



Arquitectura de Computadoras

Práctica

Lic. Mario Martin Sbarbaro

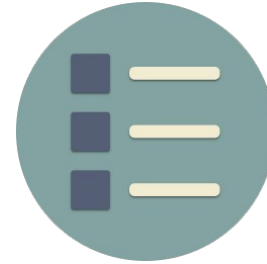
sbarbaro.martin@uader.edu.ar



Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Agenda

- Registro de estado o banderas (flags)
- Suma Hexadecimal
- Resta Hexadecimal
- Calculo de una dirección efectiva
- Modos de direccionamiento



Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Registro de estado o banderas (flags)

Registro de estado: en este registro se alojan, por nombrar algunas, todas las banderas aritméticas, banderas de modo de trabajo del microprocesador, banderas asociadas a interrupciones, etc (Quiroga, 2010).



OF: Desbordamiento

DF: Dirección en operaciones con cadenas

IF: Indicador de interrupción

TF: Modo traza

SF: Indicador de signo en operaciones con signo

ZF: Indicador de cero

AF: Acarreo del bit 3 en AL

PF: Bit de paridad

CF: Acarreo

Registro de estado o banderas (flags) ...

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.

Registro de estado o banderas (flags) ...

DF (dirección de operaciones con cadena): si es 1 el sentido de recorrido de la cadena es de izquierda a derecha, si es 0 irá en sentido contrario.

IF (indicador de interrupción): cuando vale 1 permite al procesador reconocer interrupciones. Si se pone a 0 el procesador ignorará las solicitudes de interrupción.

TF (modo traza): indica al procesador que la ejecución es paso a paso.

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Registro de estado o banderas (flags) ...

SF (indicador de signo): solo tiene sentido en las operaciones con signo.

ZF (indicador de cero): vale 1 cuando el resultado de una operación es cero.

AF (acarreo auxiliar): vale 1 cuando se produce acarreo o acarreo negativo en el bit 3.

Registro de estado o banderas (flags) ...

PF (paridad): vale 1 si el resultado de la operación tiene como resultado un número con un número par de bits a 1. Se utiliza en transmisión de datos.

CF (bit a acarreo): vale 1 si se produce acarreo en una operación de suma, o acarreo negativo en una operación de resta. Contiene el bit que ha sido desplazado o rotado fuera de un registro o posición de memoria. Refleja el resultado de una comparación.

Suma Hexadecimal

El método para sumar un número hexadecimal con otro, es el mismo que se sigue para números decimales.

El procedimiento es el siguiente:

- 1)** Sume los dos dígitos hexadecimales en decimal.
- 2)** Si el resultado de la suma es igual o menor que 15, puede expresarse directamente como un dígito hexadecimal.
- 3)** Si el resultado de la suma es mayor o igual que 16, se debe restar 16 al resultado y se acarrea 1 al dígito en la siguiente posición.

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Suma Hexadecimal

Ejercicio de ejemplo.

Realice la suma de $61CE + 2A32$. La respuesta es: $8C00$.

Actividad en clase!
7 minutos como máximo.



Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Suma Hexadecimal

Ejercicio de ejemplo.

$$\begin{array}{rcccc} & 1 & 1 & & \\ 6 & 1 & C & E & \\ + & & & & \\ 2 & A & 3 & 2 & \\ \hline 8 & C & 0 & 0 & \end{array}$$

$E + 2 \rightarrow 14 + 2 = 16 \rightarrow 16 - 16 = 0 \rightarrow$ Acarreo 1.

$1 + C + 3 = 1 + 12 + 3 = 16 \rightarrow 16 - 16 = 0 \rightarrow$ Acarreo 1.

$1 + 1 + A \rightarrow 1 + 1 + 10 = 12 \rightarrow C \rightarrow$ Sin acarreo.

$6 + 2 = 8.$

Resta Hexadecimal

El método para restar un número hexadecimal con otro, es distinto respecto de los decimales.

El método para restar números hexadecimales es el siguiente:

- 1) A cada dígito del sustraendo lo restamos de F.
- 2) Al resultado le sumamos 1.
- 3) Por último sumamos este resultado al minuendo. Debemos tener en cuenta que se descarta el acarreo de la adición del dígito más significativo.

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Resta Hexadecimal

Ejercicio de ejemplo.

Restar 27C3 (sustraendo) de 4A68 (minuyendo). La respuesta es: 22A5.

Actividad en clase!
8 minutos como máximo.



Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Resta Hexadecimal

Ejercicio de ejemplo.

$$\begin{array}{r} \text{F F F F} \\ - \\ 2 \ 7 \ \text{C} \ 3 \\ \hline \text{D} \ 8 \ 3 \ \text{C} \\ + \\ \phantom{\text{D} \ 8 \ 3} 1 \\ \hline \text{D} \ 8 \ 3 \ \text{D} \end{array}$$

$$\begin{array}{r} 1 \ 1 1 \\ 4 \ \text{A} \ 6 \ 8 \\ + \\ \text{D} \ 8 \ 3 \ \text{D} \\ \hline \text{X} \ 2 \ 2 \ \text{A} \ 5 \end{array}$$

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Calculo de una dirección efectiva

Dirección Efectiva = Registro de Segmento * 10h + Desplazamiento

Obtener la dirección efectiva a partir de los siguientes valores de segmento y desplazamiento: 0456:FC1A.

Actividad en clase!
8 minutos como máximo.



Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Calculo de una dirección efectiva

Dirección Efectiva = Registro de Segmento * 10h + Desplazamiento

Registro de Segmento * 10h

$$\begin{array}{r} \begin{array}{cccc} & 0 & 4 & 5 & 6 \\ * & & & & \\ & & & 1 & 0 \\ \hline & 0 & 0 & 0 & 0 \\ & 0 & 4 & 5 & 6 \\ \hline & 0 & 4 & 5 & 6 & 0 \end{array} \end{array}$$

$$\begin{array}{r} \begin{array}{ccccc} & 1 & 1 & & \\ & 0 & 4 & 5 & 6 & 0 \\ + & & & & & \\ & & F & C & 1 & A \\ \hline & 1 & 4 & 1 & 7 & A \end{array} \end{array}$$

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Calculo de una dirección efectiva

Dirección Efectiva = Registro de Segmento * 10h + Desplazamiento

Obtener la dirección efectiva a partir de los siguientes valores de segmento y desplazamiento: 1FFF:FC00.

Actividad en clase!
8 minutos como máximo.



Modos de direccionamiento

Indican la manera de obtener los operandos

- Direccionamiento Inmediato
- Direccionamiento por Registro
- Direccionamiento Directo
- Direccionamiento Indirecto mediante Registro
- Direccionamiento Indirecto por Registro Base
- Direccionamiento Indexado
- Direccionamiento Indexado respecto de una Base

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Modos de direccionamiento ...

Direccionamiento Inmediato

```
C:\>debug
```

```
-a 100
```

```
0769:0100 mov ax,23 ; AX = 23 direccionamiento inmediato
```

```
0769:0103 int 20
```

```
0769:0105
```

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Modos de direccionamiento ...

Direccionamiento por Registro

```
C:\>debug
```

```
-a 100
```

```
0769:0100 mov bx,23
```

```
0769:0103 mov ax,bx ; AX = BX direccionamiento por registro
```

```
0769:0105 int 20
```

```
0769:0107
```

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Modos de direccionamiento ...

Direccionamiento Directo

```
C:\>debug
```

```
-a 100
```

```
0769:0100 mov si,23
```

```
0769:0103 mov [0200],si
```

```
0769:0107 mov ax,[0200] ; AX = [0200] direccionamiento directo
```

```
0769:010A int 20
```

```
0769:010C
```

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Modos de direccionamiento ...

Direccionamiento Indirecto o Indirecto mediante Registro

```
C:\>debug
```

```
-a 100
```

```
0769:0100 mov si,23
```

```
0769:0103 mov [0200],si
```

```
0769:0107 mov bx,0200
```

```
0769:010A mov ax,[bx] ; AX = [BX] direccionamiento indirecto mediante registro
```

```
0769:010C int 20
```

```
0769:010E
```

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Modos de direccionamiento ...

Direccionamiento por registro base o relativo por registro

```
C:\>debug
```

```
-a 100
```

```
0769:0100 mov si,23
```

```
0769:0103 mov [0205],si
```

```
0769:0107 mov bx,0200
```

```
0769:010A mov ax,[bx + 05] ; AX = [BX + 05] direccionamiento por registro base
```

```
0769:010D int 20
```

```
0769:010F
```

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Modos de direccionamiento ...

Direccionamiento Indexado o Base más Índice

```
C:\>debug
```

```
-a 100
```

```
0769:0100 mov si,23
```

```
0769:0103 mov [0205],si
```

```
0769:0107 mov bx,0200
```

```
0769:010A mov di,0005
```

```
0769:010D mov ax,[bx + di] ; AX = [BX + DI] direccionamiento Indexado
```

```
0769:010F int 20
```

```
0769:0111
```

Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Modos de direccionamiento ...

Direccionamiento relativo base más Índice

```
C:\>debug
```

```
-a 100
```

```
0769:0100 mov si,23
```

```
0769:0103 mov [0207],si
```

```
0769:0107 mov bx,0200
```

```
0769:010A mov di,0005
```

```
0769:010D mov ax,[bx + di + 02] ; AX = [BX + DI + 02] direccionamiento Indexado
```

```
0769:010F int 20
```

```
0769:0111
```


Práctica – Arquitectura de Computadoras – Lic. en Sistemas de Inf. (LSI)

Bibliografía obligatoria y complementaria

Bibliografía utilizada en la clase 2

- Quiroga, P. (2010). *Arquitectura de computadoras* (1a ed.). Buenos Aires: Alfaomega Grupo Editor Argentino.



UADER | FCyT



Esta obra está bajo una [Licencia Creative Commons Atribución/Reconocimiento-NoComercial-SinDerivados 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).