2008 1S(A) GS 7

7.- Enunciado cierto respecto a la PILA:

los datos que se introducen en la pila se van sacando en orden inverso al de entrada ⇒ b (Pg 470)

2008 1S(A) GS 17

17.- Inicio:

D4: $AB041153 D5: $BCDEBA01 D6: $01FFB374

AND.B D4,D5 ; AND 53 01 ⇒ 0101 0011 / 0000 0001 / 0000 0001 (0 1) ⇒ (D5) = $BCDEBA01

OR.B D5,D6 ; OR 01 74 ⇒ 0000 0001 / 0111 0100 / 0111 0101 (7 5) ⇒ (D6) = $01FFB375

NOT.B D6 ; 75 NOT ⇒ 0111 0101 / 1000 1010 (8 A) ⇒ (D6) = $01FFB38A

EOR.B D6,D4 ; EOR 8A 53 ⇒ 1000 1010 / 0101 0011 / 1101 1001 (D 9) ⇒ (D4) = $AB0411D9

⇒ a

3

2008 1S(A) GS 18

18.- D0 después de:

```
        ORG     2500            ; 2500 = $9C4
9C4 INI EQU     $F5F            ; INI = $F5F
9C4     MOVE.L  #$000F0481,D0   ; (D0) = $000F0481
9C8     ADD.L   ET,D0           ; ET = $42 ⇒ 000F0481 / 00000042 / (D0) = 000F04C3
9D0     AND.W   #INI,D0         ;
```

AND 04C3 → 0000 0100 1100 0011 / 0F5F → 0000 1111 0101 1111 / 0000 0100 0100 0011 (0 4 4 3)

(D0) = $000F0443

```
ET      DC.L    $42
```

⇒ b

2008 1S(A) GS 19

19.- ¿bytes reservados en memoria ?

DATO DS.L $12

Reserva $12 (18 (10)) palabras largas (L = 4 bytes)

18 · 4 = 72 posiciones ⇒ d

2008 1S(A) AO 9

9) ADD.W D0, D1

D0 = $00.02.58 63

D1 = $81.05.42.21

al ser W + 5863 / 4221 / 9A84

D1 = $81.05.9A.84 luego la c

2008 1S(A) AO 13

13. La directiva EQU se utiliza para:

a) Reservar posiciones de memoria para utilizarlas posteriormente

b) Definir un símbolo que se va a utilizar posteriormente

c) Indicar la dirección absoluta de las instrucciones del programa

d) Reservar espacio en memoria y asignarle un valor

Reserva espacio en memoria y asigna un valor. -> d

2008 2S(A) AO 5

5) D4 = $00FF00FF

D5 = $AB12 12AB

EOR.L D4, D5 →

00000000 11111111 00000000 11111111

10101011 00010010 00010010 10101011

10101011 11101101 00010010 10101100

A B E D 1 2 5 4

luego la b

9./

```
                ORG      2500  →  2500(10) = 9C4(H)
$9C4  | COM     EQU      $F5F
$9C4  |         MOVE.L   #$000F0481, D0  →  D0 = $000F0481
$9CA  |         ADD.L    NUL, D0
$9D0  |         AND.W    #COM, D0        →  D0 + NUL ⇒ 000F0481
                                                      00000042
$9D4  | NUL     DC.L     $42                       D0=$000F04C3
                END
```

```
COM = 0F5F = 0000 1111 0101 1111
D0.w = Ø4C3  0000 0100 1100 0011
             -------------------
             0000 0100 0100 0011
               0    4    4    3
```

(D0) = 000F0443 ⇒ c

2008_2S_GS_6

En el MC68000, un número en base 8 se denota mediante el símbolo @ ⇒ c

2008.2S.GS.19

Indicar el valor de D1 si inicialmente
D0 = $ 0000 020A
D1 = $ 0000 FFFF
D2 = $ F0F0 9EFA

OR.W D1,D2 ; D2 = $F0F0 FFFF
OR → FFFF
9EFA
FFFF

AND.B D2,D0 ; D0 = $ 0000 020A
AND 0A
FF
0A

EOR.L D0,D1 ; D1 = $0000FDF5
0000 020A
EOR 0000 FFFF
0000 FDF5

⇒ a

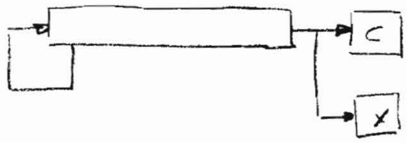
0010 - 2
1111 - F
1101 → D

1010 - A
1111 - F
0101 - 5

EOR
↓
F y cualq. cosa = $\overline{\text{cualq cosa}}$
0 y " " = cualq cosa

2008_2S_GS_C 20

Un reg. de la CPU contiene $E1_{(16}$ y se opera con una instrucción de desplaz. aritmético a dcha. El resultado es



E1 → 1110 0001 → 1, 1

1111 0000 → $F\emptyset_{(16}$ ⟹ **b**

2008_Sep_GS_A_8

El biestable de cero se pone a "1" cuando el resultado ha sido cero ⟹ **a**

2008_Sep_GS_A17

Inicialmente (D6) = $5F∅2C3∅2

ANDI.B #$F∅,D6 ; (D6) = $5F∅2C3 00 ⟹ **b**

F∅	1111 0000
AND ∅2	0000 0010
∅∅	0000 0000

2008_Sep_GS_A20

D∅ = $ 0000 0003

D1 = $ B85D 0505

D2 = $ 5D83 0385

ITER: EOR.W D2,D1 ;(D1)= B85D 0680

	0505	0000 0101 0000 0101
EOR	0385	0000 0011 1000 0101
		0000 0110 1000 0000
		0 6 8 0

ROR.W #4,D1 ; (D1) = B85D 0068

0680 → ROR (4) ⟹ 0068

SUBI.B #1, D0 = 0000 0002 ?

BNE ITER

SIGUE

EOR.W D2,D1 ; (D1) = B85D 03ED EOR 0385 = 1000 0101 / 0110 1000 = 1110 1101 = E D

ROR.W #4,D1 ; (D1) = B85D D03E

SUBI.B # 1 , D0 = 0000 0000

BNE ITER

EOR.W D2,D1 ; (D1) = B85D D3BB EOR 0B85 = 1000 0101 / 0011 1110 = 1011 1011 = B B

ROR.W #4,D1 ; (D1) = B85D BD3B

SUBI.B # 1, D0 = 0000 0000 - 1

BNE ITER , FIN

⇓

a

2008 - Sep - Res - GS - 11

D) = C3 ⟶ Rot. dcha = ROR

⇓

b

C3 = 1100 0011 → ROR

1110 0001 → 1

→ LSR

0110 0001 → 1 / 1

→ ASR

1110 0001 → 1 / 1

ROR equivale a ASR

⇓

b

2008. Sep. Res GS.16

(D1) = $ 0000 FFFF (D2) = $ 3582 9EFA

OR.W D1,D2 ⇒ (D1) = 35B2 FFFF OR FFFF / 9EFA = FFFF

⇓

a

2008- Sep. Res. GS - C 20

Cuanto serían D4 y D5 si después de MOVE.B D5,D4 el contenido es

D4 = 1245 3341 D5 = 1245 3341

MOVE.B D5,D4 ⇒ Mueve el byte bajo de D5 a D4

Si byte de D4 = 41 ⇒ antes byte de D5 = 41

⇓

C o la d

pero la d no puede ser pq el valor alto de D4 no varia con MOVE.B D5,D4 y por lo tanto antes de la operación debería ser 1245 33 ⇒ Solo puede ser la C

2008 - Sep. A0. A2

D4 = 87 0A C1 9A D6 = F1 65 F2 82

OR.W D4,D6 → (D6) = F165 F39A

⇓

b

OR C19A — 1100 0001 1001 1010
F282 — 1111 0010 1000 0010
1111 0011 1001 1010
F 3 9 A

2008 - Sep. Res. A0. C8

D0 = 0002 5861 D1 = 8105 4221.

ADD.W D0,D1 ; D0 = 8105 9A82 ADD

⇓

a

5861 → 0101 1000 0110 0001
4221 0100 0010 0010 0001
1001 1010 1000 0010
9 A 8 2

2008 - Sep. Res. A0. C10

Cuantos bits forman parte del registro de estado del 68000 → 16 bits ⇒ C