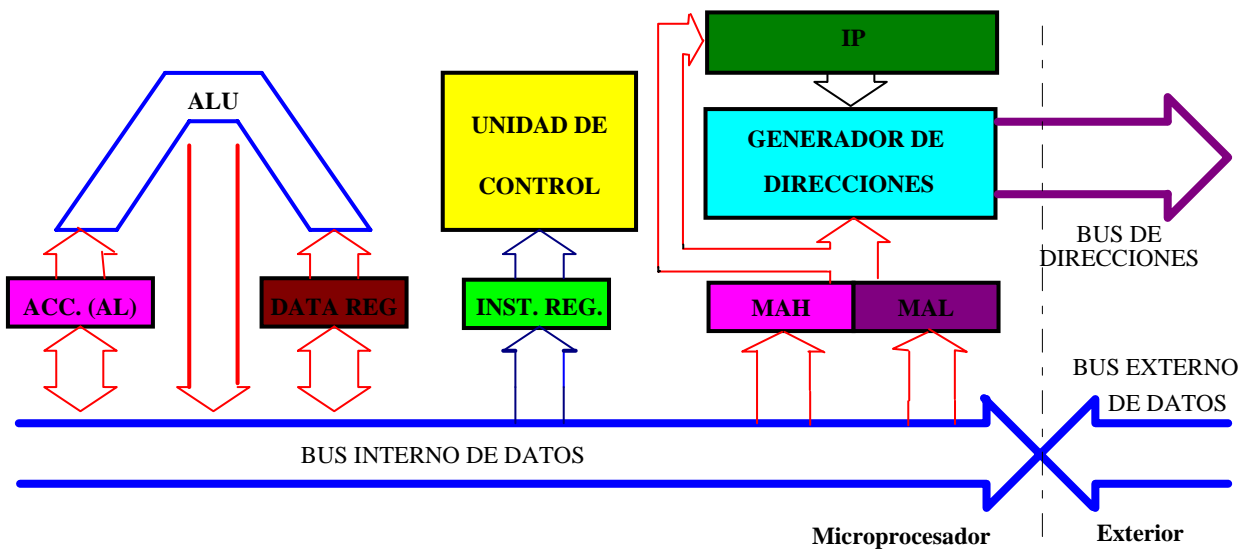


Fig. 19. Arquitectura hipotética de un microprocesador de 8 bits

10.2 Compiladores y Assembler.

El programa de una microcomputadora consiste en un conjunto de palabras binarias que se hallan en la memoria de programa y que le informarán al microprocesador (en forma adecuadamente codificada) de las operaciones internas y externas que deberá realizar para cumplimentar un dado objetivo.

De lo anterior debe resultar definitivamente claro que el procesador solo entiende palabras binarias y que las mismas son difícilmente entendibles y manejables para los seres humanos.

Por tal motivo se decidió crear un lenguaje entendible por los humanos y que por medio de nemónicos o abreviaturas provea una idea lo más próxima posible de la operación que se desea realizar.

Por ejemplo

ADD BL,DH

indicará que se desea sumar (ADD en inglés) el contenido de dos registros internos del procesador, denominados **BL** y **DH**¹².

Esta presentación clara para los humanos, es inentendible para las microcomputadoras (recordemos una vez más que sólo interpretan palabras binarias). Hasta hace algunos años, esos nemónicos se traducían manualmente a palabras binarias. Hoy en día existen programas traductores de nemónicos a palabras binarias y reciben el nombre de compiladores.

Cada microprocesador tendrá su propio repertorio de nemónicos (habitualmente son registrados y protegidos contra copia por los fabricantes) y distintos compiladores (C, Pascal, algunos Basic, etc.) con nemónicos y sintaxis propios para representar distinto tipo de operaciones.

Existe un compilador que se denomina Ensamblador o Assembler en el que se corresponden biunívocamente instrucciones con nemónicos. Esto quiere decir que a cada instrucción le corresponde un solo nemónico y viceversa a diferencia de los *lenguajes de alto nivel* en los que una sentencia (un "renglón") del programa corresponde a un grupo de instrucciones. Normalmente cuanto más poderoso sea el lenguaje de alto nivel, cada sentencia representará mayor cantidad de instrucciones.

Los ingenieros en electrónica habitualmente escribirán programas en lenguaje ensamblador (esencialmente en este curso) y en C, ya que este es de los lenguajes de alto nivel más próximos al hardware, es decir que el programador puede tener un control más acabado del circuito.

Habitualmente una instrucción o sentencia de un lenguaje de alto nivel realiza acciones más complejas que las que realiza una instrucción en lenguaje ensamblador.

10.3 Ejemplo de movimiento de datos en un programa

El microprocesador es la unidad central de proceso de una Microcomputadora. Es un componente que cumple una función sencilla, lee un código de operación (palabra binaria que le indica que debe hacer), lo interpreta y lo ejecuta. Al culminar la ejecución, deberá buscar otro código de operación y así sucesivamente en forma cíclica y permanente.

En el ejemplo que veremos a continuación, aparece por primera vez el símbolo ; para indicar los comentarios. El compilador entenderá que lo que se halla a la derecha de este símbolo es un comentario y lo trasladará sin traducirlo.

Para tratar de entender el funcionamiento del microprocesador dentro de una microcomputadora, supongamos tener el siguiente programa:

¹² La convención planteada por Intel, consiste en que el operando que se halla a la izquierda es quien retendrá el resultado de la operación, es decir que la suma de BL con DH producirá un resultado que quedará en BL

Posición de Memoria	Contenido	Nemónico	Comentario
2040H	A0	MOV AL,(1234H)	;Traer a AL el contenido de la posición de memoria 1234H
	34		;Parte baja de la dirección
	12		;Parte alta de la dirección
2043H	04	ADD AL,0F6H	;Sumar 0F6H al registro AL
	F6		;Operando a sumar
2045H	72	JC ALFA	.Si hubo acarreo saltar a la etiqueta
	F5		; Parte baja de la dirección ALFA
	23		; Parte alta de la dirección ALFA
2048H	F6	NOT AL	;Complemento a 1 de AL
.....			
23F5	ALFA:		

Analizaremos el comportamiento (a nivel de buses, movimiento de datos, actuación de registros, etc.) hipotético del microprocesador para realizar este tramo de programa.

Admitiremos que el Puntero a Instrucciones IP, ejecutando tramos previos de programa, llegó a valer 2040H, de forma de apuntar al primer byte de este programa.

10.3.1 Primera Instrucción. MOV AL,(1234H)

10.3.1.1 Primer Ciclo de Máquina. Búsqueda de Código de operación.

a) El microprocesador culminó de ejecutar una instrucción previa y sabe, por su secuenciador interno que debe buscar un código de operación. Entonces, coloca sobre el bus de direcciones el valor actual de IP, de forma de apuntar a la dirección 2040H, en donde se encuentra el primer código de operación.

b) Selecciona esa posición de memoria para realizar una lectura (transferencia de datos desde la memoria hacia el microprocesador). La memoria vuelca su contenido sobre el bus de datos y llega al registro Instruction Register (IR) del microprocesador (obvio pues se está buscando un código de operación). El contenido de este registro temporario y del cual el usuario desconoce su existencia, (lo que suele definirse como que es "transparente" para el mismo) se mantendrá activando a la Unidad de Control hasta que se culmine la presente instrucción y deba reemplazarse el corriente código de operación por el de la próxima instrucción.

c) El microprocesador incrementa automáticamente el valor de IP pues es valor anterior ya fue utilizado (IP=2041H).

Simbólicamente:

- (IP) → **Bus Direcciones** → (2040H) → **Bus Datos** → A0H → (IR)
- (IP) ← (IP + 1) (2041H)

10.3.1.2 Segundo Ciclo de Máquina. Lectura de memoria.

Al decodificar el código de operación que se halla en el IR, el microprocesador determina que se trata de un direccionamiento directo, por lo que debe buscar dos bytes que corresponderán a la dirección de donde se debe leer el dato que irá al registro AL.

Aquí aparece otro de los registros temporarios transparentes para el usuario. Toda dirección consta de 16 bits. Por el bus de datos pueden llegar de la memoria de programa, 8 bits por vez. Por ello será necesario contar con un registro que almacene una mitad de la dirección mientras se procede a leer la otra mitad. Ese registro es el "Registro de direccionamiento a memoria" o **MAR** (Memory Address Register).

a) El microprocesador efectúa un primer ciclo de lectura de memoria para lograr el byte menos significativo de la dirección (convención Intel). El contenido de la posición de memoria 2041H se almacenará en un registro temporario e invisible para el usuario llamado **MAL** (Memory Address Low) hasta obtener la otra parte de la dirección y con ella, realizar la carga del registro AL

b) El microprocesador incrementa automáticamente el valor de IP pues el valor anterior ya fue utilizado (IP=2042H). Simbólicamente:

- $(IP) \rightarrow \text{Bus Direcciones} \rightarrow (2041H) \rightarrow \text{Bus Datos} \rightarrow 34H \rightarrow (MAL)$
- $(IP) \leftarrow (IP + 1) (2042H)$

10.3.1.3 Tercer Ciclo de Máquina. Lectura de Memoria.

Según lo indicado por el código de operación, el microprocesador busca el segundo byte de la dirección. Lo almacena en el registro temporario **MAH**

El microprocesador incrementa automáticamente el valor de IP pues es valor anterior ya fue utilizado (IP=2043H). Simbólicamente:

- $(IP) \rightarrow \text{Bus Direcciones} \rightarrow (2042H) \rightarrow \text{Bus Datos} \rightarrow 12H \rightarrow (MAH)$
- $(IP) \leftarrow (IP + 1) (2043H)$

10.3.1.4 Cuarto Ciclo de máquina. Lectura de Memoria.

Ya se halla la totalidad de la dirección del operando en el registro interno **MAR** (MAH + MAL). Se pone esa dirección sobre el bus de direcciones y se lee esa posición de memoria. El contenido se lleva al registro AL del microprocesador. Supóngase que la posición de memoria 1234H había sido cargada previamente con **10H**.

- $(MAR) \rightarrow \text{Bus Direcciones} \rightarrow (1234H) \rightarrow \text{Bus Datos} \rightarrow 10H \rightarrow (AL)$

Obsérvese que no se ha incrementado el IP pues no fue utilizado en este ciclo de máquina.

10.3.2 Segunda Instrucción	ADD AL,0F6H.
-----------------------------------	---------------------

10.3.2.1 Primer Ciclo de Máquina. Búsqueda de Código de operación.

a) El microprocesador culminó de ejecutar una instrucción previa y sabe, por su secuenciador interno que debe buscar un código de operación. Entonces, coloca sobre el bus de direcciones el valor actual de IP, de forma de apuntar a la dirección 2043H, en donde se encuentra el próximo código de operación.

b) Selecciona esa posición de memoria para realizar una lectura (transferencia de datos desde la memoria hacia el microprocesador). La memoria vuelca su contenido sobre el bus de datos y llega al registro IR del microprocesador.

c) El microprocesador incrementa automáticamente el valor de IP pues es valor anterior ya fue utilizado (IP=2044H).

Simbólicamente:

- $(IP) \rightarrow \text{Bus Direcciones} \rightarrow (2043H) \rightarrow \text{Bus Datos} \rightarrow 04H \rightarrow (IR)$
- $(IP) \leftarrow (IP + 1) (2044H)$

10.3.2.2 Segundo Ciclo de Máquina. Lectura de memoria.

a) Al decodificar el código de operación que se halla en el IR, el microprocesador determina que se trata de un direccionamiento inmediato, por lo que debe buscar el byte que corresponderá al dato que debe adicionarse al contenido anteriormente cargado en AL.

b) El microprocesador efectúa un ciclo de lectura de memoria para lograr dicho dato y lo lleva al registro temporario adyacente a la ALU, para sumarlo a AL. El resultado será $10H + F6H = 106H$. Quedará en el registro AL, $06H$ con un acarreo en un indicador para una eventual segunda etapa de la suma.

c) El microprocesador incrementa automáticamente el valor de IP pues es valor anterior ya fue utilizado (IP=2045H). Simbólicamente:

- $(IP) \rightarrow \text{Bus Direcciones} \rightarrow (2044H) \rightarrow \text{Bus Datos} \rightarrow F6H \rightarrow (\text{Data Register})$
- $(IP) \leftarrow (IP + 1) \quad (2045H)$
- $(AL) \leftarrow (AL) + (\text{Data Register}) \quad AL = 06 \quad CY = 1$

10.3.3 Tercera Instrucción	JC	23F5H.
-----------------------------------	-----------	---------------

10.3.3.1 Primer Ciclo de Máquina. Búsqueda de Código de operación.

a) El microprocesador culminó de ejecutar una instrucción previa y sabe, por su secuenciador interno que debe buscar un código de operación. Entonces, coloca sobre el bus de direcciones el valor actual de IP, de forma de apuntar a la dirección 2045H, en donde se encuentra el próximo código de operación.

b) Selecciona esa posición de memoria para realizar una lectura (transferencia de datos desde la memoria hacia el microprocesador). La memoria vuelca su contenido sobre el bus de datos y llega al registro IR del microprocesador (obvio pues se esta buscando un código de operación).

c) El microprocesador incrementa automáticamente el valor de IP pues es valor anterior ya fue utilizado (IP=2046H).

Simbólicamente:

- $(IP) \rightarrow \text{Bus Direcciones} \rightarrow (2045H) \rightarrow \text{Bus Datos} \rightarrow 72H \rightarrow (IR)$
- $(IP) \leftarrow (IP + 1) \quad (2046H)$

10.3.3.2 Segundo Ciclo de Máquina. Lectura de memoria.

a) Al decodificar el código de operación que se halla en el IR, el microprocesador determina que se trata de un salto a otra dirección, condicionado a la existencia de acarreo. Se verifica que en la última operación aritmético-lógica existió ese acarreo, y se deben buscar los dos bytes que corresponderán a la dirección a donde se deberá saltar.

b) El microprocesador efectúa un primer ciclo de lectura de memoria para lograr el byte menos significativo de la dirección (convención Intel). El contenido de la posición de memoria 2046H se almacenará en un registro temporario e invisible para el usuario llamado **MAL** (Memory Address Low) hasta que al obtener la otra parte de la dirección y realizar el salto.

c) El microprocesador incrementa automáticamente el valor de IP [¹³] pues es valor anterior ya fue utilizado (IP=2042H). Simbólicamente:

- $(IP) \rightarrow \text{Bus Direcciones} \rightarrow (2046H) \rightarrow \text{Bus Datos} \rightarrow F5H \rightarrow (MAL)$
- $(IP) \leftarrow (IP + 1) \quad (2047H)$

10.3.3.3 Tercer Ciclo de Máquina. Lectura de Memoria.

a) Según lo indicado por el código de operación, el microprocesador busca el segundo byte de la dirección. Lo almacena en el registro temporario **MAH**

¹³ Obsérvese que en caso de no darse la condición de salto (si el acarreo fuera nulo), el microprocesador no leería la dirección de salto (ya que no la va a emplear), sino que para ganar tiempo, luego de leer el código de operación y verificar el no cumplimiento de la condición de salto, incrementará en dos el puntero a instrucción para apuntar al próximo código de operación.

b) El microprocesador **no** incrementa el valor de IP pues el código de operación le ha indicado que se cambiará el IP a fin de producir un salto, por lo que no tendría sentido incrementar IP para inmediatamente cambiarlo.

Simbólicamente:

- (IP) → **Bus Direcciones** → (2047H) → **Bus Datos** → 23H → (MAH)
- (IP) ← (MAR) (23F5H)

Este proceso de búsqueda de código de operación, interpretación y ejecución es eterno y es la mayor simplificación que puede realizarse referida a la actuación del microprocesador.

La actuación de un microprocesador consiste en leer códigos de operación de una memoria de programa, decodificarlos y ejecutarlos.

10.3.4 Ejercicio.

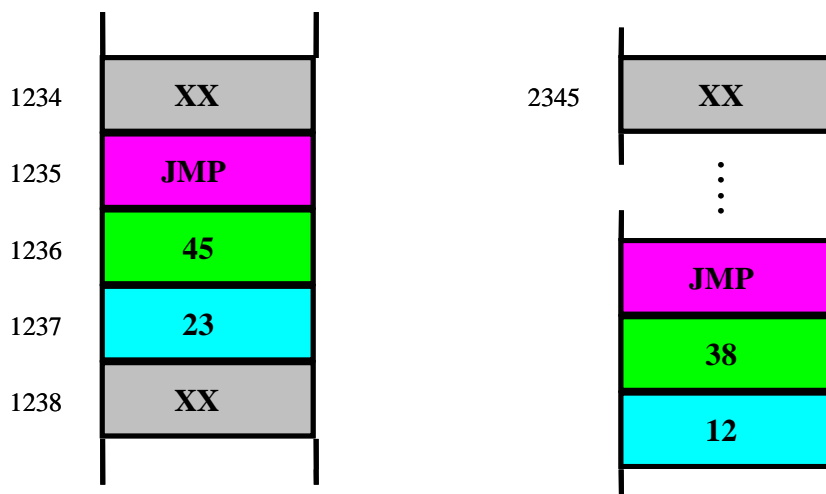
Repetir el ejemplo anterior, pero suponiendo que el contenido de la posición de memoria 1234H es 01 H.

11. Subrutinas.

11.1 Planteo.

Algunas veces el mismo tramo de programa es ejecutado en varias oportunidades. Según el esquema planteado hasta el presente, cada vez que dicho tramo de programa es requerido deberá insertarse, con el correspondiente mal uso de la memoria, ya que se reiterará en varias oportunidades el mismo conjunto de operaciones.

Una primera solución consistiría en colocar un salto incondicional a la dirección de inicio de dicho sub-programa. Al finalizar el mismo se colocará otro salto a la dirección siguiente a la que fuera llamado.



Nota: XX significa contenido de posición de memoria sin importancia.

Fig. 20_ Uso de un subprograma por medio de saltos.

En la Fig. 20 vemos un ejemplo de lo anteriormente dicho.

En la posición de memoria 2345 se ha implementado un subprograma utilitario, por ejemplo un programa que realiza la suma de operandos de más de un byte.

Dicha operación es requerida dentro del programa principal, en la dirección 1236, por lo que se coloca un salto a la dirección de comienzo del subprograma de suma.

Al finalizar dicho programa utilitario se coloca un salto a la dirección 1238 que es la siguiente a la dirección de donde fue requerido el sub-programa.

Para que este subprograma pudiera ser utilizado desde distintos lugares del programa principal, debería poder modificarse la dirección del salto que se encuentra en el fin del sub-programa de suma, puesto que si no, regresaría siempre al mismo lugar del programa principal.

11.2 Pila (Stack) y Puntero a la Pila (Stack Pointer).

El mecanismo creado para poder utilizar un mismo subprograma desde distintos lugares del programa principal, proveyendo la adecuada dirección de retorno al mismo es el siguiente:

Se reserva una zona de memoria de lectura y escritura a fin de que en ella puedan almacenarse las direcciones de retorno. Dicha zona de memoria recibe el nombre de pila (por su forma de operación). Obviamente dicha zona de memoria no deberá ser escrita por el programa, pues destruiría la información allí almacenada.

Esta memoria es del tipo LIFO [14] y tiene asociada un puntero llamado puntero a la pila (que es un registro interno al procesador y que debe ser cargado con su valor inicial por medio de la ejecución de una instrucción), que apuntará a la última dirección ocupada (procesadores de la familia Intel y Zilog) o bien a la primera dirección disponible (procesadores de la familia Motorola). En todos los casos luego de guardar una dirección (o el contenido de algún registro, según se verá posteriormente). El puntero a la pila se autodecrementa o autoincrementa adecuadamente en cada instrucción que emplee la pila.

En la Fig. 21 se observa el caso anterior con el agregado de un salto "inteligente" que sabe a donde volver pues en la pila se ha guardado la dirección de retorno. Ese salto inteligente, recibe el nombre de **RET** o retorno y extrae de la pila la dirección de retorno con dos lecturas consecutivas. En la primera toma el contenido de la dirección apuntada por ese registro **SP** y lo transfiere a la parte baja del IP, decrementa SP y repite la misma operación para lograr la parte alta de IP.

Es evidente que al encontrarse con la instrucción de retorno, el registro SP deberá estar apuntando a la dirección de retorno. Es decir que si se emplean instrucciones que utilizan la pila (y que se analizarán a continuación) deberán usarse equilibradamente, es decir que todo aquello que se almacena en la pila deberá extraerse antes de realizar el retorno.

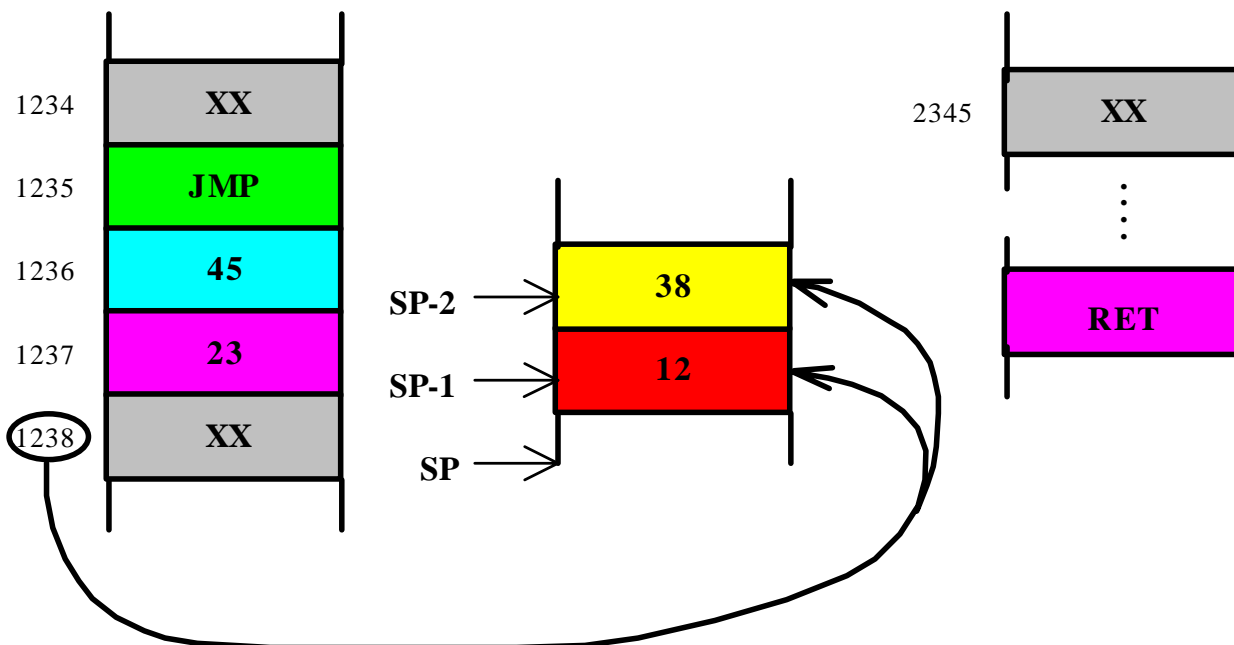


Fig. 21. Uso del puntero a la pila para almacenar la dirección de retorno.

¹⁴ Como se verá posteriormente AX es un registro interno al procesador de 16 bits y que esta subdividido en dos partes de 8 bits, llamadas AH la más significativa y AL la menos significativa.

11.3 PUSH y POP.

La pila puede emplearse para almacenar temporariamente información por medio de la instrucción PUSH. En efecto, PUSH AX¹⁵, por ejemplo almacena los 16 bits del registro AX en la pila (siempre en el denominado "orden INTEL", es decir primero parte menos significativa y luego la más significativa) para poder usar temporariamente AX con otros fines. Posteriormente se restaurará el valor de AX con la instrucción recíproca POP AX.

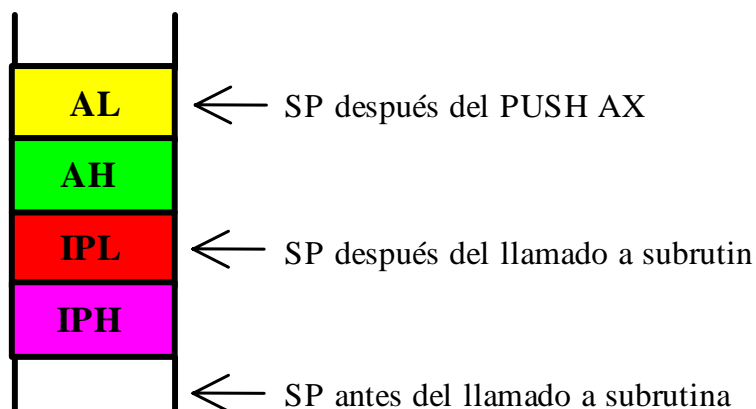


Fig. 22. Uso de la pila

Supongamos que la primera instrucción de una subrutina sea PUSH AX, luego de ejecutarse la misma, la pila quedará como se ve en la Fig. 22. Antes de intentar retornar al programa principal, se deberán realizar la cantidad de POPs necesarios para dejar el SP apuntando al lugar donde se halla la dirección de retorno.

Con las instrucciones de PUSH, se decrementa en dos el registro SP, mientras que con la POP, se lo incrementa en igual valor.

Debe destacarse que la pila posee sus datos más "viejos" en las direcciones mayores y los más "nuevos" en las menores direcciones, es decir que crece en el sentido de las bajas direcciones.

12. Bibliografía recomendada.

- Fundamentos de Diseño lógico y Computadoras. Mano y Kime. Prentice Hall 1998.
- Introducción a los microprocesadores. Szklanny y Martínez del Pezzo. Arbó 1979.
- Microprocesadores y Lógica Programada. Short. Gustavo Gili.
- The art of assembly Language. www.electron.frba.utn.edu.ar.
- Los Microprocesadores Intel. Barry B. Brey. Prentice Hall 1995
- The 8086/8088 family. Design, programming and Interfacing. Uffenbeck. Prentice Hall 1987..
- The 8086 book. Rector - Alexy. Mc Graw Hill 1980.
- Introducción al microprocesador 8086/8088. Morgan - Waite. Mc Graw Hill. 1982.
- The 8088 and 8086 Microprocessors (second edition). Triebel y Singh. Prentice Hall 1997.

¹⁵ Como se verá posteriormente, AX es un registro interno al microprocesador de 16 bits y que esta subdividida en dos registros de 8 bits cada uno, llamados AH (high) y AL (low).

13. Cuestionario.

13.1 Evolución del diseño lógico.

- a) ¿Qué entiende por lógica cableada y lógica programada?
- b) ¿Quién le da la personalidad a un equipo cuando se resuelve un problema usando lógica cableada?
- c) ¿Quién le da la personalidad a un equipo cuando se resuelve un problema usando lógica programada?
- d) ¿Qué entiende por programa?

13.2 Clasificación de las computadoras.

- a) Caracterice las computadoras hogareñas, las computadoras personales, las mini computadoras, las súper microcomputadoras y las mainframes.
- b) ¿Cuál es la diferencia entre una red de computadoras y computadoras multiusuario?.
- c) ¿Cómo se desarrolló el concepto de computadoras RISC y cuáles son sus ventajas?.
- d) ¿Qué es un sistema dedicado y uno de propósito general?.

13.3 Órganos funcionales de una computadora bajo la arquitectura Von Neumann.

13.3.1 Diagrama en bloques de una computadora.

- a) Dibuje el diagrama básico de una computadora, indicando cuáles son sus partes constitutivas.
- b) ¿Cuál es la función de la Unidad central de proceso (CPU)?.
- c) ¿Cómo se subdivide la misma?. ¿Cuál es la función de cada una de sus partes?
- d) ¿Cuál es la función de la unidad de memoria?. ¿Cómo se puede subdividir? ¿Qué almacena cada una de esas subdivisiones? (¿Dónde almacena una variable y donde una constante?). ¿Por qué?
- e) ¿Qué función cumplen los dispositivos de entrada / salida?. ¿Son los llamados periféricos?. ¿Cómo clasifica a una impresora o a un monitor de vídeo, como dispositivo de E/S o como un periférico? ¿Por qué?
- f) ¿Qué es un microprocesador? ¿Qué es una microcomputadora? ¿Qué funciones cumple un microprocesador dentro de una microcomputadora?
- g) ¿Qué entiende por hardware, por software y por firmware? Cite ejemplos.
- h) ¿Qué entiende por computadora dedicada?

13.3.2 Computadoras según arquitectura Von Neumann y Harvard.

- a) ¿Cuáles son las características y diferencias de las arquitecturas Von Neumann y Harvard?.
- b) ¿Qué ejemplos conoce de cada una de estas arquitecturas?. Ventajas y desventajas.

13.3.3 Buses.

- a) Analice la conexión de dos salidas TTL estándar en paralelo. Extraiga conclusiones. ¿Qué entiende por lógica tri-state?. ¿En qué se diferencia una compuerta tri-state de una compuerta TTL o CMOS tradicional?.
- b) Dibuje el esquema de la salida de una compuerta tri-state y analice su operación. ¿Se pueden conectar en paralelo las entradas de compuertas TTL estándar con las entradas de compuertas tri-state?. ¿Y las salidas?. ¿Qué función cumple la lógica de control?.
- c) ¿Cuáles son los buses de un microprocesador?. ¿Cuál es la función de cada uno de ellos?. ¿Qué significa que un bus sea unidireccional y bidireccional?.
- d) ¿Qué entiende por direccionar a un dispositivo? ¿Quién realiza ese direccionamiento?

e) ¿Es la capacidad de direccionamiento un índice efectivo del poderío del microprocesador?. Justifique su respuesta.

f) ¿Qué cantidad de líneas dispone el 8088 en cada uno de sus buses? ¿Cuál es la capacidad de direccionamiento del 8088?. ¿Cómo la calcula?.

g) ¿Qué función cumple un buffer?. ¿Cómo actúa?. ¿Cómo es su impedancia de salida?. ¿Y de que tipo?.

h) La resistencia de entrada de los dispositivos MOS y CMOS es prácticamente infinita. ¿Para qué se utilizan entonces los buffers en los buses de un microprocesador?

i) ¿Qué tipo de buffers se utilizan en el bus de direcciones?

j) ¿Qué entiende por bus de datos, DB ("Data Bus")?. ¿Qué características tiene?. ¿Cuál es la longitud del bus de datos externo de un 8088?. ¿Cuál es la longitud del bus de datos interno de un 8088?.

k) Averigüe el ancho del bus de datos de un Procesador Pentium III.

l) ¿Qué significa que un microprocesador es de 8 bits?. ¿Y de 8/16 bits?

m) ¿Qué tipo de buffers se utilizan en el bus de datos?.

n) ¿Qué ventaja tiene un bus de datos más ancho? ¿Es éste un índice efectivo del poderío de un microprocesador?.

o) ¿Para qué sirve el bus de control?. ¿Cómo viaja la información a través del mismo?.

p) ¿Cuántas cargas TTL permite manejar normalmente una línea de salida de un microprocesador?.

13.4 Decodificación.

a) ¿Qué entiende por decodificación?

b) ¿Cuándo se produce contención del bus?. ¿Qué ocurre en este caso?

c) ¿Típicamente que dispositivos se utilizan para implementar la lógica de control?.

d) ¿Qué entiende por decodificación parcial o unívoca?

e) ¿Qué entiende por decodificación total, completa o biunívoca?. ¿Cómo se logra?

f) ¿Qué ventajas y desventajas tiene cada una de estas técnicas de decodificación?

g) ¿Qué significa que una memoria es de 8 Kbytes? Cuantos bits tiene?. ¿Cómo se expresa la longitud de la palabra?. ¿Qué longitud tiene la palabra en este caso?.

h) ¿Cuántas líneas del bus de direcciones llegan a una memoria de 8 Kbytes? Imagine la implementación de una decodificación completa para un banco de memorias de 8 Kbytes.

i) Para los decodificadores empleados normalmente, ¿las salidas son activas bajas o altas?.

j) ¿Qué entiende por imágenes o espejos de una posición de memoria? ¿Cuándo se producen?

k) Calcule el número de imágenes cuando hay "n" líneas no usadas en la decodificación.

l) Se debe diseñar una computadora dedicada que por razones de costo utilice la menor área de circuito impreso posible ¿qué tipo de decodificación elegiría?. Justifique su respuesta.

m) ¿Qué tipo de decodificación elegiría si se necesitan extremar las medidas de seguridad a fin de evitar eventuales contenciones del bus o selecciones espúreas?. Justifique su respuesta.

n) Indique como puede generar una decodificación reconfigurable apta para una PC, que le permita generar 8 "Chip Select" para escritura y 8 "Chip Select" para lectura.

o) Que señales conviene que lleguen a la habilitación de las puertas de entrada / salida para asegurarse que se está accediendo a las mismas solo cuando se quiere leer o escribir un dato?

13.5 Máquinas de 4, 3, 2 y 1 dirección.

a) Suponga disponer de un microprocesador similar al 8088 y que responda al formato de palabra de 4 direcciones. ¿Qué cantidad de bits tendrá la palabra de instrucción?. ¿Qué entiende por código de operación?.

b) ¿Qué inconvenientes le encuentra a una computadora con palabra de 4 direcciones?

c) Describa la función de cada uno de los campos de la palabra anterior. Indique el camino a seguir para llegar a convertir al microprocesador hipotético anterior en un 8088 real (pasarle a palabra de 3, direcciones, de 2, etc.), indicando como se llama en realidad cada uno de los registros que vayan apareciendo.

d) ¿Qué es un registro?. Para almacenar variables, ¿Cuál es la ventaja de emplear registros frente a posiciones de memoria?.

e) ¿Qué son el Instruction Pointer (IP) y el acumulador? ¿Qué longitud tiene en el 8088?

f) ¿Qué diferencias presentan el Contador de Programa y el Puntero a Instrucción?.

g) ¿Con qué formato de palabra trabajan actualmente los microprocesadores? ¿Cómo esta compuesta esta palabra?. ¿Qué longitud tiene?

h) Los microprocesadores actuales, ¿con qué formato de palabra trabajan?

i) Para una computadora de una dirección, realice el diagrama de flujo del programa que permita sumar dos operandos que se hallan en memoria. ¿Cuál es el precio que se paga por llegar a la palabra de una dirección?

13.6 Modos de direccionamiento fundamentales

a) Suponiendo una instrucción, por ejemplo la suma, ¿cómo la afecta el modo de direccionamiento?. ¿Qué le provee?.

b) ¿Qué entiende por modos de direccionamiento?.

c) ¿Cuáles son los modos de direccionamiento que conoce?.

d) Escriba un ejemplo de direccionamiento directo o pleno. Explique el significado de los símbolos que usa. Describa detalladamente la operación realizada.

e) Explique la forma en que almacena Intel las direcciones en memoria. Ejemplifíquelo suponiendo que se desea almacenar la dirección 5678H a partir de la posición de memoria 1234H. Analice tanto el proceso de almacenamiento de la dirección como el proceso de lectura.

f) Escriba un ejemplo de direccionamiento indirecto con registro. Explique el significado de los símbolos que usa. Describa detalladamente la operación realizada.

g) Escriba un ejemplo de direccionamiento inmediato. Explique el significado de los símbolos que usa. Describa detalladamente la operación realizada.

h) Escriba un ejemplo de direccionamiento inherente. Explique el significado de los símbolos que usa. Describa detalladamente la operación realizada.

i) Describa detalladamente el accionar del microprocesador (información sobre los buses, alteración de registros, direccionamientos a memoria, etc.) ante las siguientes instrucciones:

MOV SI,1234

ADD AL,[SI]

cuya representación en memoria es:

BEH 34H 12H 02H 04H

j) ¿Qué entiende por nemónicos?.

k) ¿Qué es un compilador, el Assembler y el vinculador (o linker)?

13.7 Ciclos de instrucción, Máquina y Reloj.

a) Defina claramente los conceptos de ciclos de reloj, máquina e instrucción.

b) Si se esta ejecutando una instrucción y se produce algún evento externo asincrónico e impredecible, ¿puede interrumpirse el ciclo de instrucción y reanudarse posteriormente?. ¿Y el ciclo de máquina en ejecución?.

c) ¿Qué tipos de ciclos de máquina puede definir en este momento?.

d) Un ciclo de máquina, ¿puede abarcar varios ciclos de instrucción?. Un ciclo de instrucción, ¿puede abarcar varios ciclos de máquina?.

e) Un ciclo de máquina ¿cuántos ciclos de reloj puede abarcar como mínimo?.

13.8 Ejemplo de movimiento de datos en un programa

a) Analice el mismo ejemplo de la teoría, pero suponiendo que el contenido de la posición de memoria 1234H es 05H.

b) Analice el movimiento de datos del siguiente programa:

```
MOV  AL,0F6H
ADD   AL,[1234H]
```

a) Analice el movimiento de datos para el siguiente programa.

```
MOV  SI,1234H
MOV  AL,[SI]
```

Suponiendo que SI es un registro interno del microprocesador que se emplea como puntero a memoria.

13.9 Subrutinas y Uso de la Pila.

a) Suponga que el puntero a la pila se halla apuntando a la posición 4000H. Indique como se encontrará el puntero a la pila y la pila si al ejecutar la instrucción en la dirección 3333H, se llama a una subrutina que se halla en la dirección C761H y posteriormente se impulsa a la pila AX cuyo contenido es 4567H.

b) Para el ejemplo anterior, analizar detalladamente la operación del puntero a la pila y el contenido de la pila al realizar las operaciones POP AX y RET.

c) Suponga que el microprocesador tiene dos registros internos de 16 bits cada uno de ellos denominados AX y BS. Analizar la evolución de la pila y del puntero a la pila si al ejecutar la instrucción en la dirección 3333H, se llama a una subrutina que se halla en la dirección C761H y dentro de la subrutina se ejecuta el siguiente programa:

```
.....
PUSH AX
PUSH BX
.....
POP   AX
POP   BX
RET
```

Indicar el contenido de los registros AX y BX al regresar al programa principal.