

LOS MICROPROCESADORES

INTRODUCCION

La historia del microprocesador va íntimamente ligada a la de la computadora, aunque no al revés, pues las primeras computadoras no incluían microprocesador. Si, aunque cueste creerlo, las primeras computadoras (por así llamarlas) eran mecánicas, ya en el siglo XVII, el filósofo y científico francés, Blaise Pascal, con solo 18 años, invento una maquina de calcular, con un complicado sistema de ruedas y engranajes. Esta maquina solo permitía la suma y resta, operaciones básicas, a partir de las cuales los ordenadores actuales ejecutan sentencias por si no lo sabias, un microprocesador, en realidad solo sabe sumar, y por medio de ciertos trucos, como el complemento a 2, es capaz de restar, multiplicar y dividir.

Desde esos tiempos a llovido mucho (o muy poco, es el cambio climático), y posteriormente se construyeron computadoras que basaban su capacidad de calculo en válvulas o relés. Los relés, son pequeños dispositivos electrónicos, similares a interruptores, pero de movimiento mecánico, realizado por una bobina, lo cual permitía tener un 0 o un 1 según estuviera el relés abierto o cerrado. Las válvulas, por su parte, eran dispositivos más avanzados y rápidos, pero que tenían en su contra, su gran fragilidad, y su alto consumo de energía eléctrica, unido a una considerable producción de calor. Pero la verdadera revolución (de la cual nació el microprocesador) fue cuando en 1947, se invento el primer transistor, dando paso a la segunda generación de computadoras. El primer ordenador basado en transistores, se construyo en Alemania, en 1957, de la mano de Siemens, el 2002, y aquí empezó una nueva era de la informática.

A partir de ese momento comenzó la carrera para la miniaturización, en este momento es hora de hablar de Jack St. Clair Kilby, al que se le negó entrar en el MIT (Instituto de Tecnología de Massachusetts), por pocos conocimientos matemáticos (¿?) y solo consiguió un puesto en una empresa que fabrica piezas para radios, con lo consiguió familiarizarse enormemente con los transistores, consiguiendo entonces entrar en Texas Instruments, cuyo proyecto era el desarrollo de micromodulos, una especie de placas con varios transistores que se unirían por cables para crear múltiples aparatos, pero Kilby pensó que eso seria una maraña de cables y se le ocurrió crear, en una sola plaquita de silicio de un centímetro cuadrado, varios dispositivos electrónicos (transistores, resistencias, condensadores) en "integrarlos" todos en uno, así nació el primer "integrado", lo que se conoce popularmente como CHIP. Fue poco después, en 1970, de la mano de Marcian Edward, Stanley Mazor y Federico Fanggin, que se creo lo que hoy conocemos como MICROPROCESADOR, el 4004 de Intel.

Un poco de historia.

Actualidad.

Hoy en día el numero de microprocesadores, así como el de fabricantes de estos se ha incrementado enormemente. Intel, creador del primer micro, conserva solo el liderazgo en la gama de ordenadores "compatibles IBM-PC", estando el mercado profesional dominado por otros fabricantes. He de decir que en el mercado de los PC, también Intel sé esta viendo terriblemente acosada, tras el fallo de sus Pentium, en la unidad de coma flotante, las cosas no han ido muy bien. Desde el nacimiento del 8086, para el primer IBM-PC, y su versión recortada, el 8088, Intel a tenido que luchar contra otras empresas, que viendo las carencias de los micros de Intel, se pusieron manos a la obra para el desarrollo de otros dispositivos. Tal es el caso de AMD, la cual, lleva fabricando Coprocesadores para IBM-PC, casi desde el nacimiento de este, con lo que Intel, para evitar posibles competencias, pacto con AMD, para el intercambio de tecnología y creación de un nuevo micro, el 486, con coprocesador matemático incluido en la pastilla del micro, esto trajo mucha tela, ya que AMD, se vio engañada, pues más que intercambio de tecnología, fue copia de sus técnicas de construcción de Coprocesadores, lo que trajo incontables juicios entre ambas empresas, sin un final claro.

A partir de entonces, AMD, que sus viejas barbas remojar, se dedicó a potenciar la producción de micros compatibles x86, (que ya fabricaba desde el 286), consiguiendo lo que hoy a dado a llamar AMD K6, un micro compatible x86, con una estructura que solo los P2 de Intel incluyen, (algunas ni eso) a un precio ridículo, con una cache de primer orden superior incluso al P2. Otra empresa creadora de compatibles x86 es Cyrix, que con su 6x86 se acerca a los Intel y AMD, pero no superándolos en rendimiento. Y la sorpresa del verano de 1997, fue el Centaur, un micro, compatible 586 MMX, a un precio ridículo.

Llegados aquí, retomo el tema de Intel, y su no tan buena situación. El PentiumPro, el sustituto de Pentium normal, no llegó a implantarse en el mercado de consumo (verdadera fuente de ingresos de Intel) por una razón, al ejecutar instrucciones de 16 Bits, su rendimiento apenas superaba a sus hermanos pequeños. Esto, unido a la situación que presentaba el mercado de los sistemas operativos PC, no convenció al mercado, que no lo implantó como estándar (Se dice que fue la propia Microsoft, la que metió mano, pues no podía permitir que sus sistemas operativos Windows 3.11, y 95, con código de 16 Bits, se vieran marginados frente a los auténticos de 32 Bits, como OS/2 Warp, UNIX, Linux, etc. que se verían potenciados), pasando a ser "renombrado" como micro para aplicaciones profesional, donde tampoco tiene muchas ventajas, pues por su precio, junto con la de sus placas (recordemos que necesita una placa especial para él), no podía competir con los ALPHA o MIPS. En este momento, al sentirse en una situación de pérdida de mercado, Intel sacó lo que según él, sería la revolución multimedia, los MMX, micros con un juego de 57 nuevas instrucciones, diseñadas para acelerar la ejecución de aplicaciones multimedia, y alguna mejora más (como el incremento de la cache interna del micro), pero sin el inconveniente de ejecutar mejor código de 32 que de 16 Bits. Lo que permitió a Intel no perder tanto mercado que con el PentiumPro. Este micro, verdadero "parche" para Intel, se ha implantado rápidamente, por no ser un salto tan grande con su generación anterior, y permitir esto, usar las placas con Socket 7 (el conector de 586 normal) para su implementación.

Metiéndonos en otros fabricantes, nos encontramos con Motorola, la cual, con su M68000 y familia, a conseguido un éxito más que merecido, este micro, RISC, con solo 7.14 MHz, fue la base para la creación de computadoras rapidísimas (para su época) como los primeros MAC, Atari St, y Amiga, tiene un bus de datos de 16 Bits y uno de direcciones de 32 Bits pudiendo direccionar hasta 16 Mb de memoria (todo un récord para aquel entonces), hay van más datos del. 17 registros de 32 Bits de datos y direcciones, acceso directo a un rango de direcciones de 16Mb, también llamado DMA, (lo que hoy es una novedad en los Pentium), 14 modos de direccionamiento, entrada/salida mapeada en memoria, esto hace que aun en día se encuentren 68000 en controladoras de disco llamados "inteligentes" y otros dispositivos, pues es muy económico y potente. Pero la familia 680x0 no acaba aquí (es solo el principio), le siguió el 68010, que compartía patillaje con el 68000, pudiendo intercambiarse ambos, y con una mejora del 30% en rendimiento. Ha esto le siguió el 68020, completo micro de 32 Bits, y con una cache interna de 256Kb, esta versión a 14MHz, permitía usar 4Gb de RAM, así como arquitectura segmentada, que permite ejecutar varias instrucciones simultáneamente, y otras maravillas (no olvidemos su compatibilidad con el 68000). El siguiente eslabón fue el 68030, similar al 68020, pero con capacidad para ejecutar instrucciones concurrentemente, esto es, en paralelo y mientras se accede a las caches internas, el copro o el controlador de buses, dos caches independientes para datos e instrucciones a las que puede acceder simultáneamente, control de acceso a BUS, por tres métodos, Síncrono (2 ciclos), Asíncrono, y Burst (1 ciclo). Y continuamos con la familia 680x0, esta vez con el 68040, a 40MHz, con Coprocesador incluido en el micro, dos caches de 4Kb, y un largo etc.

Aquí tenemos que pegar un salto de casi dos generaciones de 680x0 pues el 68050 nunca salió a la luz con ese nombre (dicen que se los quedó en exclusiva Ford, para sus automóviles), pasando al 68060, un micro realmente impresionante. Incluye dos caches, dos MMU independientes, (una para datos y otra para instrucciones), dos unidades segmentadas de enteros, unidad de calculo en coma flotante, arquitectura Harvard (replicación de bloques funcionales), con lo que consigue paralelismo de ejecución múltiples buses internos y unidades independientes de ejecución. El manejo del consumo de corriente es gestionado por el propio CHIP, entrando en bajo consumo solo con ejecutar una instrucción (LPSTOP), diseño estático, (cuando un circuito no sé esta usando no consume), ideal para un portátil. Este micro funciona a 50MHz, y creo que salió una versión a 66MHz, siendo tres veces más rápido que un 68040.

Llegados a este punto la cadena 680x0 se rompe, dando lugar a un nuevo microprocesador, fruto de un acuerdo entre IBM, Motorola y Apple que crearan un nuevo monstruo, el PowerPC.

El PowerPC, nace como una apuesta de futuro para los Mac, y otros equipos que quieran incorporar un micro de alto rendimiento a un precio moderado, fuera del monopolio de Intel. Este microprocesador, con unas cualidades espectaculares como varias caches internas, y capaz de ejecutar varias instrucciones en un solo ciclo de reloj, es un verdadero RISC, y solo consume 5 Vatios esto es imprescindible para construir portátiles de alto rendimiento, comparados con los 60 Vatios del PentiumPro a 200MHz (no sé lo que consume el P2, pero imagínate, más que el Pro). Esto demuestra la buena construcción del PowerPC, partiendo de una buena base tecnológica.

Hay que tener en cuenta que el calor es el pero enemigo de la electrónica, y el que impide miniaturizar y acelerar los micros, y a mayor consumo mayor disipación de calor, Y sino piensa, ¿Quien se compraría un portátil a 333MHz, como un P2, si la batería no daría para más de media hora?

El PowerPC, además de ser el corazón de los PowerMac, también lo es del nuevo PowerUp, un Amiga, con microprocesador M68060 y un PowerPC trabajando en paralelo, y será implementado en los ordenadores de red, o NetComputer.

Una baza muy importante a favor del PowerPC es la cantidad de sistemas operativos que es capaz de correr, UNIX, Linux, MAC-OS, y como no, Windows NT.

Y vamos ahora con los un micro realmente alucinante. El ALPHA, fabricado por DIGITAL, un todopoderoso RISC, capaz de ejecutar hasta 4 instrucciones por ciclo de reloj, con varias caches incluidas en el micro, entre ellas, la de segundo nivel, todo dentro del micro, además de una caché de tercer nivel, incluida en la placa base, las caches de datos e instrucciones son de 8 Kb cada una (No tengo datos actualizados, esto es del los ALPHA 21164) y una caché de 96Kb de segundo orden, la caché de tercer nivel, suele ser de unos 4Mb. Con dos unidades de ejecución de enteros y dos de coma flotante, consigue ejecutar hasta 4 instrucciones por ciclo. Las unidades de enteros son iguales, pero las de coma flotante son distintas, una especializada en sumas, y la otra en operaciones complejas, todos esto, forma uno de los mejores (sino el mejor) de los micros de 64Bit (recordemos que Intel continua con los 32bit, incluido el P2). En cuanto a las frecuencias de trabajo, el ALPHA, fue el primer micro RISC en superar la barrera de los 200MHz, vendiéndose hoy a unos "tranquilos" 500MHz, con un consumo de 40 Vatios, y pronto, pero que muy pronto, saldrá una nueva generación de ALPHA, los 21264, trabajando a 1GHZ. Como nota, entre un ALPHA y un Intel a la misma velocidad (en MHz), el ALPHA da cuatro veces más rendimiento que el Intel.

Otro gran desconocido es el PA-RISC 8500, un micro de 64 bit, fabricado por HP, con arquitectura similar a la de un PowerPC, este micro, consigue en su versión a 180 MHz, un rendimiento tal, que para igualarlo, seria necesario hacer un PentiumPro a 900 MHz (toma esa Intel). Pero el futuro de estos micros es incierto, dado que HP e Intel se acordaron unir tecnologías para lo que será el Merced, un prototipo de micro de 64 Bit para compatibles IBM-Pc.

Proyectos Futuros.

El futuro se ve de diferentes maneras, según el micro en que pensemos. Hay van algunos proyectos de distintas casas:

Empezamos por un compatible x86, AMD y sus K series. El futuro más reciente es la aparición de un K6 con aceleración MPG y Direct 3D, el AMD K6 3D, seguido del AMD K6 3D III, para el segundo semestre de los años, llegando hasta los 500MHz. Y la historia no se acaba aquí, pronto saldrá el K7, un micro que superara los 600MHz, con una caché de segundo orden incluida dentro del propio micro, llegando a incluir hasta 20 Millones de transistores (Recordemos que un Pentium MMX tiene 4,4 Millones de transistores, y el K6, 8,8

Millones, el doble).

Entre los Proyectos más destacados de ALPHA, esta la comercialización de su ALPHA 21264 a 1 GHz, con más de 20 millones de transistores.

Para Intel el futuro pasa por migrar sus actuales micros de 32 Bit a 64 Bit, pero antes de ello, en 1999 veremos el último de sus micros de 32 Bit, el Katmai, este micro incorporara lo que Intel a dado en llamar MMX2 con 70 nuevas instrucciones para la aceleración de render y otros apartados gráficos. Después de Katmai (que llegara hasta los 500 MHz), marcando una separación entre el mercado domestico y el profesional, con la implementación de un nuevo slot para microprocesadores, el SLOT 2 (otro más). Para el 2000 saldrá el Merced, primer micro de 64 Bit, basado en lo que Intel y HP han dado a llamar ISA-64, (Instruction Set Architecture 64).

ÚLTIMOS LANZAMIENTOS:

Intel lanza al mercado un Pentium II sin cache, llamado Celeron, destinado principalmente al mercado de consumo.

AMD presenta su K6-2, también llamado K6 3D, pues incluye un nuevo juego de instrucciones para acelerar procesos 3D. Este nuevo juego recibe el nombre de 3D Now!. Este micro se montara sobre placas Súper 7 (las mismas que para el K6, Cyrix, Winchip y Pentium normal o MMX). Las velocidades a las que se entregara este micro son 266, 300, 350, 366, 380, 400 MHz.

El juego de instrucciones 3D Now! es incluido en la ultima generación de micros de Cyrix y Winchip.

Intel Presenta un Pentium Celeron mejorado, con 128K de cache de nivel 2 (L2) llamado 'Mendocino' o también Celeron A.

AMD consigue el apoyo de Microsoft para que las DirectX 6 y superiores hagan uso de las 3D Now!, así como de la industria del videojuego, con la optimización de muchos juegos para este tipo de micros (Quake III, Unreal, Via Voice 98, Incoming, etc.).

AMD consigue un acuerdo con las empresas fabricantes de tarjetas gráficas para que el driver de estas este optimizado para 3D Now!, repartándose el trabajo entre las 3D now! y la tarjeta gráfica, según quien sea el más rápido, consiguiendo así un aumento de prestaciones.

SAMSUNG compra derechos a DIGITAL para fabricar micros y placas ALPHA, lo que los hará más accesibles al mercado en general.

OpenGL 1.2 utilizara también las 3D Now!.

Intel Presenta un micro para estaciones de trabajo, llamado Pentium III Xeon, este micro, sustituto del Pentium Pro se caracteriza por tener una cache de nivel 2 (L2) de 1024 ó 2048, y permite configuraciones de hasta cuatro micros en una sola placa.

25 Febrero 1999 – AMD presenta su K6-III, con 256K de cache L2 incluidas dentro del micro (recordar que el Pentium II y III no la llevan dentro, sino al lado, en la misma placa-tarjeta) al estilo de como Intel nos ofreció su Pentium Pro. Este micro se monta sobre las mismas placas que sus predecesores y saldrá en unas velocidades iniciales de 400 y 450 MHz. Cuenta con la ventaja de poder gestionar tres niveles de cache, sumando así la capacidad de la L1 (64k en los K6 I, II y III y 32k en los Pentium II y III) más la L2 (256 en el K6-III a la misma velocidad que el micro y 512 en los Pentium II y III a la mitad de la velocidad del micro) más la incluida en la placa (Pentium II y III no puede gestionar esta memoria) que podrá ser de más de 2 Mb.

Según AMD un K6-III a 450MHz rinde más que un Pentium III a 500MHz. También incluirá 3D Now!.

26 Febrero 1999 – Compaq saca al mercado los primeros Compaq Presario K6-III a 400 y 450 MHz.

28 Febrero 1999 – Intel presenta su Pentium III (antes llamados Katmai) y Pentium III Xeon (Tanner) con un nuevo juego de instrucciones al que ya se le conoce como MMX 2. Los micros estarán disponibles en el mercado con unas velocidades iniciales de 400, 450, 500, 550 y 600 MHz para finales de año.

Abril 1999 – Por primera vez en las ultimas décadas AMD ha superado en ventas de procesadores a Intel, durante el mes de Enero de 1999. Según PC DAT, una empresa Americana encargada de realizar estadísticas relacionadas con los ordenadores, durante Enero de 1999 AMD vendió el 43.9% de todos los procesadores para PC en Estados Unidos, superando en varios puntos a los micros de Intel.

7 Abril 1999 – AMD declara en la 'Windows Hardware Engineering Conference and Exhibition (WINHEX 99) que el lanzamiento de K7 será en JUNIO 1999. A continuación se muestra una imagen de un prototipo publicada en una revista del sector.

Mayo 1999 – Intel lanza los Mendocino (Celeron A) en formato PPGA, para un nuevo conector de placa llamado 'Socket 370', muy similar en apariencia al Socket 7, pero incompatible entre si, evitando así que se monten Celeron 'Overclockeados' en Slot 1 y evitar quitar mercado a sus Pentium II/III.

Junio 1999 – AMD lanza el K7, cuyo nombre en final es ATHLON, sale al mercado con unas frecuencias de trabajo de 500, 550 y 600 MHz. Tiene una cache de primer nivel de 128Kb y de segundo nivel de 512Kb ampliable hasta 8 Mb.

Por primera vez es capaz de trabajar en multiproceso, en configuraciones de 2, 4, 8 y se espera que 16 microprocesadores en la misma placa. Es el primer micro con un coprocesador Super-escalar, y ofrece un rendimiento superior al Pentium III Xeon.

PROCESADORES MOTOROLA

Estructura de aprendizaje para la familia 68000

Siempre es difícil, excepto en el caso en que ya se esté familiarizado con microprocesadores, el realizar un buen aprendizaje, puesto que uno no sabe muy bien por donde empezar. Lo primero de todo no es conocer y aprender a manejar el 68000, sino aprender a trabajar con cualquier microprocesador. Esto es, conocer su estructura básica, saber lo que significa un registro, dominar la terminología específica de microprocesadores, etc.

Cuando todo esto esté bien asentado, será posible adentrarse en el mundo de los 680x0; quede claro también que a la hora de estudiar la teoría que envuelve a los uPs, es recomendable apoyarse en un ejemplo práctico fácil de entender, que en este caso es 68000.

Pasando al estudio de la familia 680x0, como en todo, es recomendable no quemar etapas y empezar por el microprocesador más básico, el 68000.

Todos los modelos posteriores se basan en este y por lo tanto su comprensión será más asequible si se conoce lo conoce bien.

El modelo 68010 no ofrece grandes adelantos (en comparación a otros saltos entre generaciones posteriores), aunque siempre es interesante examinar lo que ofreció en su día. Posteriormente, el estudio de los modelos 68020 y 68030 permite el estudio de técnicas de gestión de caché y memoria virtual, conceptos fundamentales en la informática actual.

El 68040 ofrece mejoras sustanciales sobre sus predecesores y ofrece una potencia que pocos podían igualar al integrar en el mismo chip una FPU y una MMU.

Finalmente, el 68060 permite el estudio de un microprocesador RISC con coprocesador matemático y manejador de memoria virtual incorporados que consume y disipa menos potencia que el 68040. Es el más potente de la familia, además de su último eslabón puesto que Motorola ha abandonado la familia 680x0 a favor de su familiar PowerPC.

Los coprocesadores matemáticos o unidades de coma flotante:

Una FPU (floating point unit) es un chip, o parte de un chip, encargado de procesar operaciones en coma flotante; en otras palabras, efectuar operaciones matemáticas complejas y muy potentes.

Los copros 68881 – 68882

Existen varios modelos de FPUs en Amiga: el 68881, el 68882, y la FPU interna del 68040 y del 68060.

La diferencia entre el 68881 y el 68882 es que este último procesa la información unas 1,5 veces más rápido que el primero. Esto es debido a que está dividido en dos partes: una unidad de conversión (la FPU trabaja internamente en formato de 80 bits) y la unidad aritmética.

La FPU del 68040 posee además una pipeline pero en contrapartida pierde las funciones trigonométricas, las cuales se ven suplidas por librerías externas del 68040, como por ejemplo la 68040.library y son calculadas por software.

Los microprocesadores 68000, 68010, 68020, 68030, así como sus versiones reducidas no poseen coprocesador matemático integrado; es decir, solo saben procesar enteros. Los procesos de coma flotante se suplían en estos casos vía software.

Por la parte soft, existen versiones de ciertos programas (aquellos que toquen ámbitos como la infografía, los cálculos matemáticos, etc..) específicas para ordenadores con FPU que llegan a velocidades hasta 50 veces superiores que las versiones carentes de FPU originales.

Diferencias en modelos EC y LC:

Motorola además de sacar modelos superiores con el paso del tiempo, produjo ciertas versiones recortadas de sus otros microprocesadores anteriores. Estos modelos son algo más baratos que las versiones "completas" y estaban orientados a aplicaciones específicas.

Los micros 68EC020, 68EC030 y 68LC040 se montan en algunos ordenadores y aceleradoras Amiga. Como ejemplos, se pueden citar el A1200 que lleva un 68EC020 y el A4000/30 un 68EC030.

La diferencia entre un 68020 y un 68EC20 es que este último no puede direccionar más de 16 MB de memoria. Es por ello que el A1200 no puede tener más de 10MB de memoria (2MB de Chip, 8 de Fast opcional y 6MB para el sistema).

Como precaución habrá que asegurarse de no usar un mapa de memoria incorrecto y ceñirse a los 16MB permitidos.

En el caso del 68EC030, su modificación radica en la eliminación de la MMU que el 68030 sí posee. Esto puede suponer un serio inconveniente, puesto que muchos programas de gestión de memoria no podrán ser usados.

Hay que tener cuidado a la hora de programar en no usar los registros de tratamiento de caché, puesto que no existen. En el caso de tener un 68EC030 PGA se puede reemplazar por un modelo 68030 sin problemas puesto que poseen el mismo patillaje. si se posee un PLCC, el reemplazo puede ser complicado a menos que venga instalado en un zócalo.

Por último, el 68LC040 es un 68040 al que se le ha suprimido la FPU, por lo que una llamada a una de las instrucciones correspondientes a la FPU generará un error.

Todos estos modelos recortados conservan sin embargo todo el juego de instrucciones y los modos de direccionamiento respectivos.

Familia PowerPC de Motorola

Los microprocesadores PowerPC nacieron de la unión Motorola–Apple–IBM con la intención de crear un microprocesador de futuro y que pudiese cumplir una característica largamente buscada: ser multiplataforma.

Esto significa que debe poder ejecutar todo tipo de programas independientemente del hardware y del sistema operativo en el que se integre (siempre que se cumplan ciertas reglas básicas, claro está).

Al paso del tiempo se han sacado nuevos modelos de PowerPC (también conocidos como PPC o MPC), a cada cual más potente o más dedicado a un ámbito determinado. Esta gran potencia se deba en gran parte a su arquitectura enteramente RISC.

Modelos PowerPC:

PPC 601/601v---- PPC 602----- PPC 603----- PPC 604----- PPC 620

PowerPC 620

Este el cuarto modelo desarrollado para la familia PowerPC y el primero en implementar una arquitectura íntegramente de 64 bits.

Se obtienen una ganancias en prestaciones 8 veces superiores con respecto al 601 y la frecuencia a la que se ofrece es de 133MHz.

Su arquitectura se basa en un proceso de 0,5 micras CMOS y alimentada con 3,3v de tensión, conservando la compatibilidad CMOS/TTL.

Posee 2 caches de datos e instrucciones de 32KB cada una, 2 MMUs, buses de 128 bits de datos y 40 de direccionamiento. El empaquetado es esta vez de 625 pines.

PROCESADORES INTEL

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

Los microprocesadores 8086 y 8088

Historia del 8086/8088

En junio de 1978 Intel lanzó al mercado el primer microprocesador de 16 bits: el 8086. En junio de 1979 apareció el 8088 (internamente igual que el 8086 pero con bus de datos de 8 bits) y en 1980 los coprocesadores 8087 (matemático) y 8089 (de entrada y salida). El primer fabricante que desarrolló software y hardware para estos chips fue la propia Intel. Reconociendo la necesidad de dar soporte a estos circuitos

integrados, la empresa invirtió gran cantidad de dinero en un gran y moderno edificio en Santa Clara, California, dedicado al diseño, fabricación y venta de sus sistemas de desarrollo que, como se explicó anteriormente, son computadoras autosuficientes con el hardware y software necesario para desarrollar software de microprocesadores.

Los sistemas de desarrollo son factores clave para asegurar las ventas de una empresa fabricantes de chips. La inmensa mayoría de ventas son a otras empresas, las cuales usan estos chips en aparatos electrónicos, diseñados, fabricados y comercializados por ellas mismas. A estas empresas se las llama "fabricantes de equipo original", o en inglés, OEM (Original Equipment Manufacturer). El disminuir el tiempo de desarrollo de hardware y software para las OEM es esencial, ya que el mercado de estos productos es muy competitivo. Necesitan soporte pues los meses que les puede llevar el desarrollo de las herramientas apropiadas les puede significar pérdidas por millones de dólares. Además quieren ser los primeros fabricantes en el mercado, con lo cual pueden asegurarse las ventas en dos áreas importantes: a corto plazo, ya que al principio la demanda es mucho mayor que la oferta, y a largo plazo, ya que el primer producto marca a menudo los estándares.

De esta manera la empresa Intel había desarrollado una serie completa de software que se ejecutaba en una microcomputadora basada en el 8085 llamada "Intellec Microcomputer Development System". Los programas incluían ensambladores cruzados (éstos son programas que se ejecutan en un microprocesador y generan código de máquina que se ejecuta en otro), compiladores de PL/M, Fortran y Pascal y varios programas de ayuda. Además había un programa traductor llamado CON V86 que convertía código fuente 8080/8085 a código fuente 8086/8088. Si se observan de cerca ambos conjuntos de instrucciones, queda claro que la transformación es sencilla si los registros se traducen así: A → AL, B → CH, C → CL, D → DH, E → DL, H → BH y L → BL. Puede parecer complicado traducir LDAX B (por ejemplo) ya que el 8088 no puede utilizar el registro CX para direccionamiento indirecto, sin embargo, se puede hacer con la siguiente secuencia: MOV SI, CX; MOV AL, [SI]. Esto aprovecha el hecho que no se utiliza el registro SI. Por supuesto el programa resultante es más largo (en cantidad de bytes) y a veces más lento de correr que en su antecesor 8085. Este programa de conversión sólo servía para no tener que volver a escribir los programas en una primera etapa. Luego debería reescribirse el código fuente en assembler para poder obtener las ventajas de velocidad ofrecidas por el 8088. Luego debía correr el programa en la iSBC 86/12 Single Board Computer basado en el 8086. Debido al engorro que resultaba tener dos plaquetas diferentes, la empresa Godbout Electronics (también de California) desarrolló una placa donde estaban el 8085 y el 8088, donde se utilizaba un ensamblador cruzado provisto por la compañía Microsoft. Bajo control de software, podían conmutarse los microprocesadores. El sistema operativo utilizado era el CP/M (de Digital Research).

El desarrollo más notable para la familia 8086/8088 fue la elección de la CPU 8088 por parte de IBM (International Business Machines) cuando en 1981 entró en el campo de las computadoras personales. Esta computadora se desarrolló bajo un proyecto con el nombre "Acorn" (Proyecto "Bellota") pero se vendió bajo un nombre menos imaginativo, pero más correcto: "Computadora Personal IBM", con un precio inicial entre 1260 dólares y 3830 dólares según la configuración (con 48KB de memoria RAM y una unidad de discos flexibles con capacidad de 160KB costaba 2235 dólares). Esta computadora entró en competencia directa con las ofrecidas por Apple (basado en el 6502) y por Radio Shack (basado en el Z-80).

Arquitectura de los procesadores 8088 y 8086:

El 8086 es un microprocesador de 16 bits, tanto en lo que se refiere a su estructura como en sus conexiones externas, mientras que el 8088 es un procesador de 8 bits que internamente es casi idéntico al 8086. La única diferencia entre ambos es el tamaño del bus de datos externo. Intel trata esta igualdad interna y desigualdad externa dividiendo cada procesador 8086 y 8088 en dos sub-procesadores. O sea, cada uno consta de una unidad de ejecución (EU: Execution Unit) y una unidad interfaz del bus (BIU: Bus Interface Unit). La unidad de ejecución es la encargada de realizar todas las operaciones mientras que la unidad de interfaz del bus es la encargada de acceder a datos e instrucciones del mundo exterior. Las unidades de ejecución son idénticas en ambos microprocesadores, pero las unidades de interfaz del bus son diferentes en varias cuestiones, como se

desprende del siguiente diagrama en bloques:

```
<MAP NAME="bloques"><AREA SHAPE="RECT" COORDS="33,77,212,91" ALT="Registros de uso
general" HREF="#reg_gral"><AREA SHAPE="RECT" COORDS="75,118,171,136" ALT="Registro AX"
HREF="#AX"><AREA SHAPE="RECT" COORDS="75,138,171,156" ALT="Registro BX"
HREF="#BX"><AREA SHAPE="RECT" COORDS="75,158,171,176" ALT="Registro CX"
HREF="#CX"><AREA SHAPE="RECT" COORDS="75,178,171,196" ALT="Registro DX"
HREF="#DX"><AREA SHAPE="RECT" COORDS="75,198,171,216" ALT="Registro SP"
HREF="#SP"><AREA SHAPE="RECT" COORDS="75,218,171,236" ALT="Registro BP"
HREF="#BP"><AREA SHAPE="RECT" COORDS="75,238,171,256" ALT="Registro SI"
HREF="#SI"><AREA SHAPE="RECT" COORDS="75,258,171,276" ALT="Registro DI"
HREF="#DI"><AREA SHAPE="RECT" COORDS="182,122,198,132" ALT="AX" HREF="#AX"><AREA
SHAPE="RECT" COORDS="182,142,198,152" ALT="BX" HREF="#BX"><AREA SHAPE="RECT"
COORDS="182,162,198,172" ALT="CX" HREF="#CX"><AREA SHAPE="RECT"
COORDS="182,182,198,192" ALT="DX" HREF="#DX"><AREA SHAPE="RECT"
COORDS="182,202,198,212" ALT="SP" HREF="#SP"><AREA SHAPE="RECT"
COORDS="182,222,198,232" ALT="BP" HREF="#BP"><AREA SHAPE="RECT"
COORDS="182,242,198,252" ALT="SI" HREF="#SI"><AREA SHAPE="RECT"
COORDS="182,262,198,272" ALT="DI" HREF="#DI"><AREA SHAPE="RECT"
COORDS="285,161,381,239" ALT="Registros de segmento" HREF="#reg_seg"><AREA
SHAPE="RECT" COORDS="82,540,174,558" ALT="Indicadores" HREF="#flags"><AREA
SHAPE="RECT" COORDS="255,386,343,434" ALT="Control EU" HREF="#ctrl_eu"><AREA
SHAPE="RECT" COORDS="423,26,510,94" ALT="Control bus" HREF="#control_bus"><AREA
SHAPE="RECT" COORDS="383,401,476,427" ALT="Cola de instrucciones" HREF="#cola_instr"><AREA
SHAPE="POLY" COORDS="79,442,100,442,106,459,144,459,150,442,171,442,152,498,98,498"
ALT="ALU" HREF="#alu"></MAP>
```

La ventaja de esta división fue el ahorro de esfuerzo necesario para producir el 8088. Sólo una mitad del 8086 (el BIU) tuvo que rediseñarse para producir el 8088.

Unidad aritmética y lógica

Es la encargada de realizar las operaciones aritméticas (suma, suma con "arrastre", resta, resta con "préstamo" y comparaciones) y lógicas (AND, OR, XOR y TEST). Las operaciones pueden ser de 16 bits o de 8 bits.

Indicadores (*flags*)

Hay nueve indicadores de un bit en este registro de 16 bits. Los cuatro bits más significativos están indefinidos, mientras que hay tres bits con valores determinados: los bits 5 y 3 siempre valen cero y el bit 1 siempre vale uno (esto también ocurría en los procesadores anteriores).

Registro de indicadores (16 bits)

CF (Carry Flag, bit 0): Si vale 1, indica que hubo "arrastre" (en caso de suma) hacia, o "préstamo" (en caso de resta) desde el bit de orden más significativo del resultado. Este indicador es usado por instrucciones que suman o restan números que ocupan varios bytes. Las instrucciones de rotación pueden aislar un bit de la memoria o de un registro poniéndolo en el CF.

PF (Parity Flag, bit 2): Si vale uno, el resultado tiene paridad par, es decir, un número par de bits a 1. Este indicador se puede utilizar para detectar errores en transmisiones.

AF (Auxiliary carry Flag, bit 4): Si vale 1, indica que hubo "arrastre" o "préstamo" del nibble (cuatro bits)

menos significativo al nibble más significativo. Este indicador se usa con las instrucciones de ajuste decimal.

ZF (Zero Flag, bit 6): Si este indicador vale 1, el resultado de la operación es cero.

SF (Sign Flag, bit 7): Refleja el bit más significativo del resultado. Como los números negativos se representan en la notación de complemento a dos, este bit representa el signo: 0 si es positivo, 1 si es negativo.

TF (Trap Flag, bit 8): Si vale 1, el procesador está en modo paso a paso. En este modo, la CPU automáticamente genera una interrupción interna después de cada instrucción, permitiendo inspeccionar los resultados del programa a medida que se ejecuta instrucción por instrucción.

IF (Interrupt Flag, bit 9): Si vale 1, la CPU reconoce pedidos de interrupción externas enmascarables (por el pin INTR). Si vale 0, no se reconocen tales interrupciones. Las interrupciones no enmascarables y las internas siempre se reconocen independientemente del valor de IF.

DF (Direction Flag, bit 10): Si vale 1, las instrucciones con cadenas sufrirán "auto-decremento", esto es, se procesarán las cadenas desde las direcciones más altas de memoria hacia las más bajas. Si vale 0, habrá "auto-incremento", lo que quiere decir que las cadenas se procesarán de "izquierda a derecha".

OF (Overflow flag, bit 11): Si vale 1, hubo un desborde en una operación aritmética con signo, esto es, un dígito significativo se perdió debido a que tamaño del resultado es mayor que el tamaño del destino.

Sistema de control de la unidad de ejecución

Es el encargado de decodificar las instrucciones que le envía la cola y enviarle las órdenes a la unidad aritmética y lógica según una tabla que tiene almacenada en ROM llamada CROM (Control Read Only Memory).

Cola de instrucciones

Almacena las instrucciones para ser ejecutadas. La cola se carga cuando el bus está desocupado, de esta manera se logra una mayor eficiencia del mismo. La cola del 8086 tiene 6 bytes y se carga de a dos bytes por vez (debido al tamaño del bus de datos), mientras que el del 8088 tiene cuatro bytes. Esta estructura tiene rendimiento óptimo cuando no se realizan saltos, ya que en este caso habría que vaciar la cola (porque no se van a ejecutar las instrucciones que van después del salto) y volverla a cargar con instrucciones que se encuentran a partir de la dirección a donde se salta. Debido a esto las instrucciones de salto son (después de multiplicaciones y divisiones) las más lentas de este microprocesador.

Registros de la unidad de interfaz con el bus:

El programador puede acceder a cinco registros de 16 bits cada uno, siendo cuatro de ellos registros de segmento y el restante el puntero de instrucción (IP).

Los registros de segmento se llaman:

CS: Registro de segmento de código.

DS: Registro de segmento de datos.

ES: Registro de segmento extra.

SS: Registro de segmento de pila.

La utilización de estos registros se explica más adelante, en la sección que trata de direccionamiento a memoria.

Lógica de control del bus:

El cometido de este bloque es poder unir los bloques anteriormente mencionados con el mundo exterior, es decir, la memoria y los periféricos.

El 8088 tiene un bus de datos externo reducido de 8 bits. La razón para ello era prever la continuidad entre el 8086 y los antiguos procesadores de 8 bits, como el 8080 y el 8085. Teniendo el mismo tamaño del bus (así como similares requerimientos de control y tiempo), el 8088, que es internamente un procesador de 16 bits, puede reemplazar a los microprocesadores ya nombrados en un sistema ya existente.

El 8088 tiene muchas señales en común con el 8085, particularmente las asociadas con la forma en que los datos y las direcciones están multiplexadas, aunque el 8088 no produce sus propias señales de reloj como lo hace el 8085 (necesita un chip de soporte llamado 8284, que es diferente del 8224 que necesitaba el microprocesador 8080). El 8088 y el 8085 siguen el mismo esquema de compartir los terminales correspondientes a los 8 bits más bajos del bus de direcciones con los 8 bits del bus de datos, de manera que se ahorran 8 terminales para otras funciones del microprocesador. El 8086 comparte los 16 bits del bus de datos con los 16 más bajos del bus de direcciones. El 8085 y el 8088 pueden, de hecho, dirigir directamente los mismos chips controladores de periféricos. Las investigaciones de hardware para sistemas basados en el 8080 o el 8085 son, en su mayoría, aplicables al 8088.

En todo lo recién explicado se basó el éxito del 8088.

Terminales (*pinout*) del 8088

Este microprocesador está encapsulado en el formato **DIP** (*Dual Inline Package*) de 40 patas (veinte de cada lado). La distancia entre las patas es de 0,1 pulgadas (2,54 milímetros), mientras que la distancia entre patas enfrentadas es de 0,6 pulgadas (15,32 milímetros).

Nótese en el gráfico el semicírculo que identifica la posición de la pata 1. Esto sirve para no insertar el chip al revés en el circuito impreso.

El 8086/8088 puede conectarse al circuito de dos formas distintas: el modo máximo y el modo mínimo. El modo queda determinado al poner un determinado terminal (llamado MN/MX) a tierra o a la tensión de alimentación. El 8086/8088 debe estar en modo máximo si se desea trabajar en colaboración con el Procesador de Datos Numérico 8087 y/o el Procesador de Entrada/Salida 8089 (de aquí se desprende que en la IBM PC el 8088 está en modo máximo). En este modo el 8086/8088 depende de otros chips adicionales como el Controlador de Bus 8288 para generar el conjunto completo de señales del bus de control. El modo mínimo permite al 8086/8088 trabajar de una forma más autónoma (para circuitos más sencillos) en una manera casi idéntica al microprocesador 8085.

Los 40 pines del 8088 en modo mínimo tienen las siguientes funciones:

- **GND** (Masa)
- **A14** (Bus de direcciones)
- **A13** (Bus de direcciones)
- **A12** (Bus de direcciones)
- **A11** (Bus de direcciones)
- **A10** (Bus de direcciones)
- **A9** (Bus de direcciones)

- **A8** (Bus de direcciones)
- **AD7** (Bus de direcciones y datos)
- **AD6** (Bus de direcciones y datos)
- **AD5** (Bus de direcciones y datos)
- **AD4** (Bus de direcciones y datos)
- **AD3** (Bus de direcciones y datos)
- **AD2** (Bus de direcciones y datos)
- **AD1** (Bus de direcciones y datos)
- **AD0** (Bus de direcciones y datos)
- **NMI** (Entrada de interrupción no enmascarable)
- **INTR** (Entrada de interrupción enmascarable)
- **CLK** (Entrada de reloj generada por el 8284)
- **GND** (Masa)
- **RESET** (Para inicializar el 8088)
- **READY** (Para sincronizar periféricos y memorias lentas)
- **/TEST**
- **/INTA** (El 8088 indica que reconoció la interrupción)
- **ALE** (Cuando está uno indica que salen direcciones por AD, en caso contrario, es el bus de datos)
- **/DEN** (Data enable: cuando vale cero debe habilitar los transceptores 8286 y 8287 (se conecta al pin de "output enable"), esto sirve para que no se mezclen los datos y las direcciones).
- **DT/R** (Data transmit/receive: se conecta al pin de dirección de los chips recién indicados).
- **IO/M** (Si vale 1: operaciones con ports, si vale 0: operaciones con la memoria)
- **/WR** (Cuando vale cero hay una escritura)
- **HLDA** (Hold Acknowledge: el 8088 reconoce el HOLD)
- **HOLD** (Indica que otro integrado quiere adueñarse del control de los buses, generalmente se usa para DMA o acceso directo a memoria).
- **/RD** (Cuando vale cero hay una lectura)
- **MN/MX** (Cuando esta entrada está en estado alto, el 8088 está en modo mínimo, en caso contrario está en modo máximo)
- **/SSO** (Junto con IO/M y DT/R esta salida sirve para determinar estados del 8088)
- **A19/S6** (Bus de direcciones/bit de estado)
- **A18/S5** (Bus de direcciones/bit de estado)
- **A17/S4** (Bus de direcciones/bit de estado)
- **A16/S3** (Bus de direcciones/bit de estado)
- **A15** (Bus de direcciones)
- **Vcc** (+5V)

En modo máximo (cuando se aplica +5V al pin 33) hay algunos pines que cambian de significado:

24.– **QS1**: Estado de la cola de instrucciones (bit 1).

25.– **QS0**: Estado de la cola de instrucciones (bit 0).

26.– **S0**: Bit de estado 0.

27.– **S1**: Bit de estado 1.

28.– **S2**: Bit de estado 2.

29.– **/LOCK**: Cuando vale cero indica a otros controladores del bus (otros microprocesadores o un dispositivo de DMA) que no deben ganar el control del bus. Se activa poniéndose a cero cuando una instrucción tiene el prefijo LOCK.

30.– **RQ/GT1**: Es bidireccional y tiene la misma función que HOLD/HLDA en modo mínimo.

31.– **RQ/GT0**: Como RQ/GT1 pero tiene mayor prioridad.

34.– Esta salida siempre está a uno.

Por ser este microprocesador mucho más complejo que el 8085, tiene más bits de estado que el recién mencionado. A título informativo se detallan los bits de estado:

S2	IO/M	DT/R	/SSO	Significado	S1	S0
1	0	0	0	Acceso a código (instrucciones)		
1	0	0	1	Lectura de memoria		
1	0	1	0	Escritura a memoria		
1	0	1	1	Bus pasivo (no hace nada)		
0	1	0	0	Reconocimiento de interrupción		
0	1	0	1	Lectura de puerto de entrada/salida		
0	1	1	0	Escritura a puerto de E/S		
0	1	1	1	Estado de parada (Halt)		

QS1	QS0	Significado
0	0	No hay operación
0	1	Primer byte del código de operación
1	0	Se vacía la cola de instrucciones
1	1	Siguiente byte de la instrucción

Modos de direccionamiento del 8086/8088:

Estos procesadores tienen 27 modos de direccionamiento (una cantidad bastante más grande que los microprocesadores anteriores) o reglas para localizar un operando de una instrucción. Tres de ellos son comunes a microprocesadores anteriores: **direccionamiento inmediato** (el operando es un número que se encuentra en la misma instrucción), **direccionamiento a registro** (el operando es un registro del microprocesador) y **direccionamiento inherente** (el operando está implícito en la instrucción, por ejemplo, en la multiplicación uno de los operandos siempre es el acumulador). El resto de los modos sirve para localizar un operando en memoria. Para facilitar la explicación de estos modos, se pueden resumir de la siguiente manera:

Deben sumarse cuatro cantidades: 1) **dirección de segmento**, 2) **dirección base**, 3) una **cantidad índice** y 4) un **desplazamiento**.

La dirección de segmento se almacena en el *registro de segmento* (DS, ES, SS o CS). En la próxima sección se indica la forma en que se hace esto. Por ahora basta con saber que el contenido del registro de segmento se multiplica por 16 antes de utilizarse para obtener la dirección real. El registro de segmentación siempre se usa para referenciar a memoria.

La base se almacena en el *registro base* (BX o BP). El índice se almacena en el *registro índice* (SI o DI). Cualquiera de estas dos cantidades, la suma de las dos o ninguna, pueden utilizarse para calcular la dirección real, pero no pueden sumarse dos bases o dos índices. Los registros restantes (AX, CX, DX y SP) no pueden utilizarse para direccionamiento indirecto. El programador puede utilizar tanto la base como el índice para gestionar ciertas cosas, tales como matrices de dos dimensiones, o estructuras internas a otras estructuras,

esquemas que se utilizan en las prácticas comunes de programación. La base y el índice son variables o dinámicas, ya que están almacenadas en registros de la CPU. Es decir, pueden modificarse fácilmente mientras se ejecuta un programa.

Además del segmento, base e índice, se usa un desplazamiento de 16 bits, 8 bits o 0 bits (sin desplazamiento). Ésta es una cantidad estática que se fija al tiempo de ensamblado (paso de código fuente a código de máquina) y no puede cambiarse durante la ejecución del programa (a menos que el programa se escriba sobre sí mismo, lo que constituye una práctica no aconsejada).

Todo esto genera los 24 modos de direccionamiento a memoria que se ven a continuación:

- **Registro indirecto:** 1) [BX], 2) [DI], 3) [SI].
- – **Basado:** 4) desp8[BX], 5) desp8[BP], 6) desp16[BX], 7) desp16[BP].
- **Indexado:** 8) desp8[SI], 9) desp8[DI], 10) desp16[SI], 11) desp16[DI].
- **Basado-indexado:** 12) [BX+SI], 13) [BX+DI], 14) [BP+SI], 15) [BX+DI].
- **Basado-indexado con desplazamiento:** 16) desp8[BX+SI], 17) desp8[BX+DI], 18) desp8[BP+SI], 19) desp8[BX+DI], 20) desp16[BX+SI], 21) desp16[BX+DI], 22) desp16[BP+SI], 23) desp16[BX+DI].
- **Directo:** 24) [desp16].

Aquí desp8 indica desplazamiento de 8 bits y desp16 indica desplazamiento de 16 bits. Otras combinaciones no están implementadas en la CPU y generarán error al querer ensamblar, por ejemplo, ADD CL,[DX+SI].

El ensamblador genera el tipo de desplazamiento más apropiado (0, 8 ó 16 bits) dependiendo del valor que tenga la constante: si vale cero se utiliza el primer caso, si vale entre -128 y 127 se utiliza el segundo, y en otro caso se utiliza el tercero. Nótese que [BP] sin desplazamiento no existe. Al ensamblar una instrucción como, por ejemplo, *MOV AL,[BP]*, se generará un desplazamiento de 8 bits con valor cero. Esta instrucción ocupa tres bytes, mientras que *MOV AL,[SI]* ocupa dos, porque no necesita el desplazamiento.

Estos modos de direccionamiento producen algunos inconvenientes en el 8086/8088. La CPU gasta tiempo calculando una dirección compuesta de varias cantidades. Principalmente esto se debe al hecho de que el cálculo de direcciones está programado en microcódigo (dentro de la CROM del sistema de control de la unidad de ejecución). En las siguientes versiones (a partir del 80186/80188) estos cálculos están cableados en la máquina y, por lo tanto, cuesta mucho menos tiempo el realizarlos.

Veamos un ejemplo: *MOV AL, ES:[BX+SI+6]*. En este caso el operando de la izquierda tiene direccionamiento a registro mientras que el de la derecha indica una posición de memoria. Poniendo valores numéricos, supongamos que los valores actuales de los registros sean: ES = 3200h, BX = 200h, SI = 38h. Como se apuntó más arriba la dirección real de memoria será:

$$ES * 10h + BX + SI + 6 = 3200h * 10h + 200h + 38h + 6 = 3223Eh$$

Estructura de memoria de segmentación: Como se ha mencionado anteriormente, el 8086/8088 usa un esquema ingenioso llamado segmentación, para acceder correctamente a un megabyte completo de memoria, con referencias de direcciones de sólo 16 bits.

Veamos cómo funciona. Cualquier dirección tiene dos partes, cada una de las cuales es una cantidad de 16 bits. Una parte es la dirección de segmento y la otra es el offset. A su vez el offset se compone de varias partes: un desplazamiento (un número fijo), una base (almacenada en el registro base) y un índice (almacenado en el registro índice). La dirección de segmento se almacena en uno de los cuatro registros de segmento (CS, DS, ES, SS). El procesador usa estas dos cantidades de 16 bits para calcular la dirección real de 20 bits, según la siguiente fórmula:

Dirección real = 16 * (dirección del segmento) + offset

Tal como veíamos antes, dado que 16 en decimal es 10 en hexadecimal, multiplicar por ese valor es lo mismo que correr el número hexadecimal a la izquierda una posición.

Hay dos registros de segmento que tienen usos especiales: el microprocesador utiliza el registro CS (con el offset almacenado en el puntero de instrucción IP) cada vez que se debe acceder a un byte de instrucción de programa, mientras que las instrucciones que utilizan la pila (llamados a procedimientos, retornos, interrupciones y las instrucciones PUSH y POP) siempre utilizan el registro de segmento SS (con el offset almacenado en el registro puntero de pila SP). De ahí los nombres que toman: CS es el segmento de código mientras que SS es el registro segmento de pila.

Para acceder a datos en la memoria se puede utilizar cualquiera de los cuatro registros de segmento, pero uno de ellos provoca que la instrucción ocupe un byte menos de memoria: es el llamado segmento por defecto, por lo que en lo posible hay que tratar de usar dicho segmento para direccionar datos. Este segmento es el DS (registro de segmento de datos) para todos los casos excepto cuando se utiliza el registro base BP. En este caso el segmento por defecto es SS.

Si se utiliza otro registro, el ensamblador genera un byte de prefijo correspondiente al segmento antes de la instrucción: CS → 2Eh, DS → 3Eh, ES → 26h y SS → 36h. El uso de estos diferentes segmentos significa que hay áreas de trabajo separadas para el programa, pila y los datos. Cada área tiene un tamaño máximo de 64 KBytes. Dado que hay cuatro registros de segmento, uno de programa (CS), uno de pila (SS) y dos de datos (segmento de datos DS y segmento extra ES) el área de trabajo puede llegar a 4 * 64 KB = 256 KB en un momento dado suponiendo que las áreas no se superponen.

Si el programa y los datos ocupan menos de 64 KB, lo que se hace es fijar los registros de segmento al principio del programa y luego se utilizan diferentes offsets para acceder a distintas posiciones de memoria. En caso contrario necesariamente deberán cambiarse los registros de segmento en la parte del programa que lo requiera. Los registros de segmento DS, ES y SS se cargan mediante las instrucciones MOV y POP, mientras que CS se carga mediante transferencias de control (saltos, llamadas, retornos, interrupciones) intersegmento.

Estructura de interrupciones del 8086/8088

Hay tres clases de interrupción: por hardware, por software e internas (a las dos últimas también se las llama "excepciones").

Veremos primeramente el caso de interrupciones por hardware: Como se mencionó anteriormente, el 8086/8088 tiene dos entradas de petición de interrupción: NMI e INTR y una de reconocimiento (INTA). La gran mayoría de las fuentes de interrupción se conectan al pin INTR, ya que esto permite enmascarar las interrupciones (el NMI no). Para facilitar esta conexión, se utiliza el circuito integrado controlador de interrupciones, que tiene el código 8259A. Este chip tiene, entre otras cosas, ocho patas para sendas fuentes de interrupción (IRQ0 – IRQ7), ocho para el bus de datos (D0 – D7), una salida de INTR y una entrada de INTA. Esto permite una conexión directa con el 8088/8086. Al ocurrir una petición de alguna de las ocho fuentes, el 8259A activa la pata INTR. Al terminar de ejecutar la instrucción en curso, el microprocesador activa la pata INTA, lo que provoca que el 8259A envíe por el bus de datos un número de ocho bits (de 0 a 255) llamado tipo de interrupción (programable por el usuario durante la inicialización del 8259A), que el 8086/8088 utiliza para saber cuál es la fuente de interrupción. A continuación busca en la tabla de vectores de interrupción la dirección del manejador de interrupción (*interrupt handler*). Esto se hace de la siguiente manera. Se multiplica el tipo de interrupción por cuatro, y se toman los cuatro bytes que se encuentran a partir de esa dirección. Los dos primeros indican el offset y los dos últimos el segmento del manejador, como se muestra a continuación.