Arquitectura Interna del 8088

Intel diseñó el 8088/8086 para realizar al mismo tiempo las principales funciones internas de transferencia de datos y búsqueda de instrucciones. Para conseguir esto, el 8088 y el 8086 constan de dos procesadores interconectados en la mismo circuito integrado, ver figura 1. Una unidad esta encargada de buscar instrucciones (BIU) y la otra de ejecutarlas (EU) esto diferencia al 8088 de los microprocesadores anteriores (8080, 8085).

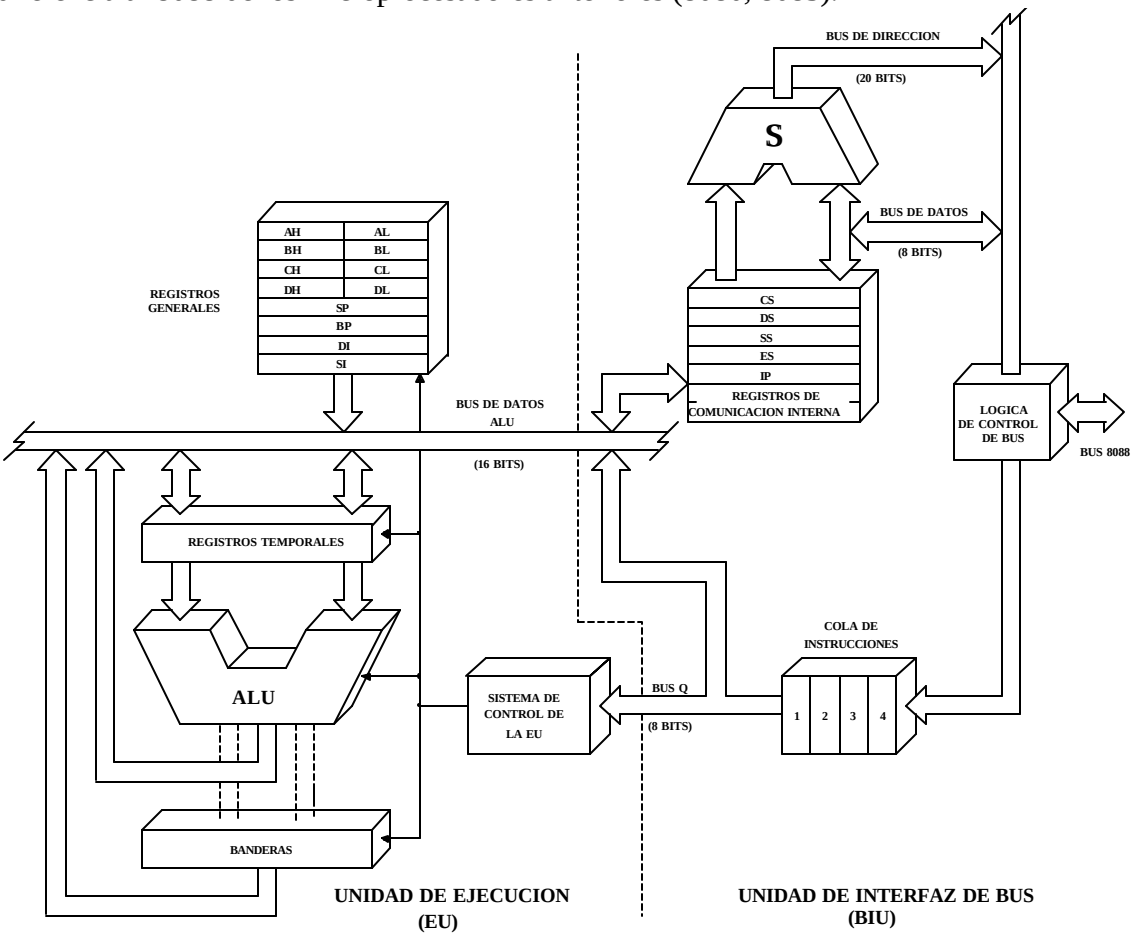

Figura 1. Arquitectura interna del 8088

En la figura 2 se ilustra una comparación operacional del 8085 con el 8088, la figura 2 (a) muestra una operación normal del 8085. Note que la instrucción es traída de la memoria por una operación de lectura de memoria, luego el 8085 ejecuta la instrucción, y el sistema de memoria-bus esta ocioso. El 8088 hace uso de este tiempo de memoria-bus ocioso para buscar la próxima instrucción mientras se esta ejecutando la actual instrucción.

La figura 2 (b) muestra la secuencia de eventos para el 8088. Note que el bus siempre esta ocupado. (Generalmente, si existen momentos en que el bus esta ocioso pero no siempre). Esto es debido a las dos unidades que componen al 8088 la Unidad de Ejecución (EU) y la Unidad de Interfaz de Bus (BIU).

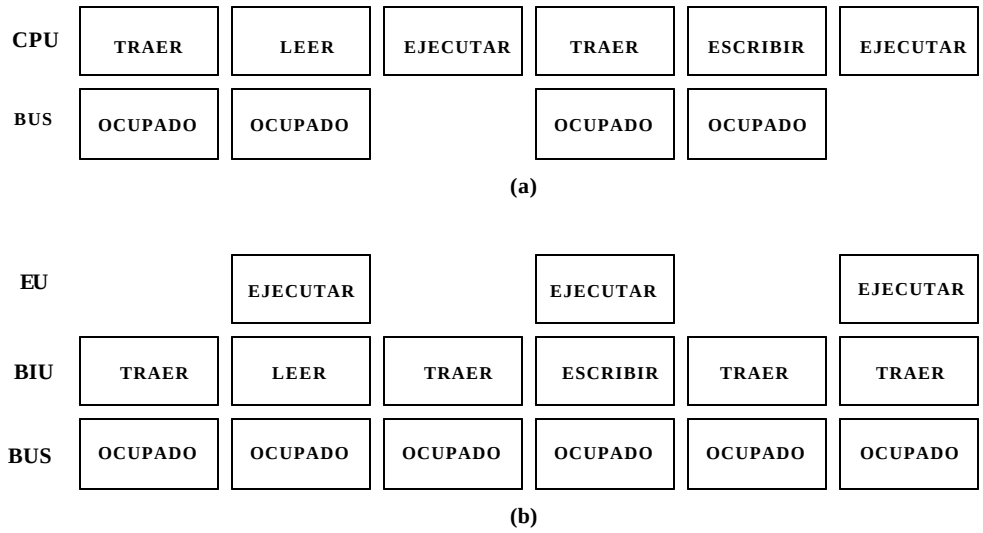

Figura 2. (a) Operación del 8085 y actividad del bus (b) Operación de las unidades del 8088 y actividad del bus.

Unidad de Interfaz de Bus (BIU): La BIU contiene una *cola de instrucciones*, un *controlador de bus*, *registros de segmento* y el *puntero de instrucción* (IP). La principal función de la BIU es mantener llena la cola de instrucciones , generar y aceptar señales de control, proveer al sistema de direcciones de memoria y número de puerto de E/S ademas de ser el mediador entre la Unidad de Ejecución (EU) y la memoria.

La BIU asegura que la cola de instrucciones este llena mediante la operación de traer la próxima instrucción de un byte si la cola de instrucciones tiene espacio. Debido a que la próxima instrucción a ejecutar esta dentro del microprocesador, la velocidad de ejecución de programas es mucha mas rápida en comparación a si cada instrucción a ejecutar fuese traída directamente de memoria.

Unidad de Ejecución (EU): La función de la EU es sacar cada instrucción de la cola de instrucciones y ejecutarla. La unidad de ejecución contiene una *unidad aritmética y lógica* (ALU), un *registro de instrucción* y una *arreglo de registros*. La ALU realiza operaciones aritméticas y lógicas sobre la memoria o sobre registros. El registro de instrucciones recibe instrucciones de la cola de instrucción y son decodificadas a operaciones directas para la unidad de ejecución. El arreglo de registros mantiene información temporalmente. También contiene registros índices y punteros utilizados para direccionar el dato operando localizado en la memoria.

Conjunto de Registros del 8088

El 8088 contiene 14 registros de 16 bits que se asocian a tres grupos: *Registro de Propósito General*, *Registros Punteros* y *de Índice* y *Registros de Segmentos*. Ademas de estos tres grupos de registros contiene un *Registro de Banderas* que indica el estado de la operación de la unidad aritmética y lógica (ALU).

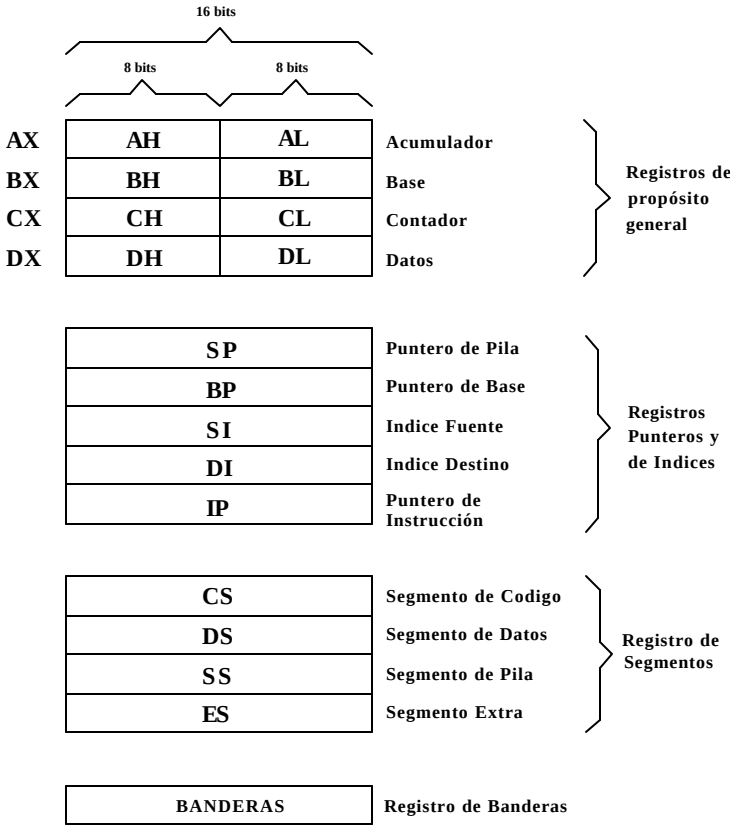

Figura 3. Conjunto de Registros del 8088.

Registros de Propósito General

Los registros de propósito general se utilizan de cualquier manera que el programador desee (siempre y cuando sea permitido). Estos registros se utilizan como registros de 16 bits (AX, BX, CX y DX) o como dos registros de 8 bits (AH, AL, BH, BL, CH, CL, DH y DL). La principal función de la registros de propósito general de describe a continuación:

AX (*Acumulador*)-Generalmente utilizado para mantener temporalmente resultados después de una operación aritmético o lógica .

BX (*Base*)- Generalmente utilizado para mantener la dirección base de un dato localizado en la memoria y también la dirección base de una tabla de datos referenciada por una instrucción (XLAT).

CX (*Contador*)-Contador en ciertas instrucciones, tal como contador de corrimientos (CL) y rotaciones, contador (CX) con la instrucción LOOP.

DX (*Dato*)-Un registro de propósito general que mantiene la parte mas significativos del producto después de una multiplicación de 16 bits, también los bits mas significativos del dividendo antes de una división, y el número del puerto de E/S en una instrucción de E/S.

Registros Punteros y de Indice

Aunque los registros punteros y de índice también son de propósito general por naturaleza, estos son mas utilizados como índice o punteros a una localidad de memoria en muchas instrucciones. Estos registro incluyen:

SP (*Puntero de Pila*)-Usado para direccionar datos de una pila de memoria, esta pila es de tipo LIFO (last-in, first-out). Esto ocurre cuando una instrucción PUSH o POP es ejecutada o cuando una subrutina es llamada mediante CALL y al retornar de una subrutina con la instrucción RET.

BP (*Puntero de Base*)-Un puntero de propósito general utilizado para direccionar un arreglo de datos en la pila.

SI (*Índice Fuente*)-Se utiliza para direccionar datos fuentes indirectamente mediante el uso de instrucciones con cadenas.

DI (*Índice Destino*)-Normalmente utilizado para direccionar el destino de datos indirectamente para instrucciones con cadenas.

IP (*Puntero de Instrucción*)-Usado para direccionar el próxima instrucción a ejecutar por el 8088. La localidad actual de la próxima instrucción esta formada por el contenido de IP y CS×10H.

Registros de Segmentos

Un segmento de memoria es un bloque de 64K-bytes de memoria direccionado por un registro especial llamado *registro de segmento,* la figura 4 muestra segmentos de memoria del 8088. Cuatro segmentos diferentes pueden existir simultáneamente en el espacio de memoria: el *segmento de código, segmento de datos, segmento de pila* y *segmento extra*. Los datos son indexados o apuntados dentro de un segmento por el *registros de índice, registros punteros, registro base* o *puntero de instrucción*.

Cada registro de segmento mantiene una porción de 16-bits de la dirección de 20- bit de inicio del segmento de 64K-bytes de memoria. Los 20 bits de la dirección esta formada 0000_2 (0H) colocado como el menos significativo del registro de segmento, esto equivale a multiplicar el registro segmento por 10H.

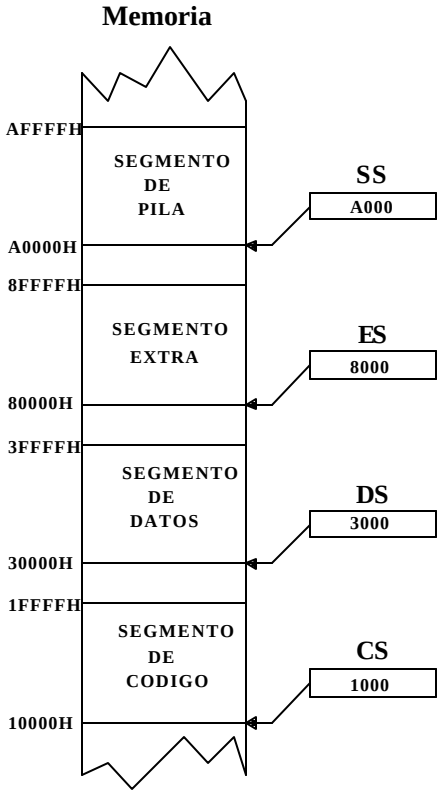

Figura 4. Ejemplo que muestra el contenido de cada registro de segmento y la localidad de la dirección del segmento direccionada por el registro de segmento.

Cada registro de segmento tiene una función especial y normalmente es asociada con uno o mas de los registros de índice o punteros. Para generar una localidad de memoria, el contenido del registro de segmento contiene la parte de la dirección del segmento y se incrementa un registro índice o puntero, el cual contiene el desplazamiento (*offset*). La figura 5 muestra una dirección de memoria generada dentro de un segmento por la combinación del contenido del registro de segmento y el desplazamiento almacenado en un registro índice o puntero y así generar la *dirección efectiva.* La dirección efectiva es la suma de la *dirección del segmento* y el *desplazamiento.* En el ejemplo de la figura 5, el registro se segmento (DS) contiene 1000H, así que el segmento inicia en la localidad de memoria 10000H y el desplazamiento (0010H) que esta contenido en el registro base (BX), forman la dirección efectiva 10010H, o 1000H $\times$ 10H + 0010H.

A continuación de describen cada uno de los segmento de memoria (**no registros de segmento**) manejados por el 8088.

CS (*Segmento de Código*)-Una sección de memoria de 64K-bytes que contiene el programa o código. Este registro es cambiado siempre al ejecutarse una instrucción CALL, JMP o RET. La dirección de la próxima instrucción a ejecutar es generada por la suma del contenido del puntero de instrucción y el contenido de CS×10H.

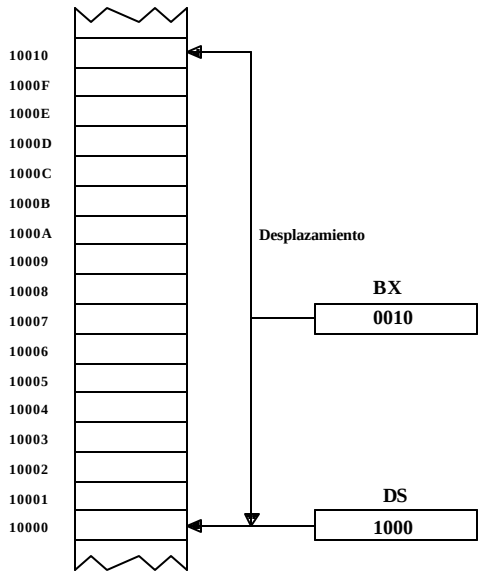

Figura 5. Mapa de memoria que muestra como la dirección 10010H es direccionada cuando DS=1000H y BX contiene un desplazamiento de 0010H.

DS (*Segmento de Datos*)-Una sección de memoria de 64K-bytes que contiene los datos direccionados por todas las instrucciones y modos de direccionamiento. Los datos generalmente son movidos dentro o fuera de la memoria mediante el segmento de datos. La dirección efectiva de un dato es generada por la suma del contenido de uno de los registros índice o puntero (BX, SI o DI) y el contenido de DS×10H.

SS (*Segmento de Pila*)-Una sección de memoria de 64K-bytes usada pora la pila (stack) tipo LIFO. La dirección efectiva de la pila es una combinación de los contenidos del puntero de pila (SP) mas SS×10H. Por ejemplo, si SS contiene 1000H y SP contiene 0000H, entonces la dirección de la pila es 10000H. Esta dirección también es escrita como 1000:0000 que es: una dirección de segmento 1000H y un desplazamiento de 0000H. Los datos apuntados por la base (BP) normalmente se encuentran en el segmento de pila.

ES (*Segmento Extra*)-Un segmento especial que normalmente se usa por instrucciones de cadenas. Cuando una instrucción de cadena es ejecutada, la localidad del destino es direccionada por el registro índice destino (DI) mas ES×10H, y la dirección fuente esta direccionada por el registro índice fuente (SI) mas DS×10H.

Registro de Banderas

El *registro de banderas*, también llamado registro de estado o palabra de estado, es un registro de 16 bits y contiene información sobre el estado de operaciones de la unidad aritmética y lógica, en la figura 6 muestra la posición relativa de cada bit de bandera y su correspondiente letra que se utiliza para identificar cada bandera.

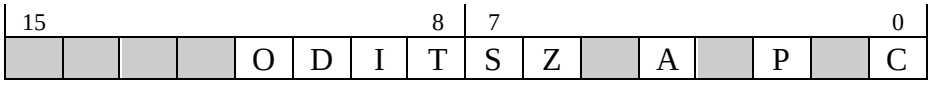

Figura 6. Esquema del Registro de Banderas

Descripción de los bits de bandera:

C (*Acarreo*)-Indica un acarreo o un préstamo en el bit mas significativo después de una operación aritmética. Esta bandera también se modifica por algunas instrucciones de corrimiento y rotación.

P (*Paridad*)-Se refiere a la paridad del resultado de una operación aritmética o lógica. Si es resultado contiene un número par de unos, el bit de paridad es puesto en 1 indicando paridad par; en otro caso se borra para indicar paridad impar.

A (*Acarreo Auxiliar*)-Representa un acarreo o préstamo entre medio-bytes de una operación aritmética o lógica entre registros de 8 bits.

Z (*Cero*)-Indica que el resultado de una operación aritmética o lógica es cero (0). Si Z=1, el resultado es 0.

S (*Signo*)-Indica el signo del resultado de una operación aritmética o lógica. Un 1 lógico en la bandera de signo índica que el resultado es negativo.

T (*Atrapar*)-Causa que el 8088 entre a un estado de operación paso a paso para depuración.

I (*Habilitar Interrupciones*)-Habilita o deshabilita la terminal INTR (requerimiento de interrupción). Si I=1 entonces INTR esta habilitada.

D (*Dirección*)-Selección el modo de operación de auto-incremento o auto-decremento para el registro índice destino (ID) y el registro índice fuente (SI) en operaciones de cadena. Si D=0, entonces SI y DI son incrementados durante la ejecución de una instrucción de cadena.

O (*Sobreflujo*)-Se activa después de que en una operación aritmética de suma o resta ha ocurrido un sobreflujo. Por ejemplo, si 7FH (+127) y 01H (+1) se suman, y se esta operando con números con signo, el resultado es 80H (-128). Debido a que -128 no es un resultado correcto, la bandera O es puesta en 1 lógico para indicar un sobrefujo.