

Tema 3: ASCII y operaciones de desplazamiento y lógicas

- Representación de cadenas de caracteres (alfanuméricas)
- Operaciones de desplazamiento
 - Desplazamientos lógicos
 - Desplazamientos aritméticos
 - Rotaciones
 - Desplazamientos concatenados
- Instrucciones lógicas
 - AND
 - OR
 - XOR
 - NOT
- Instrucciones de manejo de bits

Estructura de Computadores

Bibliografía básica

- 8088-8086/8087 programación ensamblador en entorno MS-DOS
Miguel Angel Roselló.
Ed. Anaya Multimedia
- Microprocesadores: el 8088 / 86
Fernando Remiro Domínguez
Agustín Martín García
Ed. Akal-Biblioteca tecnológica
- Lenguajes ensambladores
R. Martínez Tomás. Ed. Paraninfo
- Lenguaje ensamblador de los 80x86
Jon Beltrán de Heredia
Editorial Anaya-Multimedia. 1996



Representación de cadenas de caracteres (I)

- Representaciones alfanuméricas:
- Codifican mediante un grupo de bits (6, 7, 8, 16) cada uno de los caracteres a representar
- Ejemplos de códigos alfanuméricos:
 - 6 bits (64 caracteres posibles) Fieldata y BCDIC
 - 7 bits (128 caracteres posibles) ASCII
 - 8 bits (256 caracteres posibles) ASCII extendido y EBCDIC
 - 16 bits (65536 caracteres posibles) UNICODE



Representación de cadenas de caracteres (II)

- Las frases se forman agrupando caracteres. Existen varias alternativas:
 - Cadenas de longitud fija. Se define una longitud máxima para todas las cadenas

P E P E A N T O N I O R O S A

- Cadenas de longitud variable

- Con carácter separador

* P E P E * A N T O N I O * R O S A

- Con longitud explícita

4 P E P E 7 A N T O N I O 4 R O S A



Representación de cadenas de caracteres (III)

850 Multilingüe (Latin 1)	
0	128
1	129
2	130
3	131
4	132
5	133
6	134
7	135
8	136
9	137
10	138
11	139
12	140
13	141
14	142
15	143
16	144
17	145
18	146
19	147
20	148
21	149
22	150
23	151
24	152
25	153
26	154
27	155
28	156
29	157
30	158
31	159
32	160
33	161
34	162
35	163
36	164
37	165
38	166
39	167
40	168
41	169
42	170
43	171
44	172
45	173
46	174
47	175
48	176
49	177
50	178
51	179
52	180
53	181
54	182
55	183
56	184
57	185
58	186
59	187
60	188
61	189
62	190
63	191
64	192
65	193
66	194
67	195
68	196
69	197
70	198
71	199
72	200
73	201
74	202
75	203
76	204
77	205
78	206
79	207
80	208
81	209
82	210
83	211
84	212
85	213
86	214
87	215
88	216
89	217
90	218
91	219
92	220
93	221
94	222
95	223
96	224
97	225
98	226
99	227
100	228
101	229
102	230
103	231
104	232
105	233
106	234
107	235
108	236
109	237
110	238
111	239
112	240
113	241
114	242
115	243
116	244
117	245
118	246
119	247
120	248
121	249
122	250
123	251
124	252
125	253
126	254
127	255



Instrucciones de desplazamiento (I)

- Todas las instrucciones de desplazamiento, ya sean de desplazamiento aritmético, lógico, rotaciones o desplazamientos concatenadas presentan el mismo formato y presentan las mismas características:

- **Formato:** Nemónico Destino, veces

Donde

- Nemomnico es el nombre de la instrucción
- Destino es el operando que actúa como fuente y destino de la instrucción
- Veces es el número de desplazamientos (a nivel de bit) que se realiza la instrucción. Puede ser 1, o varios, en este último caso, el número de veces solicitado debe indicarse en el registro CL previamente
- Todas ellas modifican el flag de acarreo



Instrucciones de desplazamiento (II)

- **Nombre:** SAL
- **Formato:** SAL destino, contador
- **Descripción:**

Desplazará a la izquierda el operando destino tantos bits como indique el valor del contador, rellenando con ceros los bits que quedan libres por la derecha

Si el valor de contador es mayor que uno, obligatoriamente deberemos especificarlo en el registro CL

Si especifico un desplazamiento sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, el desplazamiento será local a esos ocho bits. Copia el contenido del bit más significativo en el flag de acarreo

- **Ejemplos:**

```
MOV AX, 2 ; AX = 2
SAL AX, 1 ; AX = 4
MOV CL, 2 ; CL = 2
SAL AX, CL ; AX = 16
```



Instrucciones de desplazamiento (III)

- **Nombre:** SAR
- **Formato:** SAR destino, contador
- **Descripción:**

El formato de la instrucción es SAR destino, contador

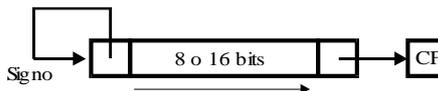
Desplazará a la derecha el operando destino tantos bits como indique el valor del contador, rellenando con el bit de signo los bits que quedan libres por la izquierda

Si el valor de contador es mayor que uno, obligatoriamente deberemos especificarlo en el registro CL

Si especifico un desplazamiento sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, el desplazamiento será local a esos ocho bits. Copia el contenido del bit menos significativo en el flag de acarreo

- **Ejemplos:**

```
MOV AX, 16 ; AX = 16
SAR AX, 1 ; AX = 8
MOV CL, 2 ; CL = 2
SAR AX, CL ; AX = 2
```



Instrucciones de desplazamiento (IV)

- **Nombre:** SHL
- **Formato:** SHL destino, contador
- **Descripción:**

Desplazará a la izquierda el operando destino tantos bits como indique el valor del contador, rellenando con ceros los bits que quedan libres por la derecha. Es un desplazamiento lógico. *El comportamiento es igual que el de la instrucción SAL*

Si el valor de contador es mayor que uno, obligatoriamente deberemos especificarlo en el registro CL

Si especifico un desplazamiento sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, el desplazamiento será local a esos ocho bits. Copia el contenido del bit más significativo en el flag de acarreo

- **Ejemplos:**

MOV AX, 0FFFFh ; AX = FFFFh (-1)

SHL AX, 1 ; AX = FFFEh (-2)

MOV CL, 2 ; CL = 2

SHL AX, CL ; AX = FFF8h (-8)



Instrucciones de desplazamiento (V)

- **Nombre:** SHR
- **Formato:** SHR destino, contador
- **Descripción:**

Desplazará a la derecha el operando destino tantos bits como indique el valor del contador, rellenando con ceros los bits que quedan libres por la izquierda. Es un desplazamiento lógico

Si el valor de contador es mayor que uno, obligatoriamente deberemos especificarlo en el registro CL

Si especifico un desplazamiento sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, el desplazamiento será local a esos ocho bits. Copia el contenido del bit menos significativo en el flag de acarreo

- **Ejemplos:**

MOV AX, 2 ; AX = FFF8h (-8)

SHR AX, 1 ; AX = 7FFCh (32764)

MOV CL, 2 ; CL = 2

SHR AX, CL ; AX = 1FFFh (8191)



Instrucciones de desplazamiento (VI)

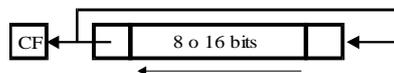
- **Nombre:** ROL
- **Formato:** ROL destino, contador
- **Descripción:**

Rota los bits del operando destino hacia la izquierda tantos bits como indique el valor del contador, rellenando con los bits que saldrían por la izquierda los bits que quedan libres por la derecha

Si especifico una rotación sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, la rotación será local a esos ocho bits. Copia el contenido del bit más significativo en el flag de acarreo

- **Ejemplos:**

```
MOV AX, 8000h ; AX = 8000h
ROL AX, 1 ; AX = 0001h
MOV CL, 2 ; CL = 2
ROL AX, CL ; AX = 0004h
```



Instrucciones de desplazamiento (VII)

- **Nombre:** ROR
- **Formato:** ROR destino, contador
- **Descripción:**

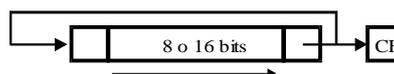
Rota los bits del operando destino hacia la derecha tantos bits como indique el valor del contador, rellenando con los bits que saldrían por la derecha los bits que quedan libres por la izquierda

Si deseamos rotar el operando una única vez a la derecha, se puede especificar directamente. Si el valor de contador es mayor que uno, obligatoriamente deberemos especificarlo en el registro CL

Si especifico una rotación sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, la rotación será local a esos ocho bits. Copia el contenido del bit menos significativo en el flag de acarreo

- **Ejemplos:**

```
MOV AX, 8001h ; AX = 8001h
ROR AX, 1 ; AX = C000h
MOV CL, 2 ; CL = 2
ROR AX, CL ; AX = 300
```



Instrucciones de desplazamiento (VIII)

- **Nombre:** RCL
- **Formato:** RCL destino, contador
- **Descripción:**

Rota los bits del operando destino hacia la izquierda a través del flag de acarreo tantos bits como indique el valor del contador

Si deseamos rotar el operando una única vez a la izquierda, se puede especificar directamente. Si el valor de contador es mayor que uno, obligatoriamente deberemos especificarlo en el registro CL

Si especifico una rotación sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, la rotación será local a esos ocho bits



- **Ejemplos:**

```
MOV AX, 8000h ; AX = 8000h y además suponemos el flag de acarreo a 1
RCL AX, 1 ; AX = 0001h
MOV CL, 2 ; CL = 2
RCL AX, CL ; AX = 0006h y el flag de acarreo estará a 0
```



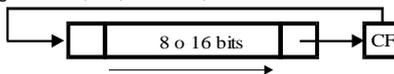
Instrucciones de desplazamiento (IX)

- **Nombre:** RCR
- **Formato:** RCR destino, contador
- **Descripción:**

Rota los bits del operando destino hacia la derecha tantos bits como indique el valor del contador a través del flag de acarreo

Si deseamos rotar el operando una única vez a la derecha, se puede especificar directamente. Si el valor de contador es mayor que uno, obligatoriamente deberemos especificarlo en el registro CL

Si especifico una rotación sobre un registro de ocho bits, la parte alta o la parte baja de los registros AX, BX, CX o DX, la rotación será local a esos ocho bits



- **Ejemplos:**

```
MOV AX, 8001h ; AX = 8001h y suponemos el flag de acarreo a 1
ROR AX, 1 ; AX = C000h
MOV CL, 2 ; CL = 2
ROR AX, CL ; AX = 7000h y el flag de acarreo estará a 0
```



Instrucciones lógicas (I)

- **Nombre:** AND
- **Formato:** AND destino, origen
- **Descripción:**

Realiza a nivel de bits la operación lógica AND entre el origen y el destino
Tanto el origen como el destino pueden ser operandos de 8 o de 16 bits, pero ambos del mismo tamaño

- **Ejemplos:**
AND AX, BX ; Si AX = 7777h y BX = 2222h entonces AX AND BX = 2222h
AND AX, 1 ; AX = 4 entonces AX AND 1 = 0
AND AX, 0FFFFh ; Sea cual sea el valor de AX, AX AND 0FFFFh = AX

a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1



Instrucciones lógicas (II)

- **Nombre:** OR
- **Formato:** OR destino, origen
- **Descripción:**

Realiza a nivel de bits la operación lógica OR entre el origen y el destino
Tanto el origen como el destino pueden ser operandos de 8 o de 16 bits, pero ambos del mismo tamaño

- **Ejemplos:**
OR AX, BX ; Si AX = 7777h y BX = 2222h entonces AX OR BX = 7777h
OR AX, 1 ; Si AX = 4 entonces AX OR 1 = 5
OR AX, 0h ; Sea cual sea el valor de AX, AX OR 0FFFFh = AX

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1



Instrucciones lógicas (III)

- **Nombre:** XOR
- **Formato:** XOR destino, origen
- **Descripción:**

Realiza a nivel de bits la operación lógica XOR entre el origen y el destino
Tanto el origen como el destino pueden ser operandos de 8 o de 16 bits, pero ambos del mismo tamaño
- **Ejemplos:**

XOR AX, BX ; Si AX = 7777h y BX = 2222h entonces AX XOR BX = 5555h
XOR AX, 1 ; Si AX = 4 entonces AX XOR 1 = 5
XOR AX, 0FFFFh ; Si AX = 7777h AX XOR 0FFFFh = 8888h

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0



Instrucciones lógicas (IV)

- **Nombre:** NOT
- **Formato:** NOT destino
- **Descripción:**

Realiza a nivel de bits la operación lógica NOT del destino
El destino pueden ser operandos de 8 o de 16 bits, pero ambos del mismo tamaño
- **Ejemplos:**

NOT AX ; Si AX = 7777h entonces NOT AX= 8888h

a	NOT a
0	1
1	0



Instrucciones de manejo de bits (I)

- **Nombre:** TEST
- **Formato:** TEST destino, contador
- **Descripción:**
 - Es igual que la operación AND (y lógico) pero no se guarda el resultado en destino
 - Se realiza a nivel de bit y modifica los flag de estado
- **Ejemplos:**
 - ; AX = 1234, BX = 0000
 - TEST AX, BX ; AX =1234, BX = 0000, Resultado = 0000
 - JZ Es_Cero ; Salta si es cero a la posición Es_Cero



Instrucciones de manejo de bits (II)

- **Nombre:** CLI
- **Formato:** CLI
- **Descripción:**
 - Desactiva las interrupciones
 - Las interrupciones no enmascarables no se pueden inhibir
- **Ejemplos:**
 - CLI _____
- **Nombre:** STI
- **Formato:** STI
- **Descripción:**
 - Permite las interrupciones
- **Ejemplos:**
 - STI



Instrucciones de manejo de bits (III)

- **Nombre:** CLC
- **Formato:** CLC
- **Descripción:**
Desactiva el flag de acarreo
- **Ejemplos:**
CLC

-
- **Nombre:** STC
 - **Formato:** STC
 - **Descripción:**
Activa el flag de acarreo
 - **Ejemplos:**
STC



Instrucciones de manejo de bits (IV)

- **Nombre:** CLD
- **Formato:** CLD
- **Descripción:**
Pone a cero el flag de dirección (para trabajar con las instrucciones de cadenas)
- **Ejemplos:**
CLD

-
- **Nombre:** STD
 - **Formato:** STD
 - **Descripción:**
Pone a uno el flag de dirección (para trabajar con las instrucciones de cadenas)
 - **Ejemplos:**
STD

