

Escuela de Ingenierías Industriales

Sistemas Electrónicos Digitales

Ejercicios resueltos



1. Enunciados

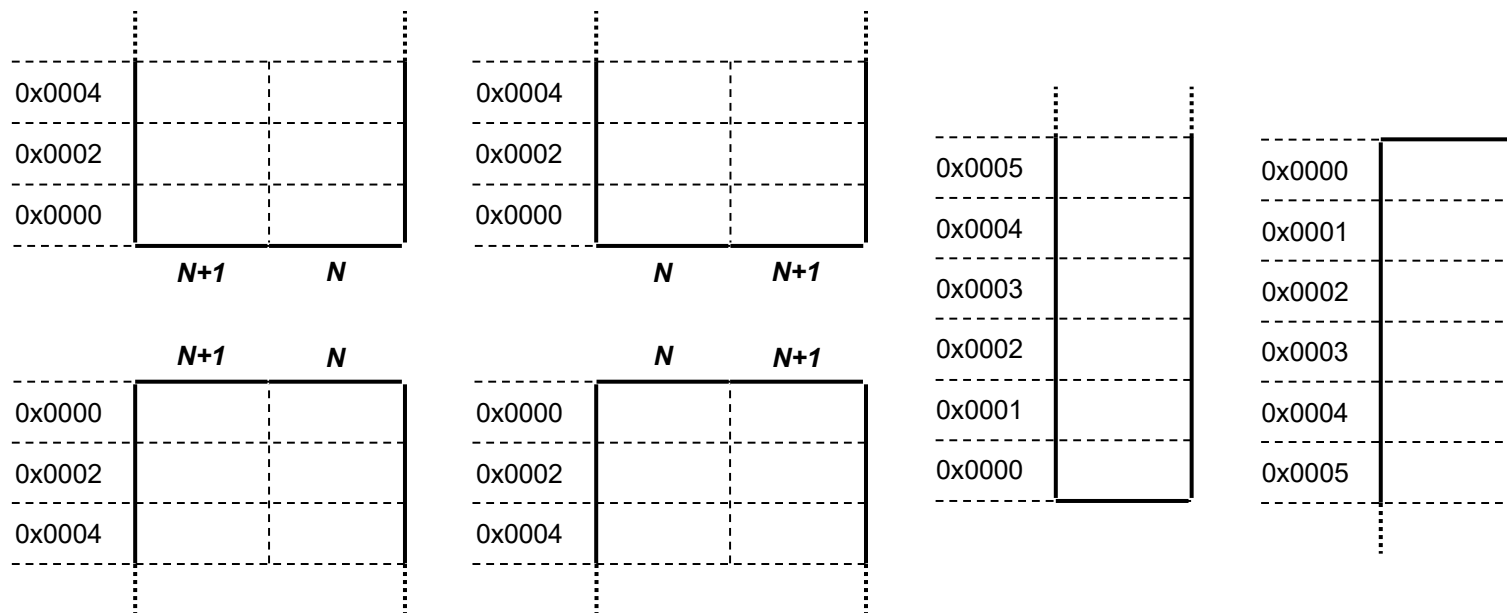
2. Solución

1. Enunciados

2. Solución

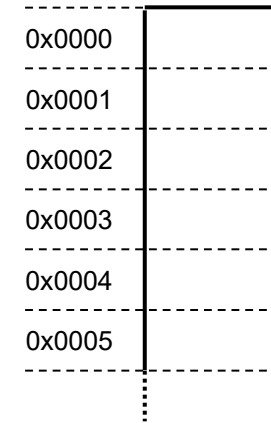
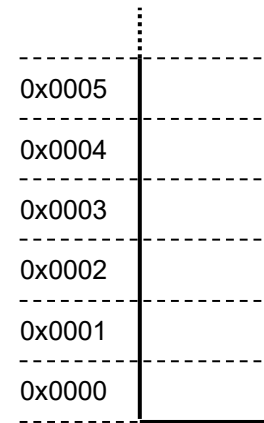
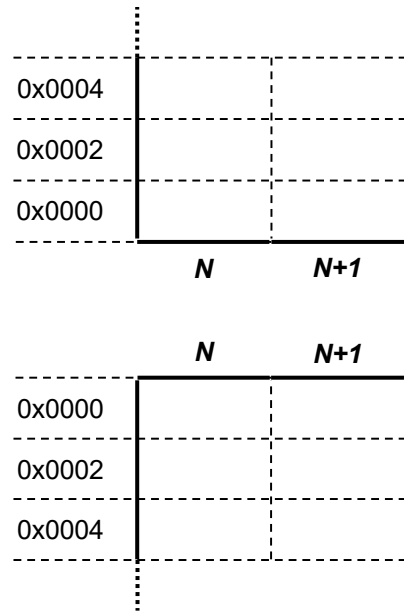
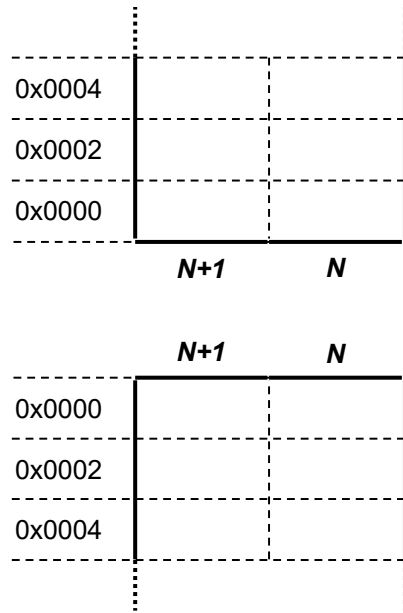
Ejercicio 1:

- Almacenar las palabras $0x0123$, $0x4567$ y $0x89AB$ en una memoria con formato **Little endian** comenzando en la posición $0x0000$, utilizando para ello las siguientes representaciones:



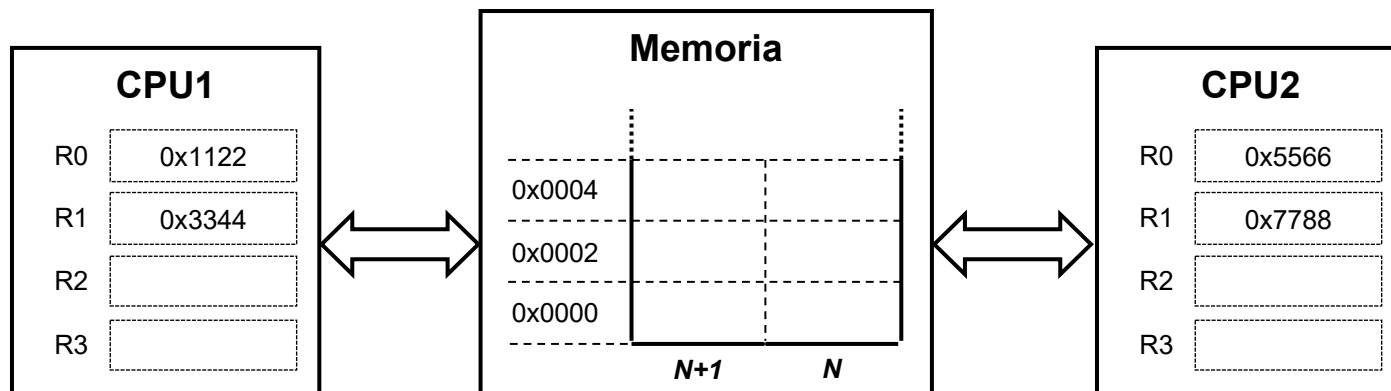
NOTA: Los símbolos “**N**” y “**N+1**” indican que, a nivel de byte, la columna correspondiente de la memoria donde se encuentran dichos símbolos contienen las direcciones pares e impares respectivamente

- Almacenar las palabras $0x0123$, $0x4567$ y $0x89AB$ en una memoria con formato **Big endian** comenzando en la posición $0x0000$, utilizando para ello las siguientes representaciones:



Ejercicio 2:

- En un microcontrolador de altas prestaciones se dispone de dos CPUs de 16 bits que comparten una misma memoria. La **CPU1** opera con formato **Little endian** mientras que la **CPU2** opera con formato **Big endian**.

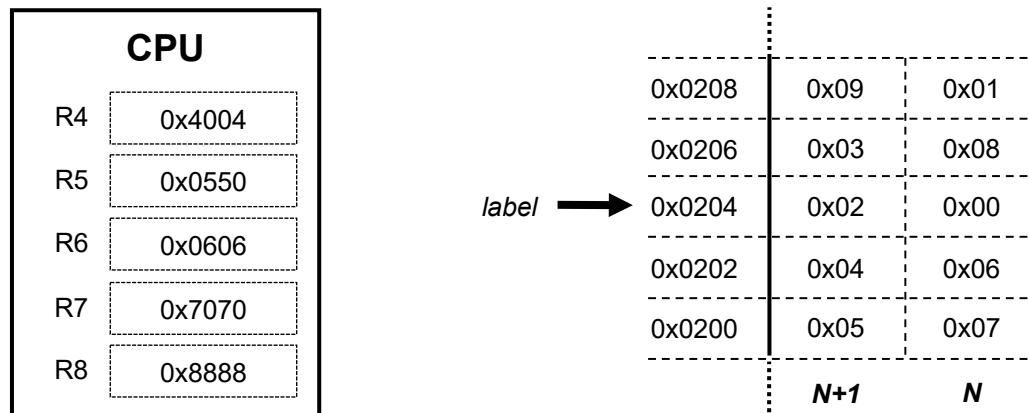


– Indicar el contenido de la memoria y de los registros R2 y R3 de ambas CPUs tras ejecutar el siguiente conjunto de instrucciones en el orden indicado:

- CPU1 → MOV.W R0, 0x0002
- CPU2 → MOV.W R1, 0x0000
- CPU2 → MOV.B R0, 0x0005
- CPU1 → MOV.B R1, 0x0004
- CPU1 → MOV.W 0x0000, R2
- CPU2 → MOV.W 0x0002, R2
- CPU1 → MOV.W 0x0004, R3
- CPU2 → MOV.W 0x0004, R3

Ejercicio 3:

– En un microcontrolador de la familia MSP430 se dispone de los siguientes datos en los registros de la CPU y en la memoria:



– Indicar el contenido de los registros de la CPU y de la memoria tras ejecutar el siguiente conjunto de instrucciones:

```

MOV.B R4, R7
MOV.W label, R4
MOV.W #label, R5
MOV.W label(R7), R6
MOV.W 2(R5), R7
MOV.B @R4, R8
MOV.B @R4, 3(R5)
MOV.B @R5+, 0(R5)
MOV.W -2(R5), 2(R5)

```

Ejercicio 4:

- El contenido de la memoria de un microcontrolador cuya CPU opera con formato **Little endian** es el que se muestra a continuación, el cual se corresponde con un programa en código máquina:

0xC016	0x23	0xFC
0xC014	0x83	0x19
0xC012	0x00	0x0C
0xC010	0x4A	0xBA
0xC00E	0x42	0x39
0xC00C	0x83	0x2A
0xC00A	0x40	0x0A
0xC008	0x01	0x20
0xC006	0x5A	0x80
0xC004	0x40	0xB2
0xC002	0x02	0x80
0xC000	0x40	0x31
	N+1	N

Ejercicio 5:

- Indicar el contenido del registro R10 tras ejecutar el siguiente conjunto de instrucciones en un microcontrolador de la familia MSP430:

```

MOV.W #0x1111,R10
ADD.W #6,PC
MOV.W #0x2222,R10
JMP    FIN
MOV.W #0x3333,R10
JMP    FIN
MOV.W #0x4444,R10
FIN    JMP    FIN
    
```

Ejercicio 6:

– Se dispone del siguiente código para el MSP430G2533:

- Programa principal: 0xC000
- RTI 1: Timer1_A3 → TA1CCR0 CCIFG
- RTI 2: Watchdog Timer+ → WDTIFG
- RTI 3: ADC10 → ADC10IFG

Se pide:

- Inicializar los valores almacenados en la tabla de vectores de interrupción

0xC00A	MOV.B #0x02,R10	}	RTI 1
0xC00C	RLA.B R10		
0xC00E	JMP 0xC00E		
...			
0xC10A	RRA.B R10	}	RTI 1
0xC10C	INCD.B R10		
0xC10E	RETI		
...			
0xC20A	RLA.B R10	}	RTI 2
0xC20C	DECD.B R10		
0xC20E	RETI		
...			
0xC30A	DEC.B R10	}	RTI 3
0xC30C	MOV.W #0xC00C,2(SP)		
0xC312	RETI		

Ejercicio 7:

- a) ¿Qué valor binario tiene la etiqueta BIT6? ¿Y la etiqueta BIT7?
- b) Dado el registro de 8 bits P1REN, inicializar a 0 el bit 6 y a 1 el bit 7 usando las etiquetas anteriores.
- c) Dado el registro TACTL de 16 bits:
 - ¿Qué valores binarios tienen las etiquetas MC0 y MC1?
 - Inicializa el campo MCx de este registro a 11 usando las etiquetas anteriores.
 - ¿Qué valores binarios tienen las etiquetas TASSEL0 y TASSEL1?
 - Inicializa el campo TASSELx a 10 utilizando las etiquetas anteriores.
 - ¿Que valor binario tiene la etiqueta TAIE?
 - Inicializa a 1 el bit TAIE y a 11 el campo TASSELx.
 - Inicializa a 00 el campo IDx.

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx	MCx		Unused	TACLr	TAIE	TAIFG	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Ejercicio 8:

En un microcontrolador de la familia MSP430 tenemos el siguiente código almacenado en memoria FLASH:

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

Este código se corresponde con el siguiente conjunto de instrucciones en ensamblador:

```

mov.w  R12,R13
mov.w  2(R12),6(R13)
mov.w  &A,&B      ;A=0200h B=0204h
mov.w  @R13+,2(R12)
mov.w  #43,R14
mov.w  A,C        ;C=0206h
    
```

Ejercicio 8:

Los valores almacenados en los registros de la CPU y en la memoria RAM son los siguientes:

PC	F83C
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	CDCD
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

Indicar como se modifican los registros y la memoria tras la ejecución de cada instrucción

Ejercicio 9:

Si la CPU del microcontrolador está funcionando con un reloj de frecuencia 1MHz (T=1us), ¿cuánto tiempo tarda en ejecutarse la secuencia de código?

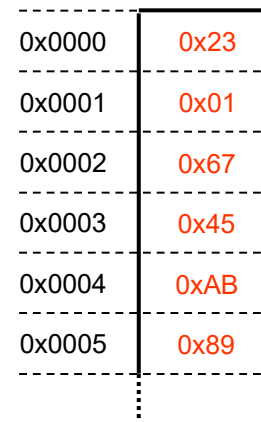
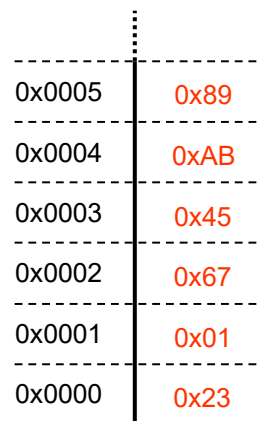
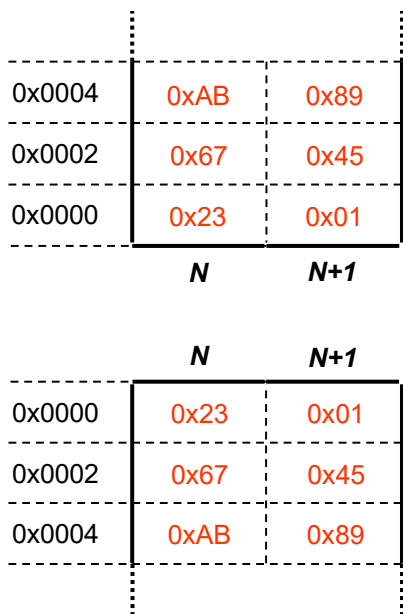
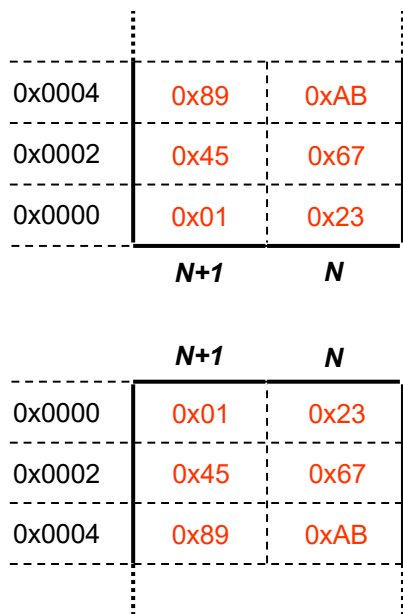
```
mov.w R12,R13
mov.w 2(R12),6(R13)
mov.w &A,&B ;A=0200h B=0204h
mov.w @R13+,2(R12)
mov.w #43,R14
mov.w A,C ;C=0206h
```

1. Enunciados

2. Solución

Ejercicio 1:

Little endian



Big endian

0x0004	0xAB	0x89
0x0002	0x67	0x45
0x0000	0x23	0x01
	N+1	N

0x0004	0x89	0xAB
0x0002	0x45	0x67
0x0000	0x01	0x23
	N	N+1

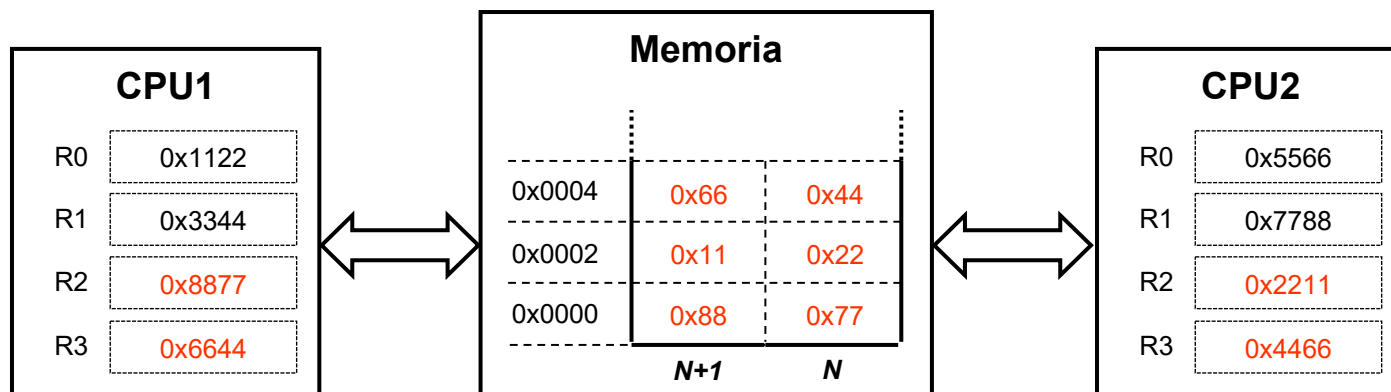
0x0005	0xAB
0x0004	0x89
0x0003	0x67
0x0002	0x45
0x0001	0x23
0x0000	0x01

0x0000	0x01
0x0001	0x23
0x0002	0x45
0x0003	0x67
0x0004	0x89
0x0005	0xAB

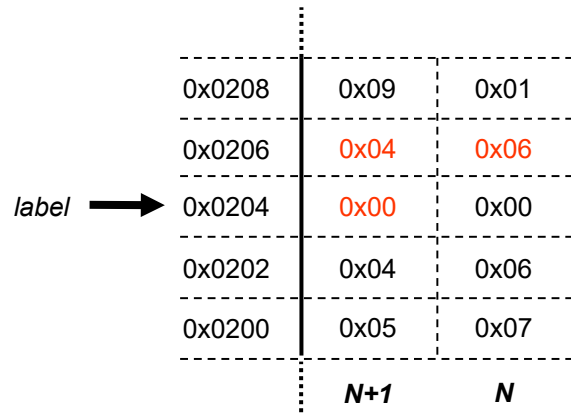
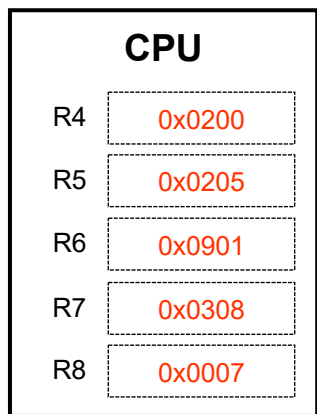
	N+1	N
0x0000	0x23	0x01
0x0002	0x67	0x45
0x0004	0xAB	0x89

	N	N+1
0x0000	0x01	0x23
0x0002	0x45	0x67
0x0004	0x89	0xAB

Ejercicio 2:



Ejercicio 3:



Ejercicio 4:

- a) ¿Cuáles son las instrucciones del programa?
- b) ¿Qué longitud de palabra tiene el microcontrolador, sabiendo que es mayor de 8 bits?
- c) ¿Qué tamaño tiene cada una de las instrucciones del programa?
- d) ¿Cuántos operandos admite como máximo una instrucción?

0x40310280

0x40B25A800120

0x400A

0x832A

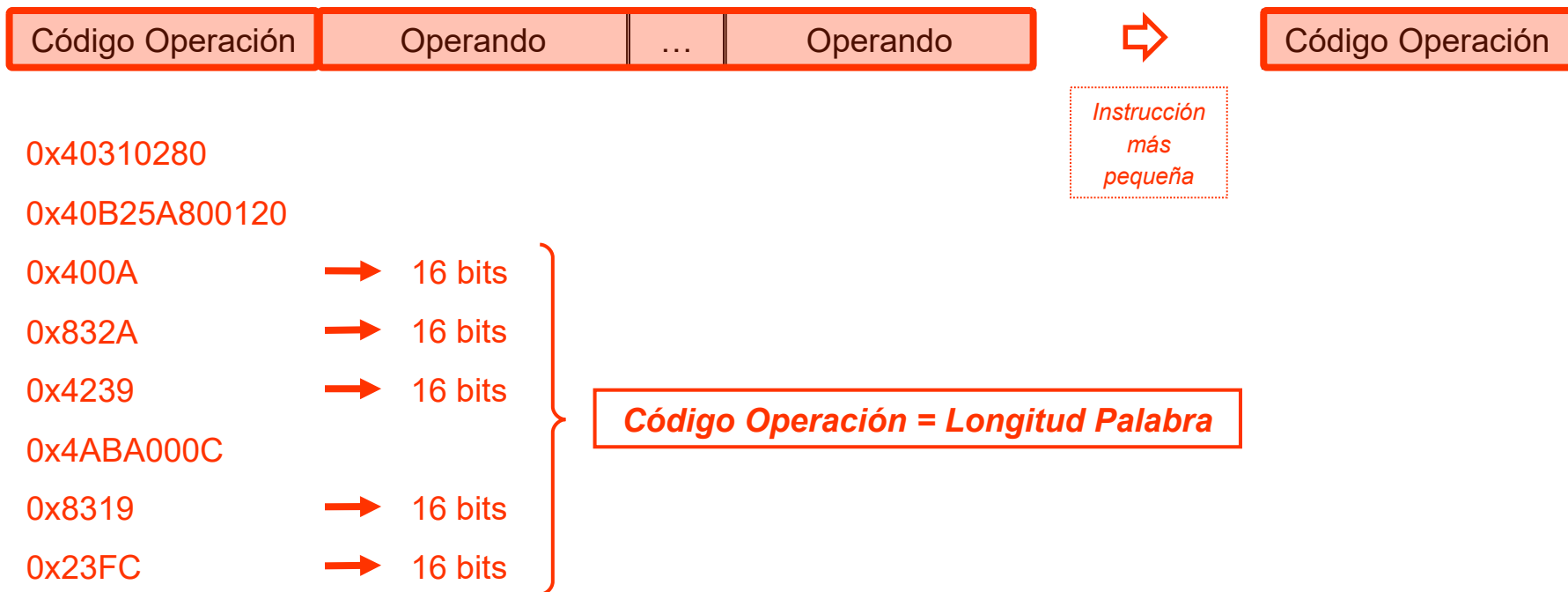
0x4239

0x4ABA000C

0x8319

0x23FC

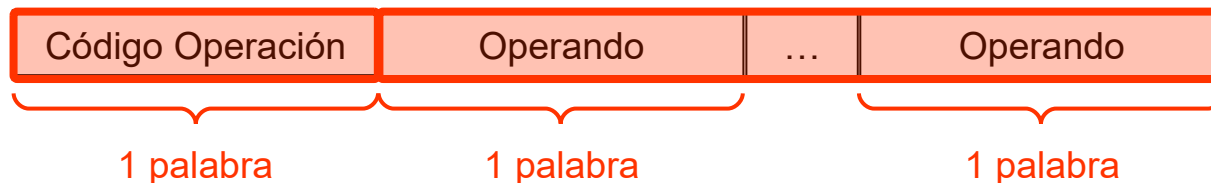
- ¿Cuáles son las instrucciones del programa?
- ¿Qué longitud de palabra tiene el microcontrolador, sabiendo que es mayor de 8 bits?
- ¿Qué tamaño tiene cada una de las instrucciones del programa?
- ¿Cuántos operandos admite como máximo una instrucción?



- a) ¿Cuáles son las instrucciones del programa?
- b) ¿Qué longitud de palabra tiene el microcontrolador, sabiendo que es mayor de 8 bits?
- c) ¿Qué tamaño tiene cada una de las instrucciones del programa?
- d) ¿Cuántos operandos admite como máximo una instrucción?

0x40310280	→	2 palabras
0x40B25A800120	→	3 palabras
0x400A	→	1 palabra
0x832A	→	1 palabra
0x4239	→	1 palabra
0x4ABA000C	→	2 palabras
0x8319	→	1 palabra
0x23FC	→	1 palabra

- ¿Cuáles son las instrucciones del programa?
- ¿Qué longitud de palabra tiene el microcontrolador, sabiendo que es mayor de 8 bits?
- ¿Qué tamaño tiene cada una de las instrucciones del programa?
- ¿Cuántos operandos admite como máximo una instrucción?



0x40310280

0x40B25A800120



3 palabras = Código Operación + 2 operandos

0x400A

0x832A

0x4239

0x4ABA000C

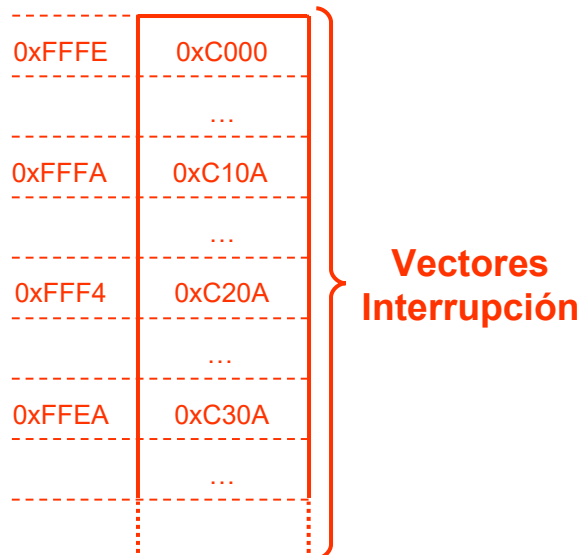
0x8319

0x23FC

Ejercicio 5:

- R10 → 0x3333

Ejercicio 6:



Ejercicio 7:

1. ¿Qué valor binario tiene la etiqueta BIT6? ¿Y la etiqueta BIT7?

En el fichero de cabecera msp430g2553.h:

```

/*****
 * STANDARD BITS
 *****/

#define BIT0          (0x0001)
#define BIT1          (0x0002)
#define BIT2          (0x0004)
#define BIT3          (0x0008)
#define BIT4          (0x0010)
#define BIT5          (0x0020)
#define BIT6          (0x0040)
#define BIT7          (0x0080)
#define BIT8          (0x0100)
#define BIT9          (0x0200)
#define BITA          (0x0400)
#define BITB          (0x0800)
#define BITC          (0x1000)
#define BITD          (0x2000)
#define BITE          (0x4000)
#define BITF          (0x8000)

```

BIT6 → 0000000001000000
 BIT7 → 0000000010000000

b) Dado el registro de 8 bits P1REN, inicializar a 0 el bit 6 y a 1 el bit 7 usando las etiquetas anteriores.

```
P1REN &= ~BIT6;
```

```
P1REN |= BIT7;
```

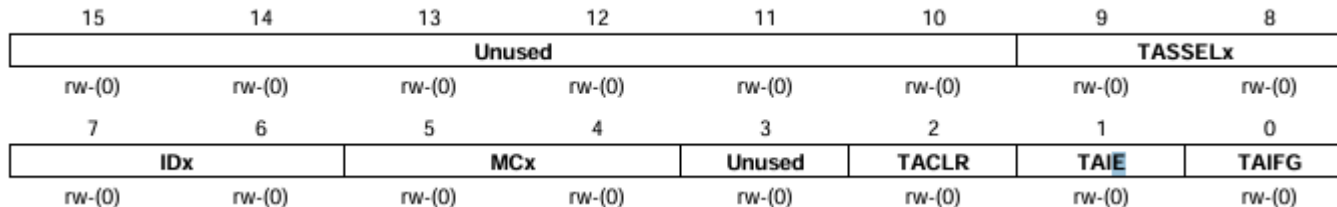
Lo comprobamos: suponemos que P1REN tiene un valor cualquiera: P1REN=XXXXXXXX

$P1REN = P1REN \& \sim BIT6 = XXXXXXXX \& 10111111 = X0XXXXXX \rightarrow$ Hemos puesto a 0 el bit 6.

$P1REN = P1REN | BIT7 = X0XXXXXX | 10000000 = 10XXXXXX \rightarrow$ Ya tenemos a 0 el bit 6 y a 1 el bit 7.

c)

TACTL



¿Qué valores binarios tienen las etiquetas MC0 y MC1?

```

545
546 #define TASSEL1          (0x0200)    /* Timer A clock source select 1 */
547 #define TASSEL0          (0x0100)    /* Timer A clock source select 0 */
548 #define ID1              (0x0080)    /* Timer A clock input divider 1 */
549 #define ID0              (0x0040)    /* Timer A clock input divider 0 */
550 #define MC1              (0x0020)    /* Timer A mode control 1 */
551 #define MC0              (0x0010)    /* Timer A mode control 0 */
552 #define TACLr            (0x0004)    /* Timer A counter clear */
553 #define TAIE             (0x0002)    /* Timer A counter interrupt enable */
554 #define TAIFG            (0x0001)    /* Timer A counter interrupt flag */
555

```

MC0 → 0000000000010000

MC1 → 000000000100000

Inicializa el campo MCx de este registro a 11 usando las etiquetas anteriores:

TACTL |= (MC1|MC0);

Lo comprobamos: si TACTL = XXXXXXXXXXXXXXXXXXXX

MC1|MC0 = 0000 0000 0010 0000 | 0000 0000 0001 0000 = 0000 0000 0011 0000

TACTL = TACTL | (MC1|MC0) = XXXX XXXX XXXX XXXX | 0000 0000 0011 0000 =
= XXXX XXXX XX11 XXXX

c)

TACTL

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx		MCx		Unused	TACLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

– ¿Qué valores binarios tienen las etiquetas TASSEL0 y TASSEL1?

```

545
546 #define TASSEL1          (0x0200)    /* Timer A clock source select 1 */
547 #define TASSEL0          (0x0100)    /* Timer A clock source select 0 */
548 #define ID1              (0x0080)    /* Timer A clock input divider 1 */
549 #define ID0              (0x0040)    /* Timer A clock input divider 0 */
550 #define MC1              (0x0020)    /* Timer A mode control 1 */
551 #define MC0              (0x0010)    /* Timer A mode control 0 */
552 #define TACLR            (0x0004)    /* Timer A counter clear */
553 #define TAIE             (0x0002)    /* Timer A counter interrupt enable */
554 #define TAIFG            (0x0001)    /* Timer A counter interrupt flag */
555

```

TASSEL0 → 0000000100000000

TASSEL1 → 0000001000000000

Inicializa el campo TASSELx de este registro a 10 usando las etiquetas anteriores:

TACTL |= TASSEL1;

TACTL &= ~TASSEL0;

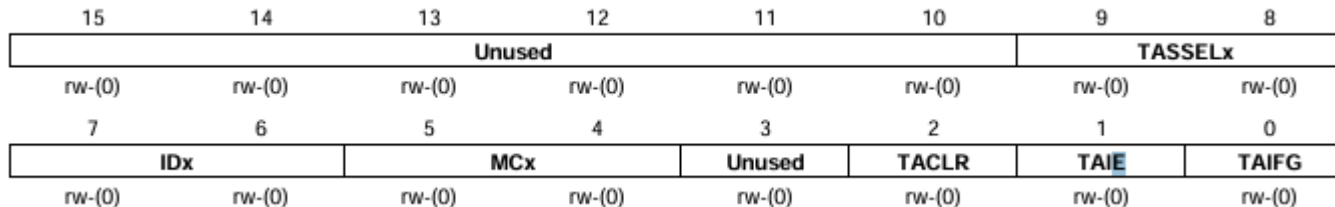
Lo comprobamos: si TACTL = XXXXXXXXXXXXXXXXXXXX

TACTL = TACTL | TASSEL1 = XXXX XXXX XXXX XXXX | 0000 0010 0000 0000 =
 = XXXX XX1X XXXX XXXX

TACTL = TACTL & ~TASSEL0 = XXXX XX1X XXXX XXXX & 1111 1110 1111 1111 =
 = XXXX XX10 XXXX XXXX

c)

TACTL



– ¿¿Que valor binario tiene la etiqueta TAIE?

```

545
546 #define TASSEL1           (0x0200)    /* Timer A clock source select 1 */
547 #define TASSEL0           (0x0100)    /* Timer A clock source select 0 */
548 #define ID1                (0x0080)    /* Timer A clock input divider 1 */
549 #define ID0                (0x0040)    /* Timer A clock input divider 0 */
550 #define MC1                (0x0020)    /* Timer A mode control 1 */
551 #define MC0                (0x0010)    /* Timer A mode control 0 */
552 #define TACLx              (0x0004)    /* Timer A counter clear */
553 #define TAIE               (0x0002)    /* Timer A counter interrupt enable */
554 #define TAIFG              (0x0001)    /* Timer A counter interrupt flag */

```

TAIE → 0000000000000010

Inicializa a 1 el bit TAIE y a 11 el campo TASSELx.

TACTL |= (TAIE|TASSEL0|TASSEL1);

Lo comprobamos: si TACTL = XXXXXXXXXXXXXXXXXXXX

$$\begin{aligned}
 \text{TAIE|TASSEL0|TASSEL1} &= 0000000000000010 \mid 0000000100000000 \mid 0000001000000000 = \\
 &= 0000\ 0011\ 0000\ 0010
 \end{aligned}$$

$$\begin{aligned}
 \text{TACTL} &= \text{TACTL} \mid (\text{TAIE|TASSEL0|TASSEL1}) = \text{XXXX}\ \text{XXXX}\ \text{XXXX}\ \text{XXXX} \mid 0000\ 0011\ 0000\ 0010 = \\
 &= \text{XXXX}\ \text{XX11}\ \text{XXXX}\ \text{XX1X}
 \end{aligned}$$

c)

TACTL

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx		MCx		Unused	TACLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

– Inicializa a 00 el campo IDx.

```

545
546 #define TASSEL1           (0x0200)    /* Timer A clock source select 1 */
547 #define TASSEL0           (0x0100)    /* Timer A clock source select 0 */
548 #define ID1                (0x0080)    /* Timer A clock input divider 1 */
549 #define ID0                (0x0040)    /* Timer A clock input divider 0 */
550 #define MC1                (0x0020)    /* Timer A mode control 1 */
551 #define MC0                (0x0010)    /* Timer A mode control 0 */
552 #define TACLR              (0x0004)    /* Timer A counter clear */
553 #define TAIE               (0x0002)    /* Timer A counter interrupt enable */
554 #define TAIFG              (0x0001)    /* Timer A counter interrupt flag */
555

```

Tengo que utilizar las etiquetas ID0 e ID1.

```
TACTL &= ~(ID0|ID1);
```

Lo comprobamos: si TACTL = XXXXXXXXXXXXXXXXXXXX

ID0 | ID1 = 0000 0000 0100 0000 | 0000 0000 1000 0000 = 0000 0000 1100 0000

TACTL = TACTL & ~(ID0|ID1) = XXXX XXXX XXXX XXXX & 1111 1111 0011 1111 =
= XXXX XXXX 00XX XXXX

Ejercicio 8:

PC	F83C
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	CDCD
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N
F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

```

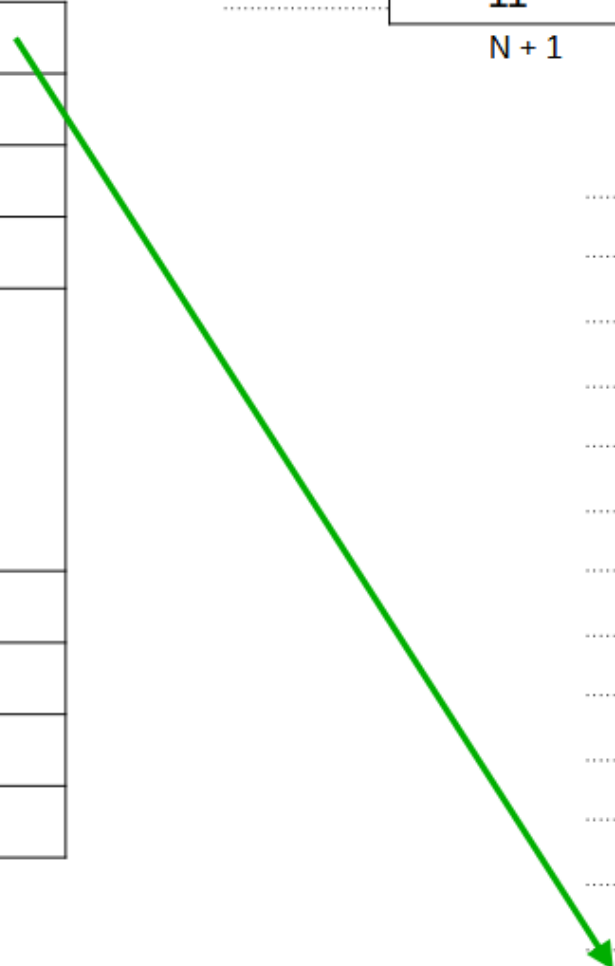
mov.w R12,R13
mov.w 2(R12),6(R13)
mov.w &A,&B ;A=0200h
;B=0204h
mov.w @R13+,2(R12)
mov.w #43,R14
mov.w A,C ;C=0206h
    
```

mov.w R12,R13

<u>PC</u>	F83C
<u>SP</u>	XXXX
<u>SR</u>	XXXX
<u>CG</u>	XXXX
R12	0200
R13	CDCD
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

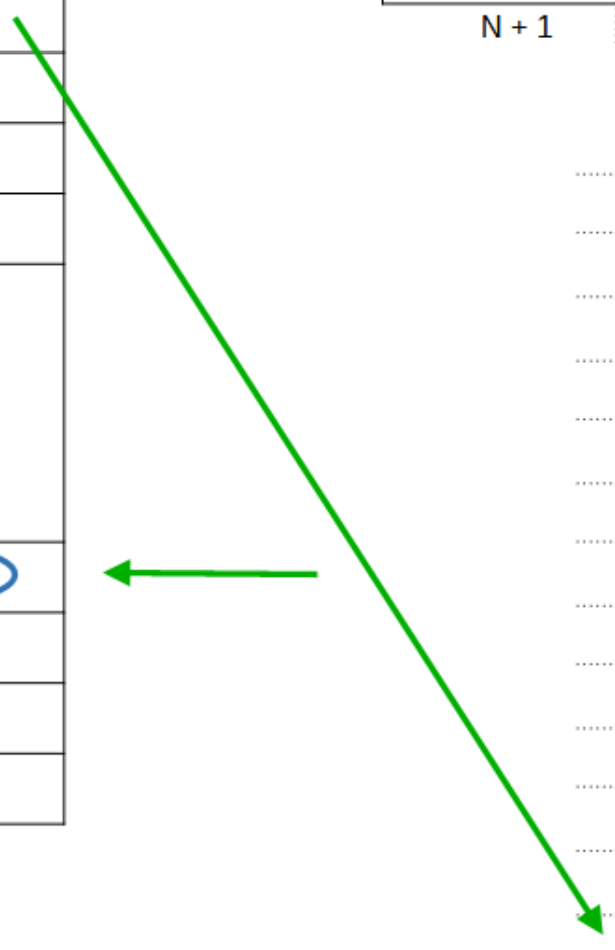


mov.w R12,R13

PC	F83C
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	CDCD
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

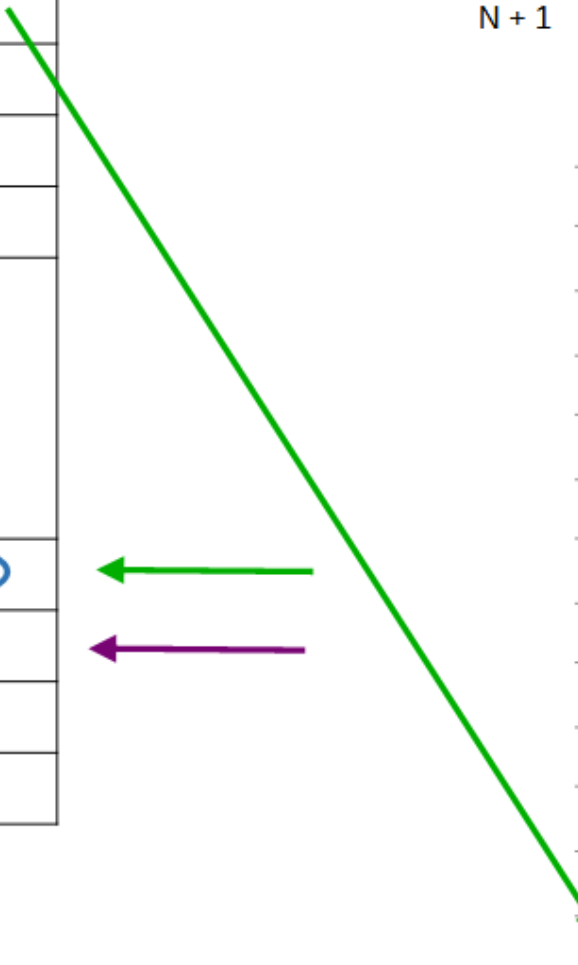


mov.w R12,R13

PC	F83C
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	CDCD
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

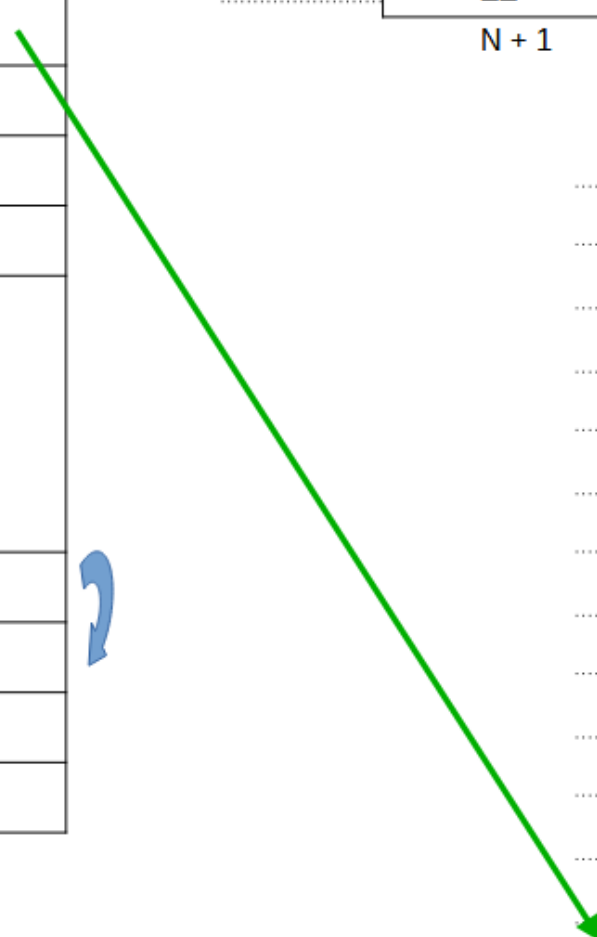


mov.w R12,R13

PC	F83C
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

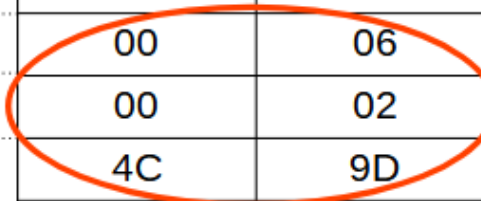
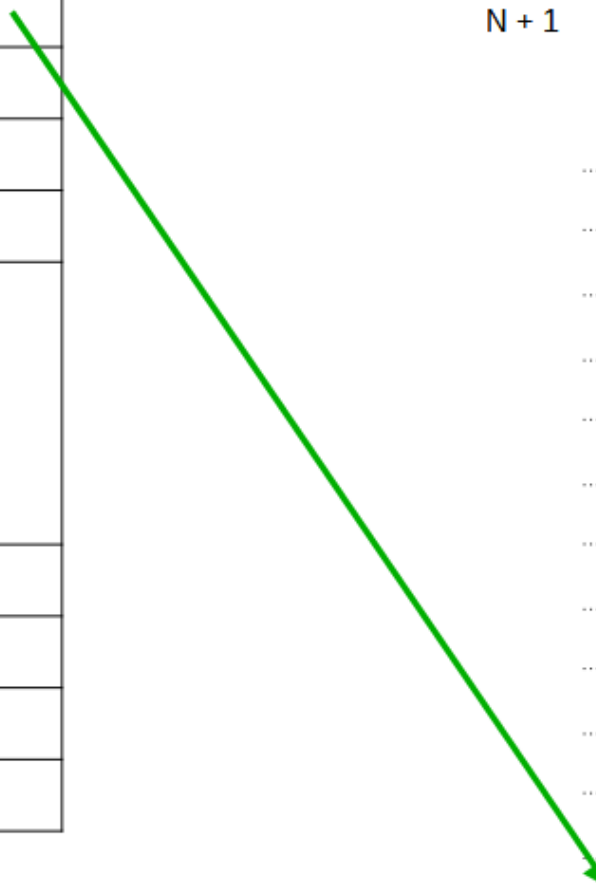


mov.w 2(R12),6(R13)

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

PC	F83E
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



mov.w 2(R12),6(R13)

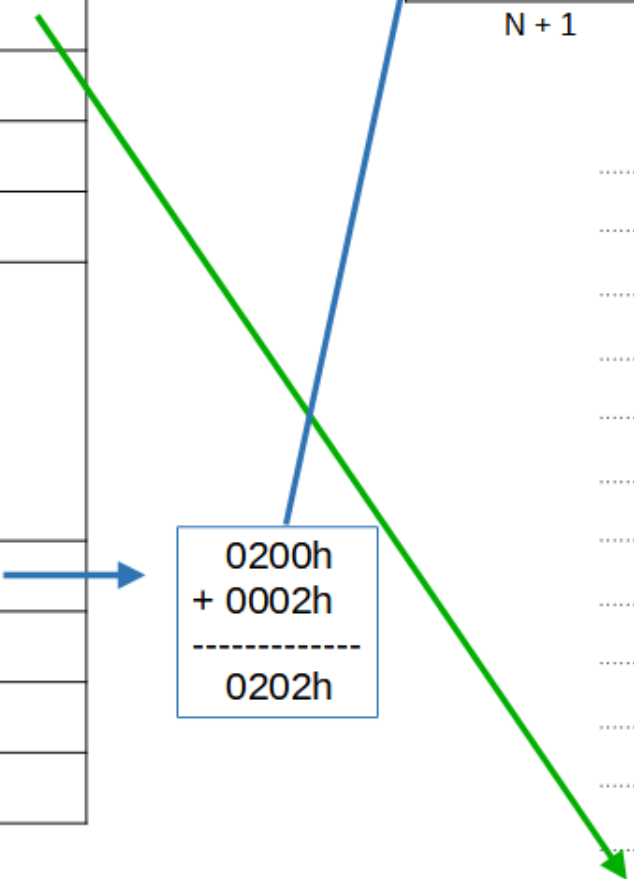
PC	F83E
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

0200h
+ 0002h

0202h

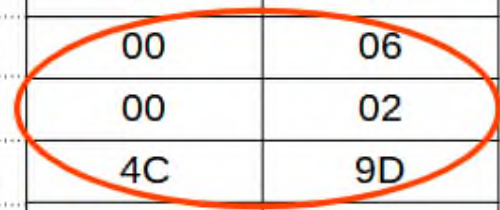
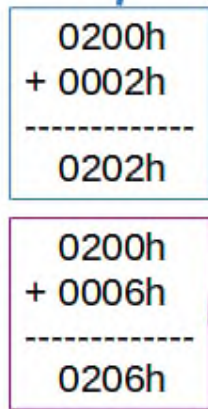


mov.w 2(R12),6(R13)

PC	F83E
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	77	88
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

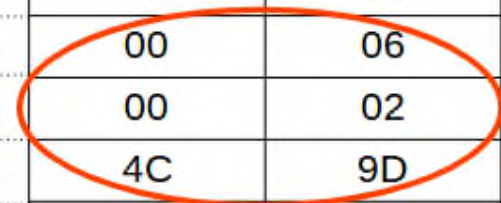
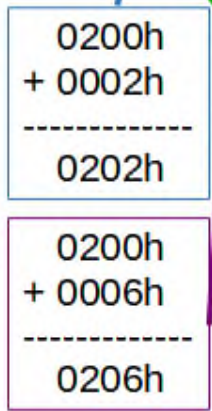


mov.w 2(R12),6(R13)

PC	F83E
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	33	44
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



mov.w &A,&B

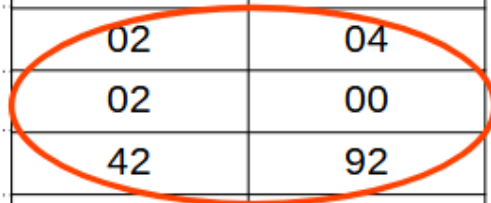
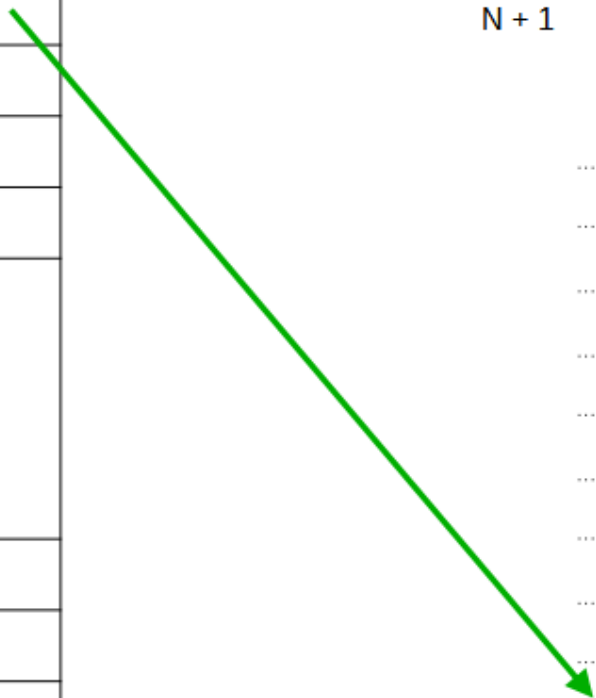
A=200h

B=204h

PC	F844
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	33	44
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



mov.w &A,&B

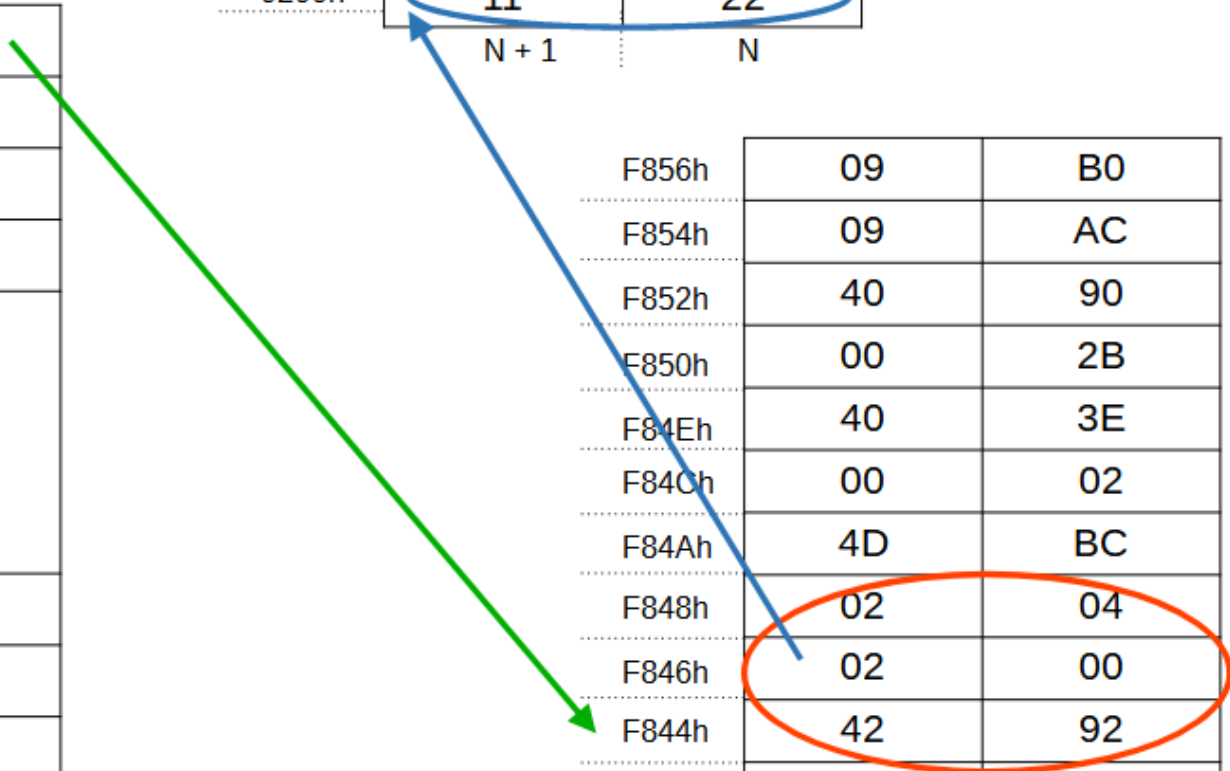
A=200h

B=204h

PC	F844
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	33	44
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



mov.w &A,&B

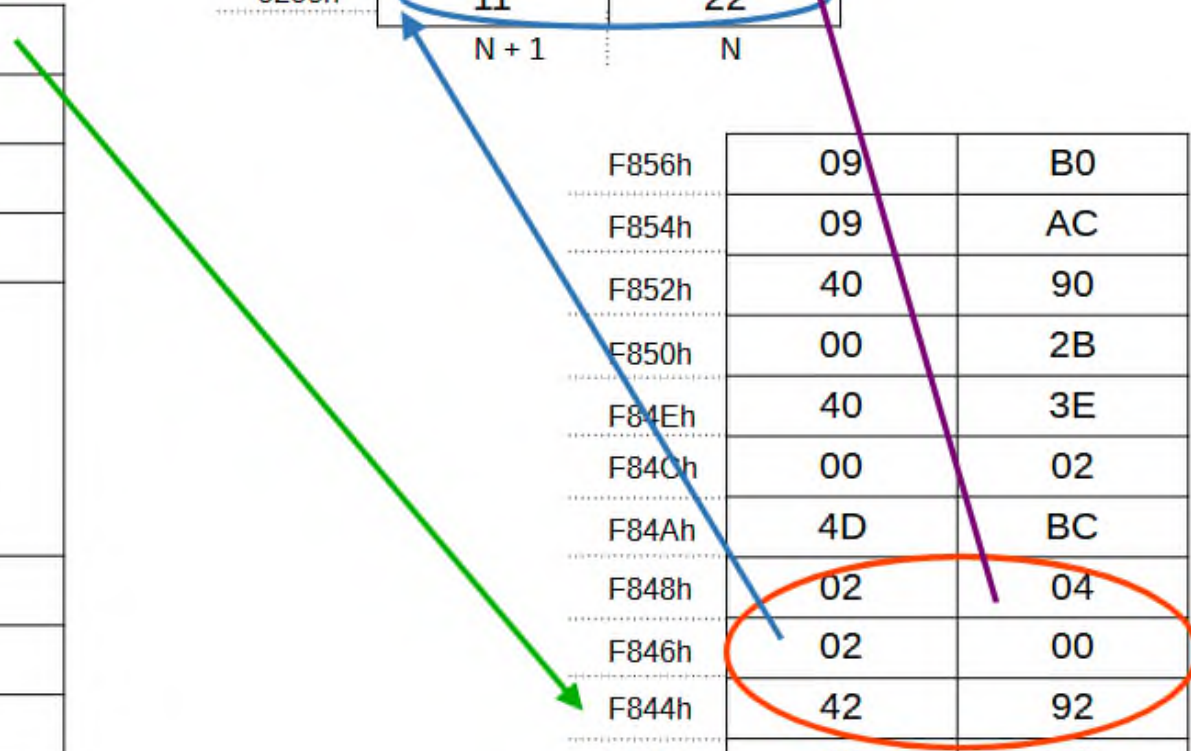
A=200h

B=204h

PC	F844
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	33	44
0204h	55	66
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



`mov.w &A,&B`

A=200h

B=204h

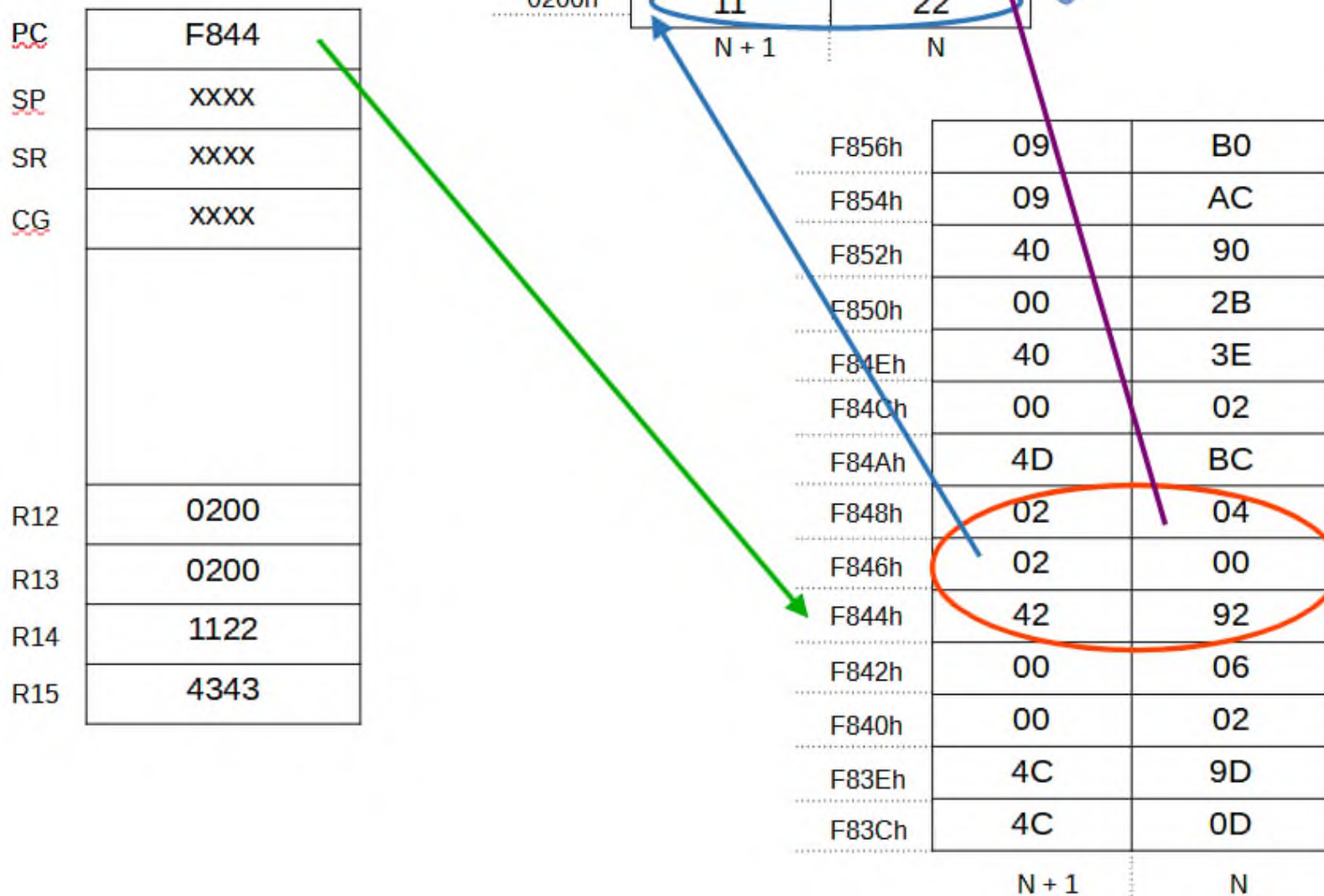
PC	F844
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	33	44
0204h	11	22
0202h	33	44
0200h	11	22

N + 1 N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D

N + 1 N

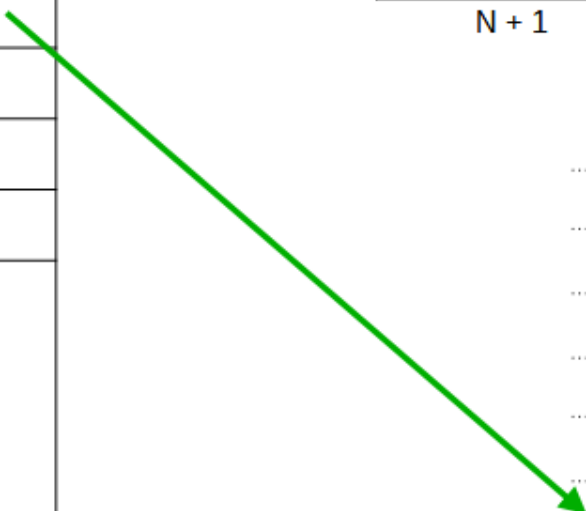


mov.w @R13+,2(R12)

0206h	33	44
0204h	11	22
0202h	33	44
0200h	11	22
	N + 1	N

PC	F84A
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

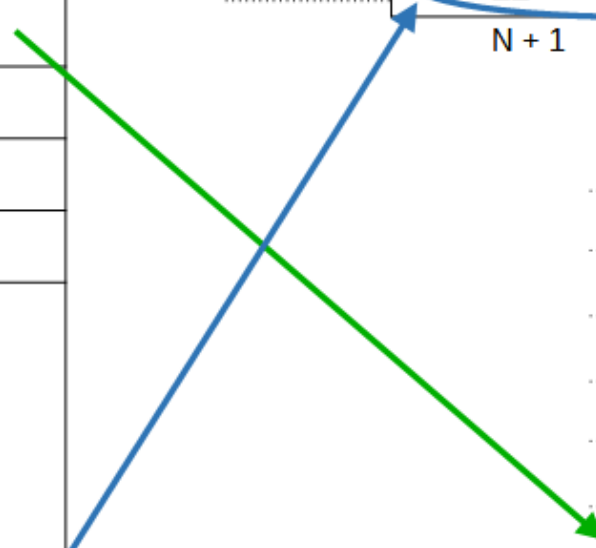


mov.w @R13+,2(R12)

PC	F84A
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	33	44
0204h	11	22
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



mov.w @R13+,2(R12)

PC	F84A
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0200
R14	1122
R15	4343

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22

N + 1 N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D

N + 1 N

0200h + 2h = 0202h

mov.w @R13+,2(R12)

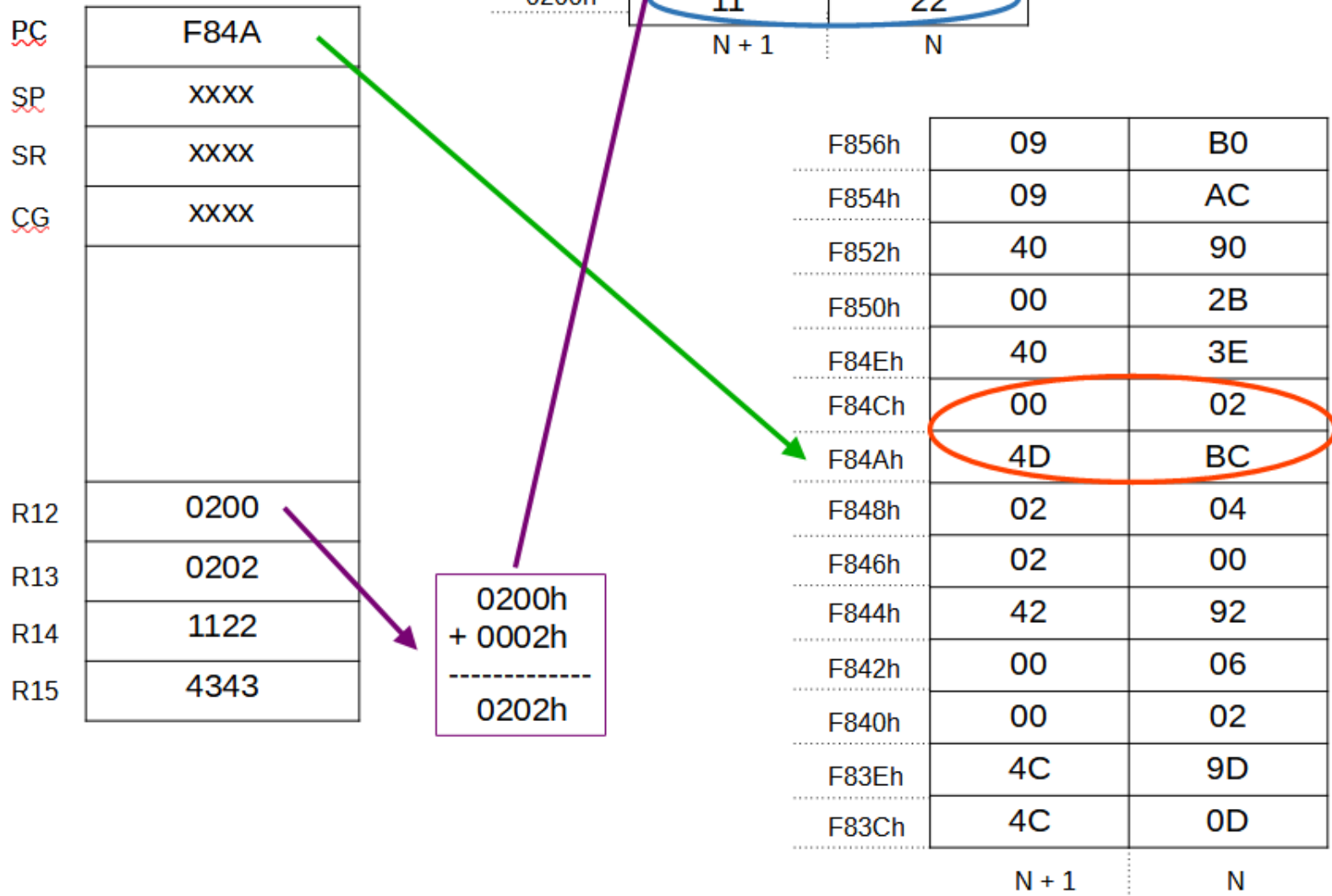
PC	F84A
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	1122
R15	4343

0206h	33	44
0204h	11	22
0202h	33	44
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

0200h
+ 0002h

0202h



`mov.w @R13+,2(R12)`

PC	F84A
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	1122
R15	4343

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22

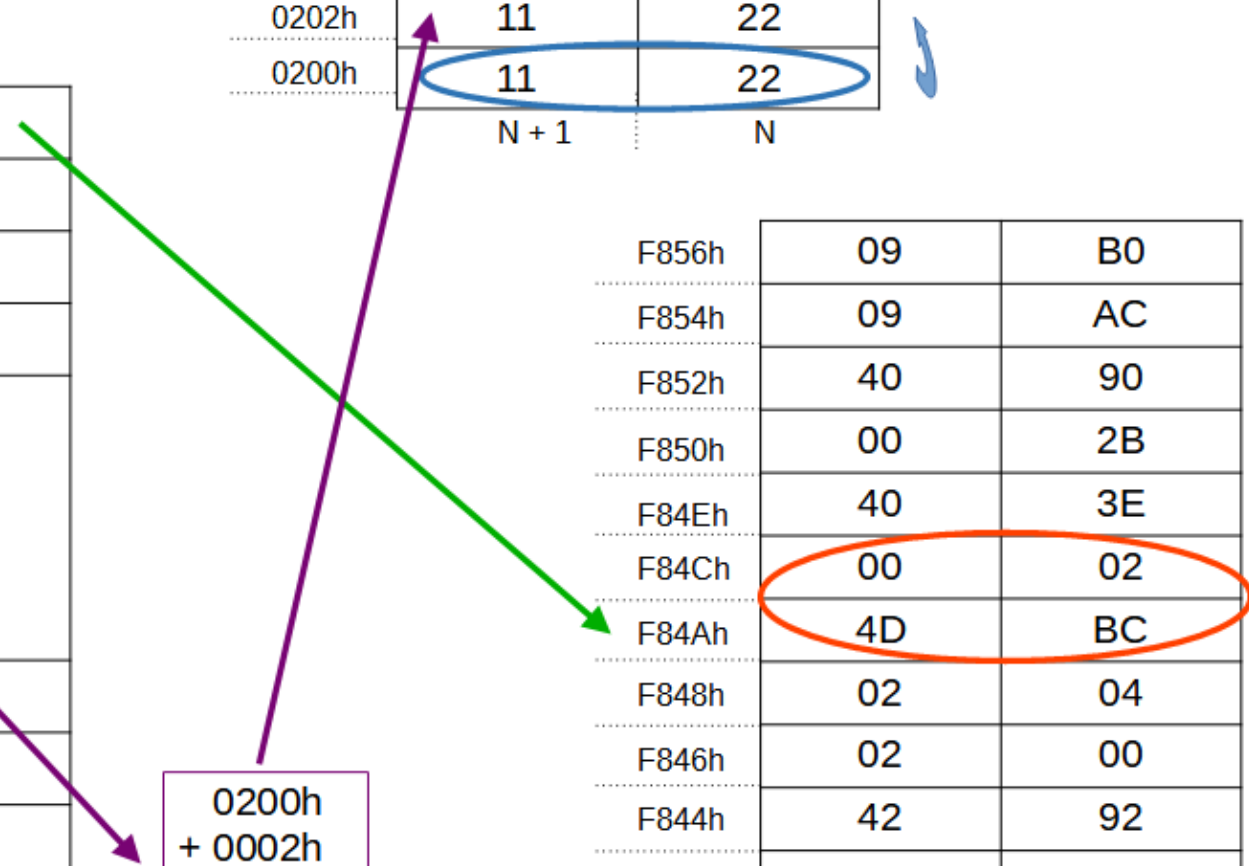
N + 1 N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D

N + 1 N

0200h
+ 0002h

0202h

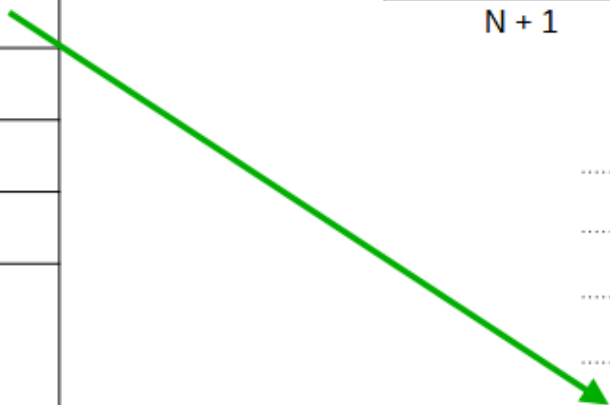


mov.w #43,R14

PC	F84E
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	1122
R15	4343

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

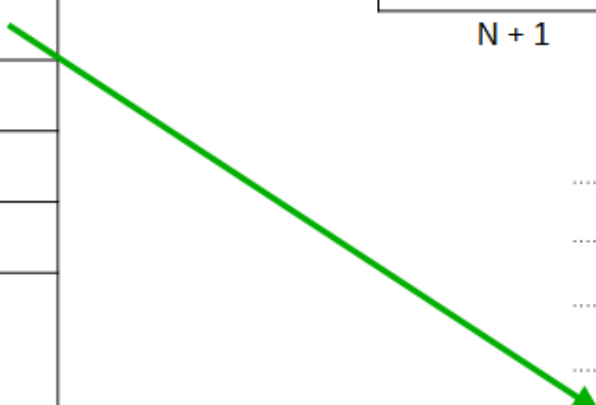


mov.w #43,R14

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

<u>PC</u>	F84E
<u>SP</u>	XXXX
<u>SR</u>	XXXX
<u>CG</u>	XXXX
R12	0200
R13	0202
R14	1122
R15	4343

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

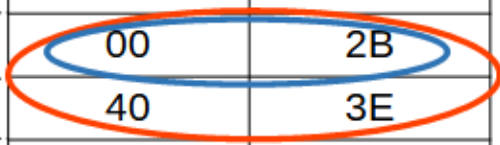
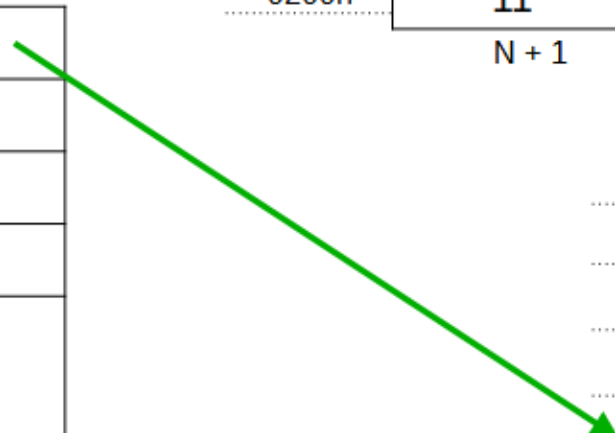


mov.w #43,R14

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

PC	F84E
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	1122
R15	4343

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

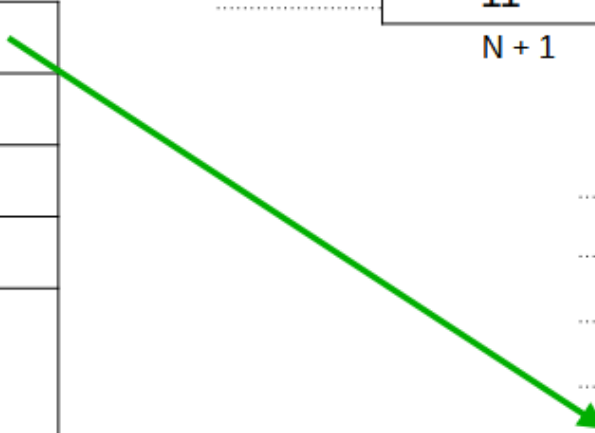


mov.w #43,R14

PC	F84E
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	002B
R15	4343

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



mov.w A,C

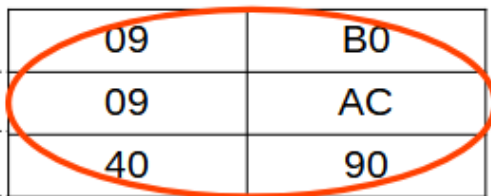
A=200h

C=206h

PC	F852
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	002B
R15	4343

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N



mov.w A,C

A=200h

C=206h

PC	F852
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	002B
R15	4343

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

F854h
+ 09ACh

0200h



mov.w A,C

A=200h

C=206h

PC	F852
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	002B
R15	4343

0206h	33	44
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

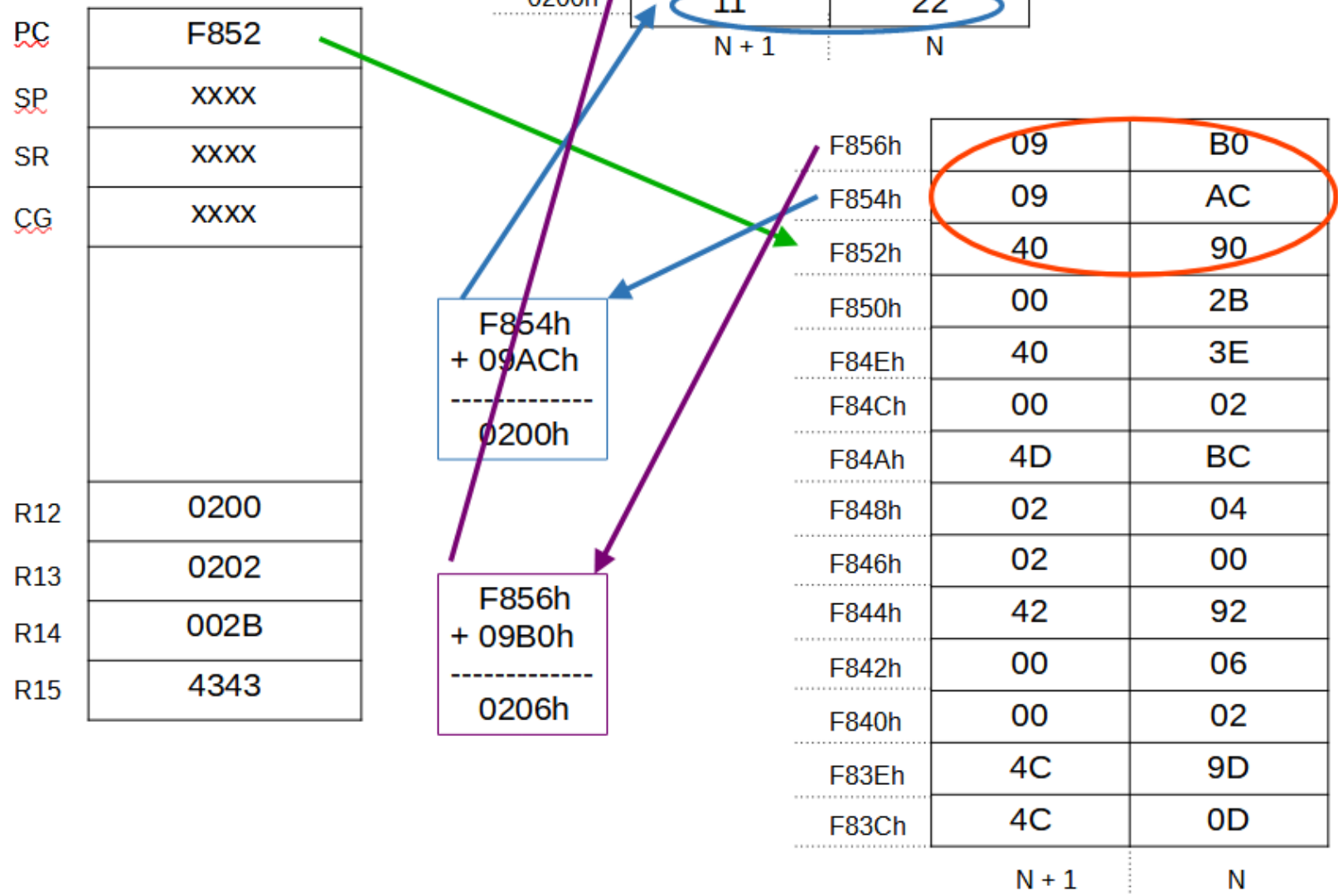
F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

F854h
+ 09ACh

0200h

F856h
+ 09B0h

0206h



mov.w A,C

A=200h
C=206h

PC	F852
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	002B
R15	4343

0206h	11	22
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

F854h
+ 09ACh

0200h

F856h
+ 09B0h

0206h

mov.w A,C

A=200h

C=206h

PC	F858
SP	XXXX
SR	XXXX
CG	XXXX
R12	0200
R13	0202
R14	002B
R15	4343

0206h	11	22
0204h	11	22
0202h	11	22
0200h	11	22
	N + 1	N

F856h	09	B0
F854h	09	AC
F852h	40	90
F850h	00	2B
F84Eh	40	3E
F84Ch	00	02
F84Ah	4D	BC
F848h	02	04
F846h	02	00
F844h	42	92
F842h	00	06
F840h	00	02
F83Eh	4C	9D
F83Ch	4C	0D
	N + 1	N

Ejercicio 9:

Src	Dst	No. of Cycles	Length of Instruction	Example
Rn	Rn	1	1	MOV R5, R8
	PC	2	1	BR R9
	x(Rn)	4	2	ADD R5, 4 (R6)
	EDE	4	2	XOR R8, EDE
	&EDE	4	2	MOV R5, &EDE
@Rn	Rn	2	1	AND @R4, R5
	PC	2	1	BR @R8
	x(Rn)	5	2	XOR @R5, 8 (R6)
	EDE	5	2	MOV @R5, EDE
	&EDE	5	2	XOR @R5, &EDE
@Rn+	Rn	2	1	ADD @R5+, R6
	PC	3	1	BR @R9+
	x(Rn)	5	2	XOR @R5, 8 (R6)
	EDE	5	2	MOV @R9+, EDE
	&EDE	5	2	MOV @R9+, &EDE
#N	Rn	2	2	MOV #20, R9
	PC	3	2	BR #2AEh
	x(Rn)	5	3	MOV #0300b, 0 (SP)
	EDE	5	3	ADD #33, EDE
	&EDE	5	3	ADD #33, &EDE
x(Rn)	Rn	3	2	MOV 2 (R5), R7
	PC	3	2	BR 2 (R6)
	TONI	6	3	MOV 4 (R7), TONI
	x(Rn)	6	3	ADD 4 (R4), 6 (R9)
	&TONI	6	3	MOV 2 (R4), &TONI
EDE	Rn	3	2	AND EDE, R6
	PC	3	2	BR EDE
	TONI	6	3	CMF EDE, TONI
	x(Rn)	6	3	MOV EDE, 0 (SP)
	&TONI	6	3	MOV EDE, &TONI
&EDE	Rn	3	2	MOV &EDE, R8
	PC	3	2	BRA &EDE
	TONI	6	3	MOV &EDE, TONI
	x(Rn)	6	3	MOV &EDE, 0 (SP)
	&TONI	6	3	MOV &EDE, &TONI

```

mov.w R12,R13
mov.w 2(R12),6(R13)
mov.w &A,&B
mov.w @R13+,2(R12)
mov.w #43,R14
mov.w A,C
    
```

- 1 ciclo de reloj
- 6 ciclos de reloj
- 6 ciclos de reloj
- 5 ciclos de reloj
- 2 ciclos de reloj
- 6 ciclos de reloj

26 ciclos de reloj

26 * 1us = 26us