Material de apoyo
Registros de la memoria

CONCEPTOS

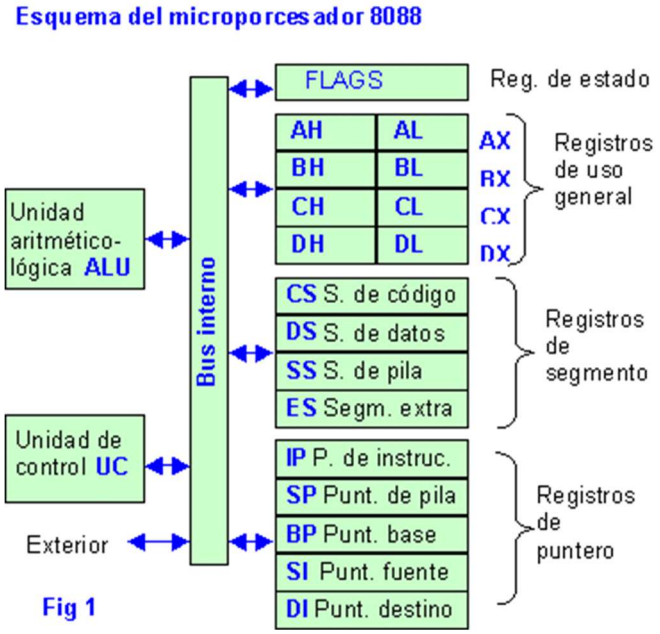
Para poder comprender el funcionamiento de la memoria se ha incluido en este documento información referente a dicho apartado, especificando la función que cumple cada uno de los registros de la memoria en arquitecturas de 32 bits.

Stack

Consiste en una región de la memoria virtual que usa una estructura de datos de tipo LiFo (Last input, First output), en el stack se almacenan las variables, excepciones, argumentos, etc. Crece desde lo más alto hacia lo más bajo.

REGISTROS DE LA MEMORIA

Consisten en una memoria ubicada en el procesador (el nivel más alto de la jerarquía de memoria), contando con una velocidad muy alta, pero por el contrario con una capacidad de almacenamiento muy baja (puede ir desde los 4 bits a los 64 bits en las arquitecturas actuales de 64 bits, en el caso de la arquitectura de 32 bits su tamaño es de 32 bits).



Por compatibilidad todos los registros de 16 bits funcionan de forma semejante a los de 32 bits, la única diferencia es que tienen de prefijo la letra E, por ejemplo, IP pasa a ser EIP. En 64 bits ha ocurrido algo semejante para poder tener compatibilidad con los registros de 32 bits, usando de prefijo la letra R, por ejemplo, EIP pasa a ser RIP.

Registros de 32 bits: EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP, EIP

Registros de 16 bits: AX, BX, CX, DX, SI, DI, BP, SP, IP

Registros de 8 bits: AH, AL, BH, BL, CH, CL, DH, DL

Registros GPR

Se utilizan para propósitos generales.

- **AX, AH, AL (Acumulador)**: Conserva el resultado temporal tras realizar una operación aritmética o lógica. EAX es el ejemplo más visto en las arquitecturas de 32 bits.
- **BX, BH, BL (Base)**: Almacena la dirección "base" de listas de datos en la memoria. EBX es muy visto en arquitecturas de 32 bits.
- **CX, CH, CL (Contador)**: Realiza las funciones de contador para algunas instrucciones (bucles, por ejemplo). ECX es el ejemplo más visto en las arquitecturas de 32 bits.
- **DX, DH, DL (Datos)**: Se usan para almacenar la parte más significativa de una operación tras terminar una multiplicación o antes de realizar una división. EDX es el ejemplo más usado en arquitecturas de 32 bits.

Registros apuntadores y de índice

- **SP:** Se trata del final de la pila de memoria. Tras ejecutarse una función la dirección de retorno vuelve a cargarse en el ESP para continuar con la ejecución desde donde se ha quedado. ESP es el registro en 32 bits.
- **BP:** Se puede usar como registro de tipo general o como un puntero al marco de la pila de memoria. EBP es el registro en 32 bits.
- **DI:** Es el índice de destino, siempre apunta al destino de los datos. EDI es el registro en 32 bits.
- **SI:** Se usa para apuntar funciones que requieren un origen con los datos que se van a usar, SI almacena el origen de esos datos. ESI es el registro en 32 bits.
- **IP:** Contiene la dirección de la siguiente instrucción que se ejecutará en el microprocesador. EIP es el registro en 32 bits.

Registros de segmento

- **CS (Código):** Contiene la dirección lógica del segmento en el que está actualmente el código de un programa, si el código ocupase más de un segmento contiene la dirección lógica de uno de esos segmentos. ECS es el registro en 32 bits.

- **DS (Datos):** Tiene la dirección lógica del segmento donde están los datos "estáticos" de un programa. Si los datos ocupasen más de un segmento contiene la dirección lógica de uno de esos segmentos. EDS es el registro en 32 bits.
- **ES (Extra):** Contiene la dirección lógica de uno de los segmentos en los que almacena datos "estáticos" de un programa, pero se usa para el manejo de cadenas.
- **SS (Pila):** Contiene la dirección lógica del segmento en el que se encuentra actualmente la pila del sistema. La pila no puede ser mayor a un segmento.

Registros de control

Estos registros se utilizan para controlar las operaciones del procesador. La mayoría no son visibles al usuario y pueden ser accesibles a las instrucciones ejecutadas en un modo de "control".

- **MAR:** Contiene la dirección donde se va a efectuar la próxima lectura/escritura de datos. El número de direcciones depende del registro MAR.
- **MBR:** Contiene los datos que van a ser escritos en la memoria, o que ya han sido leídos de la memoria.
- **Registro de direcciones de entrada y salida (Input/Output AR):** Indica el dispositivo (sea de entrada o salida, un teclado, un monitor).
- **Registro de datos de entrada y salida (Input/Output BR):** Zona temporal donde se va a realizar el intercambio de datos entre el procesador y el dispositivo de entrada y salida que se especificó en el registro de direcciones de entrada y salida (sea de entrada o salida, un teclado, un monitor).
- **Registro de instrucciones (IR):** Contiene la dirección de la siguiente instrucción que se va a ejecutar.
- **Word de estado del programa (PSW):** Tiene códigos de condición además de otro tipo de información de estado como el signo, desbordamiento, etc.

FLAGS

Se tratan de indicadores de estado.

- **0 = CF (Carry Flag)** Indica a la CPU cuando hacer cálculos aritméticos con el punto lógico significativo. Usa un único bit por lo que 0 desactivada dicha opción, 1 activada.
- **2 = PF (Parity Flag)** Comprueba si en la última operación realizada el número de bits (1 en la secuencia) se corresponden a un número par o un número impar. 11010 la bandera de paridad sería 0 ya que existen 3 bits (1) en la secuencia, 01010 sería 1 ya que existe un numero par de bits, dos en total.
- **4 = AF (Adjust Flag)** Indica si la operación de un Carry o Borrow ocurre en los 4 bits significativos menores, se usa principalmente con códigos decimales aritméticos.
- **6 = ZF (Zero Flag)** Cuando el resultado de una operación es 0. Por ejemplo, operaciones que no se almacenan en los resultados (comparaciones, tests de bits).

- **7 = SF (Sign Flag)** Se utiliza cuando la operación resulta en un número negativo (cuando el bit más significativo estaba establecido).
- **8 = TF (Trap Flag)** Causa una interrupción del paso tras cada instrucción ejecutada.
- **9 = IF (Interrupt enable Flag)** Define cuando el procesador debe de reaccionar o no reaccionar a cada interrupción entrante.
- **10 = DF (Direction Flag)** Controla la dirección de las operaciones con cadenas. La operación se realiza desde la dirección más baja a la más alta si el bit está a 0. Y de la más alta a la más baja si está en 1.
- **11 = OF (Overflow Flag)** Suele ser uno de los complementos a la carry flag, se establece cuando el resultado de la operación es demasiado pequeña o demasiado grande para poder "caber" en la operación de destino.

Hack by Security

We attack for your safety