

Tarea 1 Microprocesadores IE0623

Brynn Cortés Espínola C22422

1. Considere que dispone de un 9S12DG32. Se ha cargado INITRAM = %0010X00X, INITREG = \$18 e INITEE = \$10. Determine las direcciones de inicio y fin de cada bloque de memoria. Dibuje el mapa de organización de la memoria. Si se ejecuta la instrucción Ldaa \$2200 cuál bloque de memoria responde? Justifique su respuesta.

Se observa que el 9S12DG32 dispone la siguiente Memoria:

Registros: 1K → Comienzo: \$0000 → Permite saltos de 2K
final: \$03FF

EEPROM: 1K → Comienzo: \$0000 → Permite saltos de 2K
final \$07FF de \$0400 a \$07FF no usable

RAM: 2K → Comienzo: \$0000 → Permite saltos de 4K
\$0FFF parte baja (\$0000 a \$07FF) no usable

Flash: 32K

Para los Registros:

$$\text{INITREG} = \$18 = \%00011000$$

Se suma a la parte alta de la direcciones de Registros

Registros → Comienzo: \$1800
Final: \$1BFF

Para la EEPROM:

$$\text{INITEE} = \$10 = \%00010000$$

EEPROM → Comienzo: \$1000
Final: \$17FF

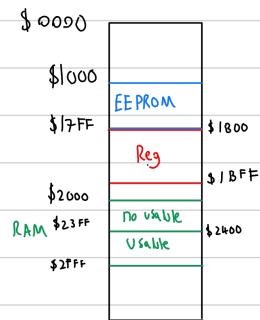
Para la RAM

$$\text{INITRAM} = \%0010\text{x}00x \\ = \$20$$

→ **SOFTURK**: No puek durr un salto de 2K por ende se toma como Cero.
→ **RAMHAL**: Como el bloque es de un tamaño de potencia de dos no importa el valor de este, se toma como cero.

RAM → (comienzo → \$2000
f:nu) → \$2FFF

Por lo que el mapa de memoria luce así:



Por lo que al realizar la instrucción `ldau $2200`, La dirección \$2200 se encuentra en el bloque de la RAM, pero se encuentra la parte baja, la cual en el 9S12 D632 no es visible.

- 2) Considere el siguiente segmento de programa, donde el contenido de la posición de memoria \$1000 es \$B3 y el contenido de posición de memoria \$1010 es \$5A

```

Ldaa $1000      # d = $B3
Ldab $1010      # y = $200
Ldy #2000       # Guarda d en $1010 = $B3, $2010 = $5A
Std 16, Y        # b = $10
Ldab #10
Bset b, Y, $55
Bclr 3, +Y, $37

```

- a. Luego de que este programa es ejecutado, cuáles son los números que quedan en las posiciones de memoria modificadas por BSet y BClr. Consideré que el contenido original de la posición para BClr es \$E5. Cuáles son las direcciones efectivas de esas posiciones de memoria.

Siguiendo el orden de ejecución del programa instrucción por instrucción se tiene que:

| d a y \$1000 \Rightarrow Cargar el contenido de la posición de memoria \$1000 en el acumulador a, $a = \$B3$

| d a b \$1010 \Rightarrow Cargar el contenido de la posición de memoria \$1010 en el acumulador b, $b = \$5A$

| ldy #\$2000 \Rightarrow Cargar en en índice y el valor de \$2000

| Std 16, y \Rightarrow guardar el contenido de d en $y+16 = \$2010$, d está compuesto por la parte alta por a y la baja b, por lo que $d = \$B35A$, Siguiendo la convención big endian, en \$2010 se almacena \$B3, y en \$2011 se guarda \$5A

| lab #\$10 \Rightarrow Cargar en el acumulador b el valor de \$10

| Bset b, y, \$55 \Rightarrow pone en 1 según la máscara. Se utiliza el modo de direccionamiento indexado por acumulador por ende la dirección efectiva del operando es $y+b = \$2010$

el cuál contiene \$B3 y al aplicar la máscara resulta en

$$\$B3 = \%10110011, \quad \$55 = \%01010101$$

$$\begin{array}{r}
 & 10110011 \\
 \text{bset} & \underline{01010101} \\
 & 11110111 = \$F7
 \end{array}$$

Por lo tanto en la posición de memoria \$2010 queda guardado \$F7

| Bclr 3, +Y, \$37 \Rightarrow pone en 0 según la máscara, se utiliza el modo de direccionamiento indexado pre-incremento, por lo cual el operando se ubica en $y+3 = \$2003$, la cual sería su dirección efectiva donde según el enunciado se encuentra \$ES, por lo cual al aplicar la operación se obtiene: $\$ES = \%1100101, \quad \$37 = \%00110111$

$$\begin{array}{r}
 & 1100101 \\
 \text{Bclr} & \underline{00110111} \\
 & 11000000 = \$C0
 \end{array}$$

Por lo tanto en \$2003 queda almacenado \$CO

- b. Determine cuántos ciclos de máquina requiere la ejecución de este programa. Incluya una tabla con los ciclos de máquina para cada instrucción y calcule el tiempo total a partir de esta tabla.

Instrucción	Moj. direccionamiento	Ciclos
ldaa \$1000	EXT	3
ldab \$1010	EXT	3
ldy #\$2000	IMM	2
std 16,Y	IDXI	3
ldab #\$10	IMM	1
bset b,Y,\$55	IDX	4
bclr 3,Y,\$37	IDX	4

Total : 20 ciclos

Cada ciclo dura alrededor de 125 ns, por lo tanto el programa dura en ejecutarse: $20 \cdot 125\text{ns} = 2500\text{ns} = 2,5\text{ ms}$

- a) Escriba el código de programa en ensamblador, que haga toggle a los bits pares (el LSB es el bit 0) del word en las posiciones de memoria cuya dirección efectiva es calculada por direccionamiento indirecto indexado por acumulador, utilizando el puntero X, el cual debe ser cargado por programa con \$3000. El programa debe cargar el acumulador D con \$0100. Además el resultado debe ser guardado en la dirección apuntada por el índice Y menos 3 sin alterar el puntero Y.

Se procede de 4 construir el diagrama de flujo y codificar

Código

ldx # \$3000

ldd # \$0100

lld [D, X]

eorq # \$55 ; toggle a la parte alta

eorb # \$55 ; toggle a la parte baja

s+3, Y
bra *, Fin

toogle-par

J ← \$3000

RR1 ← \$0100

RR4 ← ((J)+(RR1))

R1 ← R1 ^ \$55

R2 ← R2 ^ \$55

(X-3) ← RR4



En memoria

Posición de memoria	Contenido
\$30FF	\$10
\$3100	\$20
\$3101	\$00
\$3102	\$02
\$3103	\$43
\$3104	\$55
\$3105	\$AA

→ \$2000

\$14FO	\$F6
\$14FE	\$49

Y →

\$1500

valor almacenado en esa dirección luego de ejecutado el programa desarrollado en la parte a)

ldx # \$3000 ; X = \$3000

ldd # \$0100 ; D = \$0100

lld [D, X] ; Cargar D con el contenido de la dirección ubicada

en X+D = \$3100

eorq # \$55 ; Hacer toggle los bits pares de la parte alta del word

$$A = \$A3 = \%10100011$$

$$MASK = \$55 = \%01010101$$

$$\begin{array}{r} 10100011 \\ \times 01010101 \\ \hline 11101010 = \$F6 \end{array}$$

eorb # \$55 ; Hacer toggle los bits pares de la parte baja del word

$$B = \$1C = \%00011100$$

$$MASK = \$55 = \%01010101$$

$$\begin{array}{r} 00011100 \\ \times 01010101 \\ \hline 01001001 = \$49 \end{array}$$

s+3, Y ; Guardar en Y-3 = \$14FD

bra *, Fin

Posición de memoria	Contenido
\$1FFF	\$10
\$2000	\$A3
\$2001	\$1C
\$2002	\$2D
\$2003	\$A5
\$2004	\$A8
\$2005	\$05

- 4) Un arreglo es un conjunto de posiciones de memoria con direcciones consecutivas. Considera un arreglo de N valores ubicado a partir de la dirección Datos_Fuente, donde N es un valor numérico conocido (no es una variable) con $N < 200$. Los valores en el arreglo son de 1 byte con signo. Escriba un programa en lenguaje ensamblador para el S12 que revise los N valores y copie los valores que siendo negativos sean mayores o iguales que A3, a las localizaciones de memoria a partir de la posición Datos_Destino. Utilice direccionamiento indexado de post incremento tanto para la lectura de los valores como para su movimiento.

Con las anteriores especificaciones se procede a construir el diagrama de flujo y código.

Diagrama de flujo

Código

```
|ldx #Datos_Fuente ; Cargar puntero y acumulador
|ldy #Datos_Destino
|ljaq #N
```

loop

```
|ldab 1,xt
```

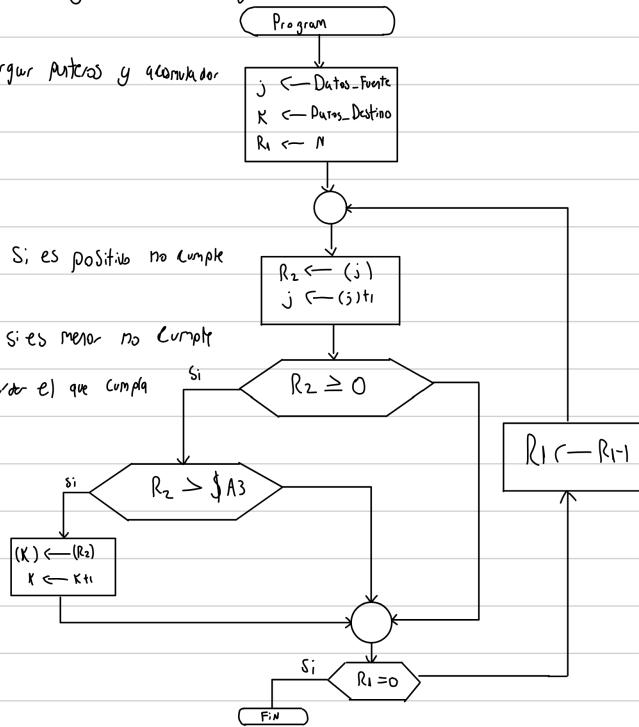
```
bpl no_cumple ! Si es positivo no cumple
cmpb #\$A3
```

```
blt no_cumple ; si es menor no cumple
```

```
stab 1,yt ; guardar el que cumpla
```

No Cumple DB NE a, Loop

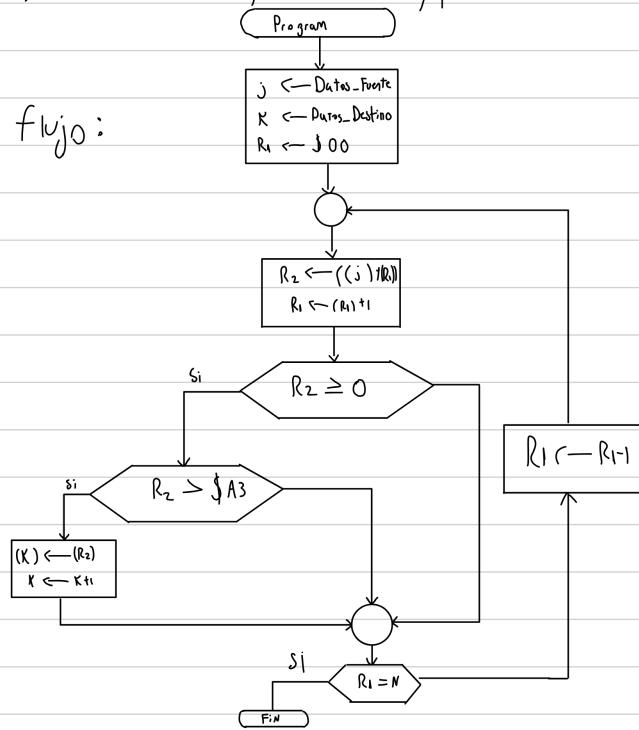
```
bra * ; Fin
```



- 5) Repita el ejercicio #4 utilizando direccionamiento indexado por acumulador A para la lectura de los datos. Observación: Note que en este caso el valor de los punteros permanece inalterado luego de la ejecución del programa.

Se volvió a repetir el mismo procedimiento, pero utilizando direccionamiento indexado por acumulador

Diagrama de flujo:



Diagrama

| dx #Datos_Fuente
| dy #Datos_Destino
C1 ra ; q = \$00

loop

| dab a, x

bpl no_cumple; saltar si es positivo

cmpb # \$A3

blt no_cumple; salta si es menor

stab i, y+

no_cumple cmpq # N

beq fin; termina loop

inca

bra loop

fin

bra *