

2007-15-G-S-A9

ave directiva sirve para reservar posiciones de memoria  $\rightarrow DS \rightarrow \underline{b}$ .

2007-15-G-S-A14

```
MOVE.W D2, 8(A0, D3.W)
```

A0 = 00000004  
 D3.W = 0002  
 8  
 -----  
 0000000E

	Dir. mem.	Conten.
A0 = 00000004	00000008	10
D2 = 01234BCD	9	1B
D3 = 000C0002	A	1C
	B	14
	C	0C
	D	0A
	E	0B
	F	03

D2  $\rightarrow$  (0000000E)  $\Rightarrow$  (0000000E) = AB (0000000F) = CD

Se da por válida la a pero no lo es pq en una dir de memoria solo cabe un byte. La solución buena sería AB

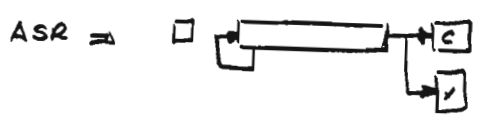
2007-15-G-S-A15

MOVE.B # \$1A, \$B  $\rightarrow$  Mueve el valor # \$1A a la dir \$B

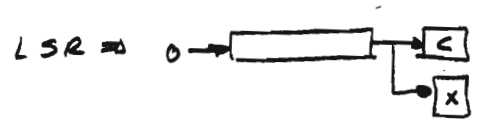
$(\$0000000B) = 1A \Rightarrow \underline{a}$

2007-25-G-S-F.12

Un registro de la CPU contiene C3 y se realiza una oper. aritm. a dicha. El resultado es



$\frac{11000011}{C \quad B} \rightarrow ASR \rightarrow \frac{11100001}{E \quad I}$



$11000011 \rightarrow LSR \rightarrow \frac{01100001}{6 \quad I}$

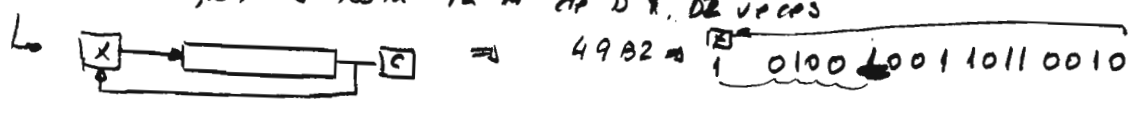
NO es 86 y NO depende del carry  $\Rightarrow EI \Rightarrow \underline{b}$

2007-25-G-S-F.16

Contenido de D1 después de ROR.W D2, D1

Inicialmente C = X = 1 / (D1) = \$A1FC49B2 (D2) = \$00000005

ROR.W D2, D1  $\Rightarrow$  Rota la W de D1, D2 veces



49 B2 →

1 <sup>a</sup>	0100 1001 1011 0010	F
2 <sup>a</sup>	0 1010 0100 1101 1001	0
3 <sup>a</sup>	1 0101 0010 0110 1100	1
4 <sup>a</sup>	0 1010 1001 0011 0110	0
5 <sup>a</sup>	0 0101 0100 1001 1011	0
5 <sup>a</sup>	1 0110 1010 0100 1101	1

2      A      4      D      ⇒ (D1) = \$ A18C2A4D } c  
c = 1

2007. 2S. G. S. F20

Inicialmente (D0) = \$ 0000 020A

SUBI.B # \$ E, D0 ⇒ Restar a

0A	0000 1010
DE	0000 1110
	<u>1111 1100</u>
	F      C

(D0) = \$ 0000 02FC ⇒ b

No se corre la llevada al 02

2007. Sep. G. S. A9

¿Qué registro del 69000 no tiene 32 bits? El de estado ⇒ a

2007. Sep. G. S. A10

EOU se utiliza para definir datos ctes ⇒ d

2007. Sep. G. S. A17 / 2007. 2S. AD. C1

Inicialmente (D0) = \$ 0000 0003 (D1) = \$ B85D 0505 D2 = \$ 5D 8B 0385

ITER	EOR.W	D2, D1	0505 0385	0068 0385	D03E 0385
	ROR.W	#4, D1	↓	0000 0000 0110 1000	1101 0000 0011 1110
	SUBI.B	#1, D0	0000 0101 0000 0101	0000 0011 1000 0101	0000 0011 1000 0101
	BNE	ITER	<u>0000 0110 1000 0000</u>	0 3 E D	D 3 B B
			0 6 8 0	D 0 3 E	B D 3 B
	ROR		0 0 6 8		

(D1) = \$ B85D B03B ⇒ a

2007-Sep-G-S-A 19 / 2007-15-AD-A1

¿Cuál podría ser el contenido de D4 y D5 si después de MOVE.B D5,D4

D4 = \$12453341    D5 = \$12453341

MOVE.B D5,D4 ⇒ Orja igual D5 y los 3 bytes altos de D4 mueve al byte bajo de D5 a D4

Antes del move ⇒ D5 = \$12453341  
 D4 = \$124533xx } ⇒ solo puede ser **C**

2007-Sep-Res-G-S-1 / 2007-25-AD-C10

El registro de códigos de condición forma parte del: Byte menos significativo del registro de estado. ⇒ **a**

2007-Sep-Res-G-S-17 / 2007-15-AD-C5

COLOR EQU \$90  
 LINV EQU 256 ⇒ 256 decimal a hexadecimal = 100116  
 LINH EQU 128 ⇒ 128 " " " = 80116  
 COND DC.L COLOR, LINV, LINH

Definir codig. largos (32 bits) ⇒ 4 bytes por label ⇒ 12 bytes  
 ↓  
**d**

2007-Sep-Res-G-S-17 / 2007-25-AD-C7

D0 = \$0000020A    D1 = \$0000FFFF    D2 = \$FFFF9EFA

OR.W D1,D2 →

	9EFA	1001 1110 1111 1010
	FFFF	1111 1111 1111 1111
	FFFF	

AND.B D2,D0

EOR.L D0,D1

FF	1111 1111
0A	0000 1010
0A	0000 1010
	0 0 ⇒ D0 = \$0000020A

D0 = 0000020A    D1 = 0000FFFF

0010	1010
1111	1111
1101	0101

← **a** 0000FD F5

2007-15-AD-A2

D4 = 3A8041153

D5 = 5BCDE8A01

D6 = 501FFB374

AND.B D4,D5 → 01 ⇒ D5 = 5BCDE8A01

OR.B D5,D6 → 01

NOT.B D6 → 74 - 0111 0100

EOR.B D6,D4 → 75 ⇒ 0111 0101 ⇒ 1000 1010 = 8A

8A - 1000 1010

53 - 0101 0011

1101 1001 ⇒ D9 ⇒ D4 = 3A80411D9

D6 = 501FFB38A

⇓

a

2007-15-AD-A3

DATO D5.L \$12 ⇒ Reserva 18 grupos de 4 bytes = 72 ⇒ d

⇓

J3 ⇒ 18 decimal ⇒ 18

2007-Sep-AD-A7

D4 = 580108040

D6 = 51F67ASC9

OR.W D4,D6 ⇒

ASC9 - 1010 0101 1100 1001

8040 - 1000 0000 0100 0000

---

1010 0101 1100 1001

A S C 9

D6 = 51F67ASC9 ⇒ b

2007-Sep-Res-AD-C8

↳ d Hecho en Sep 2008 - Reserva

Pg E.AP68000.S7

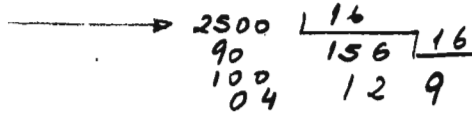
2007-Sep-AD-A13 / 2007-Sep-Res-AD-A10

MOVE.L #000080FS, D4

Transfere el dato 80FS al D4 ⇒ d

2007. Sep. AD. A 8

ORG 2500



2500 = \$ 9C4

9C4 INI EQU \$FDF

9C4 ----- MOVE.L # \$00F0481, D2 → (D2) = \$ 00F0481

9CA ----- ADD.L ET, D2 → 00F0481

9D0 ----- AND.W #INI, D2 → 4E  
 000F04CF

9D4 ET DC.L 34E → 00F 04CF  
 0FDF

END

0000 0100 1100 1111  
 0000 1111 1101 1111

0 4 C F

↓  
 D2 = \$ 00F04CF → 6