



NOMBRE DEL ALUMNO: EDDI DAVID AGUILAR

NOMBRE DE LA MATERIA: MICROPROCESADORES

NOMBRE DEL PROFESOR: JUAN JOSE OJEDA TRUJILLO

***NOMBRE DE LA LICENCIATURA: INGENIERÍA EN SISTEMAS
COMPUTACIONALES***

CUATRIMESTRE: 7

Unidad I

Estructura de bits y bytes

1.1.- Sistemas numéricos decimal, binario y hexadecimal.

La base decimal tiene 10 dígitos representativos que van del 0 al 9 con lo que puede representarse cualquier valor, en la base hexadecimal se tiene 16 dígitos que van del 0 al 9 y de la letra A hasta la F las que representan los números del 10 al 15. La conversión entre números binarios y hexadecimales, es sencilla notando que cada cuatro binarios equivalen a un hexadecimal y cada hexadecimal equivale a cuatro (4) binarios, esta agrupación de cuatro dígitos binarios se debe comenzar desde la posición menos significativa desde el punto separador de enteros y fracciones a la izquierda o a la derecha.

Un bit es la unidad más pequeña de información, pero en su forma más simple un bit representa un dígito binario, es decir, un “uno” o un “cero”, físicamente los bits requieren de un espacio o celda donde almacenar una cantidad de energía definida en dos estados estables que representen los dígitos binarios, esta energía es llamada voltaje, que es la energía necesaria para mover una carga eléctrica (electrón) de un punto a otro.

1.2.- Formatos de datos.

Bus de datos Es el encargado de llevar datos e instrucciones hacia y desde el microprocesador, es un bus bidireccional en donde los datos varían en tamaño desde 8 a 64 bits, por el bus de datos las instrucciones y los datos son transferidos al microprocesador y los resultados de una operación son enviados desde el microprocesador. **Bus de direcciones** Contiene la información digital que envía el microprocesador a la memoria y demás dispositivos direccionales del sistema para seleccionar una posición de memoria, una unidad de entrada/salida o un registro particular de la misma, la cantidad de líneas del bus de direcciones determina la capacidad máxima de la memoria que puede direccionar el sistema basado en microprocesador. Los primeros microprocesadores utilizados en sistemas de cómputo tenían 16 líneas de direcciones con lo que podían direccionar hasta $2^{16} = 2^{10} * 2^6 = 1024 * 64 = 65536 = 64KB$.

I.3.- Funcionamiento interno de una PC.

Los sistemas de cómputo basados en microprocesadores se componen de tres bloques fundamentales, la Unidad Central de Procesamiento – CPU, Dispositivos de memoria y Puertos de Entrada / Salida, de igual forma el microprocesador está conformado por varios bloques, entre ellos están la Unidad Aritmético – lógica (ALU), una Unidad de Control (UC) y un bloque o matriz de registros. La unidad de entrada conformada dispositivos encargados de recibir la información del mundo exterior, estos dispositivos llamados de entrada / salida o I/O (Input/Output), la información es llevada hacia la unidad de memoria para ser procesada posteriormente. La información luego es llevada desde la unidad central de proceso o CPU, hacia circuitos o periféricos utilizando el bus de datos. La unidad de memoria se encarga de almacenar los datos y los programas que operan sobre esos datos. La CPU obtiene las instrucciones y los datos colocando una dirección en el bus de direcciones para posteriormente ser transferidos a través del bus de datos cuando la CPU lo solicite.

I.4.- La evolución de los microprocesadores.

En los comienzos de la implementación de circuitos lógicos el diseñador debía poseer amplios conocimientos tanto en lógica digital como en los componentes y dispositivos que debía acoplar, esto presentaba inconvenientes, la llamada lógica cableada no permitía hacer modificaciones sin afectar la construcción física del circuito, en 1970 esto cambia con la aparición del microprocesador, con lo que aparece lo que se denomina Lógica programable, la cual permite modificar el comportamiento lógico digital del circuito sin tener que cambiar su configuración física.

Partiendo de las técnicas digitales consolidadas en los años sesenta, se comienza a desarrollar y profundizar en el estudio de las técnicas y desarrollo de aplicaciones para los microprocesadores junto con la programación en lenguaje máquina y la programación en assembler. Con esta breve introducción es hora de iniciar una exploración por los momentos históricos que han sido claves para la invención, evolución y desarrollo del microprocesador. Primeros dispositivos para realizar cálculos El dispositivo más reconocido es el Abaco, su origen se pierde en el tiempo algunos estiman cerca de 3000 años A.C otros entre 600 y 500 A.C en Egipto o China, su efectividad ha superado el pasar de los años.

Dispositivos mecánicos Basados en los logaritmos, surge las primeras máquinas analógicas de cálculo derivadas de prototipos construidos por Edmund Gunter (1581-1626), matemático y astrónomo inglés y William Oughtred (1574-1660).

Microprocesadores de Intel y otros fabricantes Intel, Con el desarrollo del 8080 de 8 bits lanzado en abril de 1974 y posteriormente el 8085 binariamente compatible con el 8080 pero que tenía menos exigencia de hardware sigue una serie de avances y desarrollo de nuevos microprocesadores con características que mejoran el desempeño con respecto a sus predecesores. En los 80"s aparecen los microprocesadores de 16 bits UNIVERSIDAD DEL SURESTE 25 bits más potentes con la incursión al mercado del 8086 de Intel en 1978, utilizado por IBM para la fabricación de la gama de IBM PC serie PS/2. Intel prosigue el desarrollo del 8088 para IBM XT, el 80186, el 80286 utilizado para IBM AT, el 80386, 80486, Pentium que fue un nombre comercial para evitar la duplicación de su nomenclatura por parte de AMD y Cyrex, más adelante sigue con el desarrollo y producción de Pentium II, Pentium III, Pentium IV, hasta los actuales microprocesadores multinúcleo.

1.5.- Arquitectura del microprocesador 80x86.

Los procesadores de 16 bits fueron una nueva generación de microprocesadores desarrollados para reemplazar o completar a las microcomputadoras de 8 bits de los años setenta, que fueron las que comenzaron la revolución de las microcomputadoras. El 8086 fue diseñado para trabajar con lenguajes de alto nivel, disponiendo de un soporte hardware con el que los programas escritos en dichos lenguajes ocupan un pequeño espacio de código y pueden ejecutarse a gran velocidad. Esta concepción, orientada al uso de compiladores, se materializa en un conjunto de facilidades y recursos, y en unas instrucciones entre las que cabe destacar las que permiten efectuar operaciones aritméticas de multiplicar y dividir, con y sin signo; las que manejan cadenas de caracteres, etc. En su momento, el 8086 junto con el 8088 fueron los microprocesadores más empleados dentro de su categoría, especialmente desde que IBM los adoptó para la construcción de su computadora personal. Muchos fabricantes de microordenadores utilizaron esta familia microcomputadora para fabricar equipos de tipo profesional. Hoy en día, la utilización del 8086 es más reducida, quedando principalmente orientado a la enseñanza, como base de los microprocesadores de la última generación.

I.6.- Estructura de un programa ejecutable cargado en memoria.

Las direcciones completas de las instrucciones y de las posiciones de la pila se forman sumando el contenido de los registros IP y SP con el segmento de código (CS) y el segmento de pila (SS) respectivamente. La dirección de un dato puede formarse mediante la suma de los contenidos de los registros BX ó BP, los contenidos de SI ó DI, y un desplazamiento. El resultado de este cálculo se denomina dirección efectiva (EA) ó desplazamiento de segmento. La dirección definitiva del dato, sin embargo, se determina mediante la EA y el registro de segmento apropiado, el segmento de datos (DS), el segmento extra (ES) ó el segmento de pila (SS). El uso de los diferentes segmentos significa que hay áreas de trabajo separadas para el programa, la pila y los datos. Cada área de trabajo tiene un tamaño máximo de 64K bytes y un mínimo de 0. Dado que hay cuatro registros de segmento, uno de programa (CS), uno de pila (SS), uno de datos (DS) y uno extra (ES) el área de trabajo puede llegar hasta 256K.

I.7.- Arreglo de registros internos.

IP (índice de programa) almacena el desplazamiento dentro del segmento de código. Este registro junto al registro CS apunta a la dirección de la próxima instrucción. No puede ser usado como operando en operaciones aritmético/lógicas.

SI (índice de origen) almacena el desplazamiento del operando de origen en memoria en algunos tipos de operaciones (operaciones con operandos en memoria).

DI (índice de destino) almacena el desplazamiento del operando de destino en memoria en algunos tipos de operaciones (operaciones con operandos en memoria).

SP (índice de pila) almacena el desplazamiento dentro del segmento de pila, y apunta al último elemento introducido en la pila. Se usa conjuntamente con el registro SS.

BP (índice de base) se usa para almacenar desplazamiento en los distintos segmentos. Por defecto es el segmento de la pila Registro de estado El registro de estado contiene una serie de banderas que indican distintas situaciones en las que se encuentra el procesador

OF (desbordamiento) es el principal indicador de error producido durante las operaciones con signo. Vale 1 cuando:

- La suma de dos números con igual signo o la resta de dos números con signo opuesto producen un resultado que no se puede guardar (más de 16 bits).
- El bit más significativo (el signo) del operando ha cambiado durante una operación de desplazamiento aritmético.
- El resultado de una operación de división produce un cociente que no cabe en el registro de resultado.

1.8.- Operación en modo real.

El modo real (también llamado modo de dirección real en los manuales de Intel) es un modo de operación del 8086 y posteriores CPUs compatibles de la arquitectura x86. El modo real está caracterizado por 20 bits de espacio de direcciones segmentado (significando que solamente se puede direccionar 1 MB de memoria), acceso directo del software a las rutinas del BIOS y el hardware periférico, y no tiene conceptos de protección de memoria o multitarea a nivel de hardware. Todos los CPUs x86 de las series del 80286 y posteriores empiezan en modo real al encenderse el computador; los CPUs 80186 y anteriores tenían solo un modo operacional, que era equivalente al modo real en chips posteriores.

La arquitectura 286 introdujo el modo protegido, permitiendo, entre otras cosas, la protección de la memoria a nivel de hardware. Sin embargo, usar estas nuevas características requirió instrucciones de software adicionales no necesarias previamente. Puesto que una especificación de diseño primaria de los microprocesadores x86 es que sean completamente compatibles hacia atrás con el software escrito para todos los chips x86 antes de ellos, el chip 286 fue hecho para iniciarse en 'modo real', es decir, en un modo que tenía apagadas las nuevas características de protección de memoria, de modo que pudieran ejecutar sistemas operativos escritos para microprocesadores más viejos. Al día de hoy,

incluso los más recientes CPUs x86 se inician en modo real al encenderse, y pueden ejecutar el software escrito para cualquier chip anterior. Los sistemas operativos DOS (MS-DOS, DR-DOS, etc.) trabajan en modo real. Las primeras versiones de Microsoft Windows, que eran esencialmente un shell de interface gráfica de usuario corriendo sobre el DOS, no eran realmente un sistema operativo por sí mismas, corrían en modo real, hasta Windows 3.0, que podía ejecutarse tanto en modo real como en modo protegido. Windows 3.0 podía ejecutarse de hecho en dos "sabores" de modo protegido - el "modo estándar", que corría usando modo protegido, y el "modo mejorado 386", que además usaba direccionamiento de 32 bits y por lo tanto no corría en un 286 (que a pesar de tener modo protegido, seguía siendo un chip de 16 bits; los registros de 32 bits fueron introducidos en la serie 80386). Con Windows 3.1 se retiró el soporte para el modo real, y fue el primer ambiente operativo de uso masivo que requirió por lo menos un procesador 80286 (no contando con el Windows 2.0 que no fue un producto masivo). Casi todos los sistemas operativos modernos x86 (Linux, Windows 95 y posteriores, OS/2, etc.) cambian el CPU a modo protegido o a modo largo en el arranque.

1.9.- Operación en modo protegido.

El modo protegido es un modo operacional de los CPUs compatibles x86 de la serie 80286 y posteriores. El modo protegido tiene un número de nuevas características diseñadas para mejorar la multitarea y la estabilidad del sistema, tales como la protección de memoria, y soporte de hardware para memoria virtual como también la conmutación de tarea. A veces es abreviado como p-mode y también llamado Protected Virtual Address Mode (Modo de Dirección Virtual Protegido) en el manual de referencia de programador del iAPX 286 de Intel, (Nota, iAPX 286 es solo otro nombre para el Intel 80286). En el 80386 y procesadores de 32 bits posteriores se agregó un sistema de paginación que es parte del modo protegido. UNIVERSIDAD DEL SURESTE 36 La mayoría de los sistemas operativos x86 modernos se ejecutan en modo protegido, incluyendo GNU/Linux, FreeBSD, OpenBSD, NetBSD, Mac OS X y Microsoft Windows 3.0 y posteriores. (Windows 3.0 también se ejecutaba en el modo real para la compatibilidad con las aplicaciones de Windows 2.x). El otro modo operacional principal del 286 y CPUs posteriores es el modo real, un modo de compatibilidad hacia atrás que desactiva las características propias del modo protegido, diseñado para permitir al software viejo ejecutarse en los chips más recientes. Como una especificación de diseño, todos los CPUs x86 comienzan en modo real en el momento de

carga (boot time) para asegurar compatibilidad hacia atrás con los sistemas operativos heredados, excepto el Intel 80376 diseñado para aplicaciones en sistemas embebidos. Estos procesadores deben ser cambiados a modo protegido por un programa antes de que esté disponible cualquier característica de este modo. En computadores modernos, este cambio es generalmente una de las primeras tareas realizadas por el sistema operativo en el tiempo de carga. Mientras que la multitarea en sistemas ejecutándose en modo real es ciertamente posible mediada por software, las características de protección de memoria del modo protegido previenen que un programa erróneo pueda dañar la memoria "propia" de otra tarea o del núcleo del sistema operativo. El modo protegido también tiene soporte de hardware para interrumpir un programa en ejecución y cambiar el contexto de ejecución a otro, permitiendo pre-emptive multitasking.

Unidad 2

Modos de direccionamiento

La forma en que se especifica un operando se denomina modo de direccionamiento, es decir, es un conjunto de reglas que especifican la localización (posición) de un dato usado durante la ejecución de una instrucción. El 8086 tiene 25 modos de direccionamiento o reglas para localizar un operando de una instrucción. Los modos de direccionamiento más frecuentes son los que calculan la dirección del operando mediante la suma de la dirección base de un registro segmento, multiplicado por 16 y el valor de un desplazamiento.

La gran variedad de direccionamientos proviene de las muchas formas en que se puede determinar el desplazamiento. El tipo de direccionamiento se determina en función de los operandos de la instrucción. La instrucción MOV realiza transferencia de datos desde un operando origen a un operando destino (se verá más con más detalle en los siguientes apartados). Su formato es el siguiente: MOV destino, origen

2.1.- Direccionamiento por registros. Cuando ambos operando son un registro. Ejemplo: MOV AX,BX ;transfiere el contenido de BX en AX

2.2.- Direccionamiento inmediato. Cuando el operando origen es una constante. Ejemplo: MOV AX,500 ;carga en AX el valor 500.

2.3.- Direccionamiento directo. Cuando el operando es una dirección de memoria. Ésta puede ser especificada con su valor entre [], o bien mediante una variable definida previamente (cómo definir etiquetas se verá más adelante).

2.4.- Direccionamiento base más índice. Cuando el operando esta en memoria en una posición apuntada por el registro BX o BP al que se le añade un determinado desplazamiento Ejemplo: MOV AX, [BP] + 2 ; almacena en AX el contenido de la posición de memoria que resulte de sumar 2 al contenido de BP (dentro de segmento de pila). Equivalente a MOV AX, [BP + 2]

2.5.- Direccionamiento relativo. Cuando la dirección del operando se obtiene de la suma de un registro base (BP o BX), de un Índice (DI, SI) y opcionalmente un desplazamiento. Ejemplo: MOV AX, TABLA[BX][DI] ; almacena en AX el contenido de la posición de memoria apuntada por la suma de TABLA, el contenido de BX y el contenido de DI.

2.5.1.- Instrucciones para transferencia de datos. Las instrucciones de transferencia de datos copian datos de un sitio a otro y son: MOV, CHG, XLAT, LEA, LDS, LES, LAHF, SAHF, PUSH, PUSHF, POP, POPF. MOV realiza la transferencia de datos del operando de origen al destino. Como ya hemos visto en la parte de los modos de direccionamiento, MOV admite todos los tipos de direccionamiento. Ambos operandos deben ser del mismo tamaño y no pueden estar ambos en memoria. MOV reg, reg ; reg es cualquier registro. MOV mem, reg ; mem indica una posición de memoria MOV reg, mem MOV mem, dato ; dato es una constante MOV reg, dato MOV seg-reg, mem ;seg-reg es un registro de segmento MOV seg-reg, reg MOV mem, seg-reg MOV reg, seg-reg XCHG realiza el intercambio entre los valores de los operandos. Puede tener operando en registros y en memoria: XCHG reg, mem XCHG reg, reg XCHG mem, reg

FUENTES DE INFORMACION

- <https://plataformaeducativauds.com.mx/assets/docs/libro/ISC/f3c10f122d8b67929274266e14d96544-LC-ISC702.pdf>