

Ejercicio 1

Si inicialmente los registros se encuentran en este estado:

a = \$ffffffffffffff
b = \$ffffffffffffff
x = \$ffffffffffff

y se ejecuta el siguiente código:

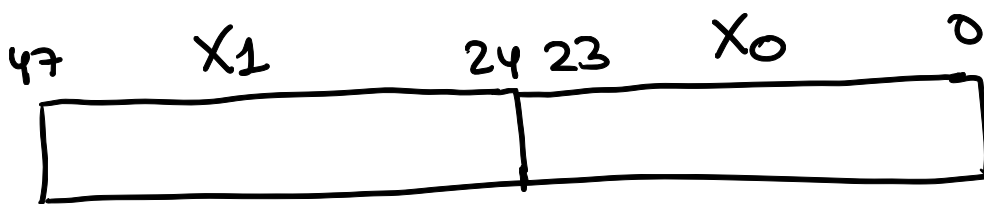
```
move #3d,x1  
move #3d,a1  
move #3d,b
```

Indicar el estado final de los registros

a =
b =
x =

La instrucción `move #3d,x1` lo que hace es usar direccionamiento inmediato y cargar el número 3d (hexa) en x1 /

(ver pág 80 pdf del manual)



Es un `move immediate short into 24 bit register` lo que resulta:

porque transfiere a x1

x1 x0
f f f f f f ; f f f f f f → 3d 000 ; f f f f f f

lo que tengo `move #3d,a1` y como a1 es parte del acumulador A, se transfiere a la parte menos

significativa

A2 A1 A0 A2 A1 A0
f f f f f f f f f f f f → f f 0000 3d f f f f f f

(ver ejemplo A pág 80 (pdf) del manual)

La última instrucción es `move #3d,b 10`

Cual resulta en:

B2	B1	B0	B2	B1	B0
FF	FFF	FFF	00	3d	000000

→

∴ luego, los estados finales de los registros son:

`a = FF 0000 3d FFFF FF`

`b = 00 3d 0000 000000`

`x = 3d 000 FFFF FF`

Ejercicio 2

Si inicialmente los registros se encuentran en este estado:

`a = $0000000000000000`
`b = $0000000000000000`
`x = $0000000000000000`

y se ejecuta el siguiente código:

```
move #caba00,x1
move x1,a
move x1,b1
```

Indicar el estado final de los registros

`a =`
`b =`
`x =`

inicialmente tengo

	X1	X0
000000	000000	000000

 y ahora tengo

	X1	X0
00	000000	000000

 luego con el `move x1,a`

con `a = 00 000000 000000` ahora tengo

A0	A1	A2
FF	caba	00 000000

 → mover de registro a acumulador

es signado frecuentemente. Se pone FF para mantener signo.

y con el nove x_1, b_1 tempo $\overset{B_0}{00}$ $cab\overset{B_1}{2}00$ $000\overset{B_2}{000}$

Finalmente, tempo

$a = ff\ cab\overset{B_0}{2}00\ 000000$

$b = 00\ cab\overset{B_0}{2}00\ 000000$

$x = cab\overset{B_0}{2}00\ 000000$