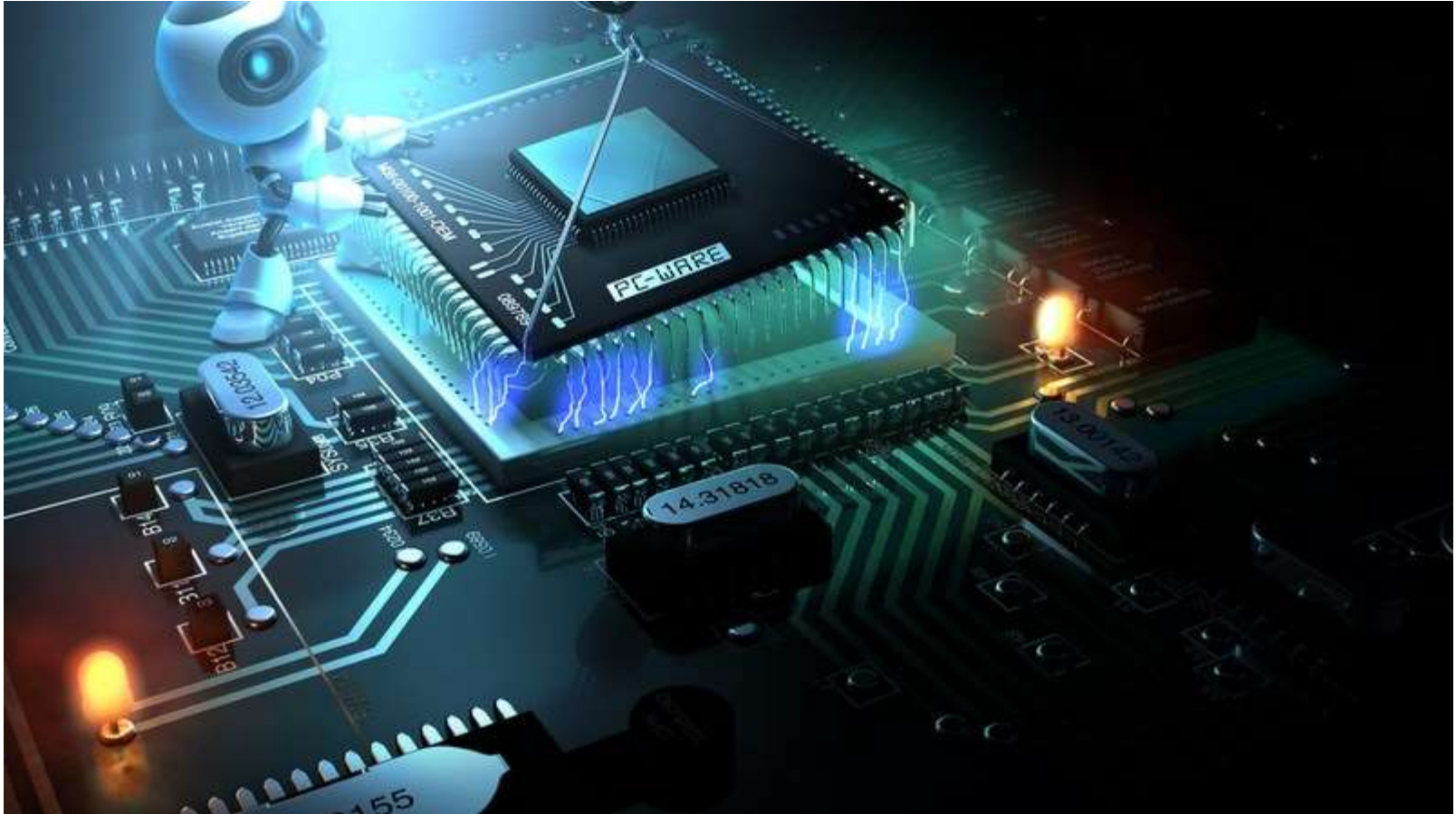


LENGUAJE ENSAMBLADOR

(7°A-ICI)



Mtro. en Ing. Armando Álvarez Fdez.
Ago-2024

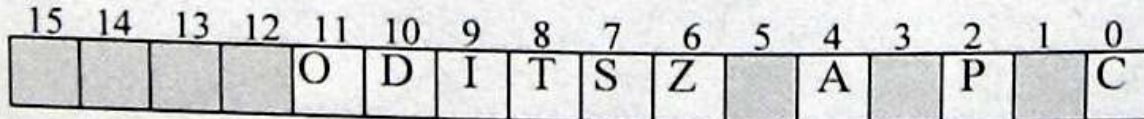
REGISTRO DE BANDERAS DEL 8086

REGISTRO DE BANDERAS DEL 8086

(También conocido como Registro de Estatus del CPU 8086)

4. Banderas de Registro Status (El registro de banderas).

Las banderas indican la condición del microprocesador a la vez que controlan su funcionamiento.



REGISTRO DE BANDERAS DEL 8086

Registro de flags



C : acarreo en la suma y arrastre en la resta

P : paridad del dato (0, impar y 1, par)

A : acarreo auxiliar. Indica el acarreo o arrastre entre los bits 3 y 4

Z : indicación de resultado igual a cero

S : indicador de signo del resultado. 0, positivo y 1, negativo

T : trampa. Habilita la característica de depuración del procesador

I : habilitación de interrupciones de hardware

D : selección de incremento o decremento en los índices

O : sobreflujo.

EXPLICACIÓN DE LAS BANDERAS

C (Acarreo): Indica un acarreo después de una suma o un préstamo después de una resta. La bandera de acarreo también indica condiciones de error en ciertos programas y procedimientos. También se utiliza en algunas instrucciones de rotación y desplazamiento.

P (Paridad): Es un cero para la paridad impar y un uno para la paridad par. La paridad es un conteo de “unos” expresado en un número par o impar. Por ejemplo, si un número contiene 3 bits con uno binario, tiene paridad impar. Si un número contiene cero bits de uno se considera que paridad par.

A (Acarreo Auxiliar): Indica un acarreo después de una suma o un préstamo después de una resta entre las posiciones de los bits 3 y 4 en el resultado. Este indicador muy especializado se prueba con las instrucciones DAA y DAS para ajustar el valor de AL después de una suma o resta BCD. El microprocesador, no utiliza en otra forma el bit de bandera A.

Z (Zero): Indica que el resultado de una operación aritmética o lógica es cero. Si $Z=1$, el resultado es cero, si $Z=0$ entonces el resultado no es cero.

S (Signo): Indica el signo aritmético del resultado después de una suma o resta. Si $S=1$ la bandera de signo se activa y el resultado es negativo. Si $S=0$ la bandera de signo se desactiva y el resultado es positivo. Se debe tener en cuenta que el valor de la posición del bit más significativo se coloca en el bit de signo para cualquier instrucción que afecte las banderas.

T (Trampa): Cuando se activa la bandera de trampa se habilita la característica de depuración del microprocesador. Mas adelante aparecen mayores detalles de esta característica.

I (Interrupción): Controla el funcionamiento de la terminal de la entrada de INTR (*Interrupción externa*). Si $I=1$ se habilita la interrupción y si $I=0$ se deshabilita la entrada INTR.

D (Direccion): Controla la selección de incremento o decremento de los registros DI o SI durante las instrucciones de cadenas o arreglos. Se utiliza para indicar si las operaciones con string se ejecutaran en forma ascendente o descendente.

O (Sobreflujo): Es una condición que ocurre cuando se suman o restan números con signo. Un sobreflujo indica que el resultado ha excedido de la capacidad de la maquina. Por ejemplo, si se suma un 7FH (+127) a 01H (+1) el resultado es 80H (-128). Este resultado representa una situación de sobreflujo señalado por la bandera para la suma con signo. Para operaciones sin signo no se toma en cuenta esta bandera.

NOTAS ADICIONALES SOBRE LAS BANDERAS DEL 8086

Flags

The 8086 and 8088 have six 1-bit status flags (figure 2-9) that the EU posts to reflect certain properties of the result of an arithmetic or logic

operation. A group of instructions is available that allows a program to alter its execution depending on the state of these flags, that is, on the result of a prior operation. Different instructions affect the status flags differently; in general, however, the flags reflect the following conditions:

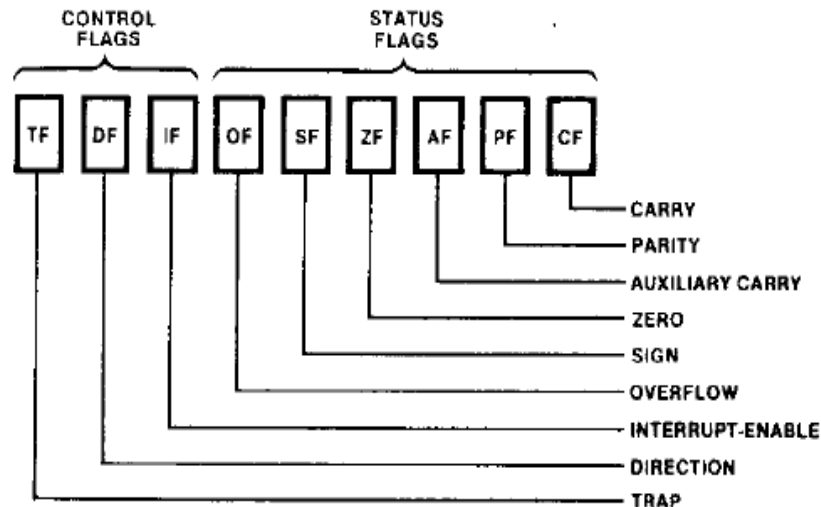


Figure 2-9. Flags

1. If AF (the auxiliary carry flag) is set, there has been a carry out of the low nibble into the high nibble or a borrow from the high nibble into the low nibble of an 8-bit quantity (low-order byte of a 16-bit quantity). This flag is used by decimal arithmetic instructions.
2. If CF (the carry flag) is set, there has been a carry out of, or a borrow into, the high-order bit of the result (8- or 16-bit). The flag is used by instructions that add and subtract multibyte numbers. Rotate instructions can also isolate a bit in memory or a register by placing it in the carry flag.

NOTAS ADICIONALES SOBRE LAS BANDERAS DEL 8086

3. If OF (the overflow flag) is set, an arithmetic overflow has occurred; that is, a significant digit has been lost because the size of the result exceeded the capacity of its destination location. An Interrupt On Overflow instruction is available that will generate an interrupt in this situation.
4. If SF (the sign flag) is set, the high-order bit of the result is a 1. Since negative binary numbers are represented in the 8086 and 8088 in standard two's complement notation, SF indicates the sign of the result (0 = positive, 1 = negative).
5. If PF (the parity flag) is set, the result has even parity, an even number of 1-bits. This flag can be used to check for data transmission errors.
6. If ZF (the zero flag) is set, the result of the operation is 0.

Three additional control flags (figure 2-9) can be set and cleared by programs to alter processor operations:

1. Setting DF (the direction flag) causes string instructions to auto-decrement; that is, to process strings from high addresses to low addresses, or from "right to left." Clearing DF causes string instructions to auto-increment, or to process strings from "left to right."
2. Setting IF (the interrupt-enable flag) allows the CPU to recognize external (maskable) interrupt requests. Clearing IF disables these interrupts. IF has no affect on either non-maskable external or internally generated interrupts.
3. Setting TF (the trap flag) puts the processor into single-step mode for debugging. In this mode, the CPU automatically generates an internal interrupt after each instruction, allowing a program to be inspected as it executes instruction by instruction. Section 2.10 contains an example showing the use of TF in a single-step and breakpoint routine.

VISTA GENERAL DEL SET DE INSTRUCCIONES DEL 8086

6. Juego de instrucciones.

Las categorías de instrucciones descritas en esta sección incluyen: Transferencia de datos, aritméticas, manipulación de bits, cadenas o arreglos, transferencia de programa, etc.

Transferencia de datos: Incluye instrucciones para transferencia de datos que transfieren bytes, palabras o dobles palabras de datos entre la memoria y los registros así como entre el acumulador y los puertos de E/S.

| Código Operación | Función |
|------------------|---|
| IN | Mete datos al acumulador desde un dispositivo de E/S. |
| LAHF | Carga banderas en AH. |
| LEA | Carga la dirección efectiva. |
| LDS | Carga DS y registro de 16 bits con los datos de memoria de 32 bits. |
| LES | Carga ES y registro de 16 bits con los datos de memoria de 32 bits. |
| MOV | Carga byte, palabra o doble palabra. |
| OUT | Saca datos del acumulador a un E/S. |
| POP | Recupera una palabra de la pila. |
| POPF | Recupera los indicadores de la pila. |
| PUSH | Salva las palabras en la pila. |
| PUSHF | Salva banderas en la pila. |
| SAHF | Carga AH en las banderas. |
| XCHG | Intercambia bytes, palabras o dobles palabras. |
| XLAT | Emplea AL para entrar a una tabla de conversión. |

Aritmética: Instrucciones para sumar, restar, multiplicar y dividir datos como bytes, palabras o dobles palabras. El sistema suma y resta con el empleo de datos con signo o sin signo y datos BCD o de ASCII. Multiplica y divide números ASCII con signo o sin signo.

| Código Operación | Función |
|-------------------------|---|
| AAA | Ajuste ASCII para la suma. |
| AAD | Ajuste ASCII para la división. |
| AAM | Ajuste ASCII para la multiplicación. |
| AAS | Ajuste ASCII para la resta. |
| ADD | Suma datos entre registros o la memoria y otro registro. |
| ADC | Suma datos con la bandera de acarreo. |
| CBW | Convierte byte a palabra. |
| CMP | Compara los datos. |
| CWD | Convierte palabra a doble palabra. |
| DAA | Ajuste decimal de AL después de una suma de BCD: |
| DAS | Ajuste decimal de AL después de una resta de BCD: |
| DEC | Decremento. |
| DIV | División sin signo. |
| IDIV | División con signo. |
| IMUL | Multiplicación con signo. |
| INC | Incrementa. |
| MUL | Multiplicación sin signo. |
| NEG | Cambia el signo (complemento a dos). |
| SBB | Resta con acarreo. |
| SUB | Resta datos entre los registros y la memoria u otro registro. |

Manipulación de bits: Se utilizan para controlar los datos hasta el nivel bits. Estas instrucciones incluyen operaciones lógicas, corrimientos y rotaciones.

| Código Operación | Función |
|------------------|--|
| AND | Y lógica. |
| NOT | Invertir (complemento a uno). |
| OR | O lógica. <i>inclusive</i> |
| SAR | Corrimiento aritmético a la derecha. |
| SHL/SAL | Corrimiento a la izquierda. |
| SHR | Corrimiento lógico a la derecha. |
| RCL | Rotación a la izquierda con acarreo. |
| ROL | Rotación a la izquierda. |
| RCR | Rotación a la derecha con acarreo. |
| ROR | Rotación a la derecha. |
| TEST | Operación con el AND lógico, pero solo afectando banderas. |
| XOR | O exclusivo. |

Instrucciones para cadenas: Se emplean para manipular cadenas de datos en la memoria. Cada cadena consta, ya sea de bytes o de palabras y tiene hasta 64K bytes de longitud.

| Código Operación | Función |
|-------------------------|---|
| CMPS | Comparación entre memoria y memoria. |
| LODS | Cargar el acumulador. |
| MOVS | Mover de memoria a memoria. |
| SCAS | Comparación entre la memoria y el acumulador. |
| STOS | Almacenar en el acumulador. |

Transferencia de programa: Incluyen brincos, llamadas (CALL) y para retorno.

Saltos basados en datos sin signo.

| | |
|------------------|--|
| JE / JZ | Jump if equal / jump if zero |
| JNE / JNZ | Jump if non equal / jump if non zero |
| JA / JNBE | Jump if above / jump if no below or equal |
| JAE / JNB | Jump if above or equal / jump if not below |
| JB / JNAE | Jump if below / jump if not above or equal |
| JBE / JNA | Jump if below or equal / jump if not above |

Saltos basados en datos con signo

| | |
|------------------|---|
| JE / JZ | Jump if equal / jump if zero |
| JNE / JNZ | Jump if non equal / jump if non zero |
| JG / JNLE | Jump if greater / jump if non less or equal |
| JGE / JNL | Jump if greater or equal / jump if non less |
| JL / JNGE | Jump if less / jump if not greater or equal |
| JLE / JNG | Jump if less or equal / jump if not greater |

Pruebas aritméticas especiales.

| | |
|------------------|---|
| JS | Jump if sign |
| JNS | Jump if not sign |
| JC | Jump if carry |
| JNC | Jump if not carry |
| JO | Jump if overflow |
| JNO | Jump if not overflow |
| JP / JPE | Jump if parity / jump if parity even |
| JNP / JPO | Jump if not parity / jump if parity odd |

5. Modos de direccionamiento.

INTRODUCCION A LOS

Modos de direccionamiento

(Primera aproximación de clasificación)

Modos de direccionamiento. El microprocesador 8086 tiene 25 modos de direccionamiento. El modo de direccionamiento indica la forma en que ha de calcularse la dirección efectiva de los operandos de la instrucción. Los modos de direccionamiento son complicados como para explicarse en detalle cada uno de ellos sin embargo pueden ser clasificados como sigue:

- Directo.
- Segmentado.
- Base.
- Indexado.
- Con offset.

Estos modos de direccionamiento pueden ser combinados de tal forma que conforman así los 25 modos de direccionamiento.

El modo de direccionamiento directo hace referencia a la localidad de memoria en forma inmediata.

El modo de direccionamiento segmentado hace referencia al segmento involucrado para calcular la dirección efectiva.

El modo de direccionamiento de base utiliza el registro base BX para el cálculo de la dirección del operando.

El modo indexado involucra el registro de índice correspondiente en el cálculo de la dirección efectiva del operando.

El offset es un valor de desplazamiento que ha de tener la dirección calculada con los índices anteriores.

Un ejemplo de la forma más general en que se usa el modo de direccionamiento es la siguiente:

```
INC 6[BX][DI]
```

En este modo de direccionamiento la dirección efectiva del operando, en este caso del dato que ha de ser incrementado, es $(16 \cdot DS + BX + DI + 6)$. Esta instrucción utiliza el modo de direccionamiento segmentado, base, indexado con offset. Cabe hacer notar que no todas las instrucciones admiten todos los modos de direccionamiento, sin embargo, es preferible dejar que con la práctica se aprenda cuales los permiten y cuales no ya que sería un trabajo interminable el tratar de aprenderse todas las instrucciones con todos los modos de direccionamientos permitidos.

FIN DE PRESENTACION

APENDICES

EJERCICIOS Y NOTAS:

