Problema 10.- A partir de la dirección $F0 nos encontramos con 16 datos. Se pide diseñar un programa que almacene en $E0 el número de 1's totales existentes en los 16 datos.¿qué modificaciones habría que realizar para contar el número de 0's totales?.

	Pseudocódigo	Ensamblador		
1				
2	PUNT ← $DF		LAIM	$DF
			STA	PUNT
3	CONT ← 16		LAIM	16
			STA	CONT
4	SUMA ← 0		LAIM	0
			STA	SUMA
5	Hacer	BUCLE:		
6	N1 ← PUNT[]		LAIM	0
			ADDI	PUNT
			STA	N1
7	RESP ← CALCULA1S(N1)		JSR	CALCULA1S
8	SUMA←SUMA+RESP		LDA	SUMA
			ADD	RESP
			STA	SUMA
9	PUNT←PUNT-1		DBZ	PUNT
10	CONT←CONT-1		DBZ	CONT
	Mientras (CONT ≠ 0)		JMP	BUCLE
11	Fin		STOP	
12	Subr CALCULA1S (N1) dev RESP	CALCULA1S:		
13	CONT1←12		LAIM	12
			STA	CONT1
14	RESP ← 0		LAIM	0
			STA	RESP
15	P1 ← N1		LDA	N1
			STA	P1
16	Hacer	BUCLERUT:		
17	P1 ← ROR (P1)		LDA	P1
			ROR	
			STA	P1
18	Si(carry)		BCS	INCREM
19	RESP←RESP +1			
20	Fsi			
	CONT1←CONT1-1	SALTORUT:	DBZ	CONT1
21	Mientras (CONT1 ≠0)		JMP	BUCLERUT

22	Fin CALCULA1S		RTS	
		INCREM:	LAIM	1
			ADD	RESP
			STA	RESP
23			JMP	SALTORUT

Para el caso en que se cuenten el número de 1's en lugar del número de 0's se podrían dar dos soluciones distintas:

a) Al resultado total obtenido del programa principal se le resta 192, que es el número de bits que contienen los 16 operandos.

10	CONT←CONT-1		DBZ	CONT
	Mientras (CONT ≠ 0)		JMP	BUCLE
11	SUMA ← 192 - SUMA		LAIM	192
			SUB	SUMA
			STA	SUMA
12	fin		STOP	
13	Subr CALCULA1S (N1) dev RESP	CALCULA1S:		

b) Se modifica la subrutina para que ésta incremente la variable RESP cuando carry sea cero.

12	Subr CALCULA1S (N1) dev RESP	CALCULA1S:		
13	CONT1←12		LAIM	12
			STA	CONT1
14	RESP ← 0		LAIM	0
			STA	RESP
15	P1 ← N1		LDA	N1
			STA	P1
16	Hacer	BUCLERUT:		
17	P1 ← ROR (P1)		LDA	P1
			ROR	
			STA	P1
18	Si(carry=0)		BCS	SALTORUT
	RESP←RESP +1		LAIM	
			ADD	RESP
19			STA	RESP
20	Fsi			
	CONT1←CONT1-1	SALTORUT:	DBZ	CONT1
21	Mientras (CONT1 ≠0)		JMP	BUCLERUT
22	Fin CALCULA1S		RTS	