

Ejercicios

1. *Reconfigure el puerto serial asíncrono a una velocidad diferente de 9600 baudios (Seleccionada por usted) considerando que el cristal de cuarzo es de 4.0 MHz. Suponga que debe emplear 9 bits en lugar de 8.*

```
BAUD EQU  $102B
SCCR2 EQU  $102D
SCCR1 EQU  $102C
```

```
ORG  $8000
LDD  $012C
STAA BAUD
STAB SCCR2
LDAA #$18
STAA SCCR1
```

*CONFIGURA EL PUERTO SERIAL.
 *BAUD 31.25K PARA CRISTAL DE 4MHz.
 *HABILITA RX Y TX, PERO INTERRUPTCION SOLO RX.
 *9 BITS.

```
LDAA #$FE
STAA DDRD
```

*CONFIG PUERTO D COMO SALIDAS (EXCEPTO PD0)
 *SEA ENABLE DEL DISPLAY PD4 Y RS PD3

```
LDAA #$04
STAA HPRI0
```

```
LDAA #$00
TAP
```

2. *En la tabla de la derecha se observa la equivalencia de un GB a Bytes. Si se desea transmitir un GB a 9600 baudios utilizando el puerto serial asíncrono del MC68HC11 y se sabe que utiliza un cristal de cuarzo de 4MHz, ¿Cuánto tiempo demoraría este proceso?*

bit (b):	<input type="text" value="8589934592"/>	Ejemplo: 5452595200
byte (B):	<input type="text" value="1073741824"/>	Ejemplo: 685000000
kilobyte (kB):	<input type="text" value="1048576"/>	Ejemplo: 650000
megabyte (MB):	<input type="text" value="1024"/>	Ejemplo: 650
gigabyte (GB):	<input type="text" value="1"/>	
terabyte (TB):	<input type="text" value="0.000976563"/>	

Teniendo una configuración de 9600 baudios, entonces tenemos que son 9600 bytes/segundo. Un gigabyte tiene 1073741824 bytes, por lo tanto, para obtener el tiempo total tenemos que:

$$\text{Tiempo} = \frac{1073741824}{9600} = 111848.1067 \text{ [s]}$$

Para obtener los ciclos de reloj tenemos que:

$$f = 4 \text{ [MHz]}$$

Para el MC68HC11 se debe dividir esta frecuencia entre cuatro para obtener su reloj interno

$$E = \frac{4M}{4} = 1 \text{ [MHz]}$$

Sabiendo que el inverso de la frecuencia es el tiempo, entonces:

$$t = \frac{1}{1M} = 1 [\mu s]$$

Por último, para obtener los ciclos de reloj que demorará el MC68HC11 se tiene que:

$$C_{reloj} = \frac{111848.1067}{1\mu} = 111.8481067 \text{ giga ciclos}$$

3. *Hacer una subrutina que transmita vía puerto serial su nombre completo a través del puerto serial (incluyendo espacios).*

NOMBRE

```
LDX  #SCSR
LDAA SCSR
LDAA #'A
STAA SCDR
C1
BRCLR      $00,X,#00 C1

LDX  #SCSR
LDAA SCSR
LDAA #'L
STAA SCDR
C2
BRCLR      $00,X,#00 C2

LDX  #SCSR
LDAA SCSR
LDAA #'E
STAA SCDR
C3
BRCLR      $00,X,#00 C3

LDX  #SCSR
LDAA SCSR
LDAA #'J
STAA SCDR
C4
BRCLR      $00,X,#00 C4

LDX  #SCSR
LDAA SCSR
LDAA #'A
STAA SCDR
C5
BRCLR      $00,X,#00 C5

LDX  #SCSR
LDAA SCSR
LDAA #'N
STAA SCDR
```

C6

BRCLR \$00,X,#00 C6

LDX #SCSR

LDAA SCSR

LDAA #'D

STAA SCDR

C7

BRCLR \$00,X,#00 C7

LDX #SCSR

LDAA SCSR

LDAA #'R

STAA SCDR

C8

BRCLR \$00,X,#00 C8

LDX #SCSR

LDAA SCSR

LDAA #'O

STAA SCDR

C9

BRCLR \$00,X,#00 C9

LDX #SCSR

LDAA SCSR

LDAA #\$20

STAA SCDR

C10

BRCLR \$00,X,#00 C10

LDX #SCSR

LDAA SCSR

LDAA #'G

STAA SCDR

C11

BRCLR \$00,X,#00 C11

LDX #SCSR

LDAA SCSR

LDAA #'O

STAA SCDR

C12

BRCLR \$00,X,#00 C12

LDX #SCSR

LDAA SCSR

LDAA #'M

STAA SCDR

C13
 BRCLR \$00,X,#00 C13

 LDX #SCSR
 LDAA SCSR
 LDAA #'E
 STAA SCDR

C14
 BRCLR \$00,X,#00 C14

 LDX #SCSR
 LDAA SCSR
 LDAA #'Z
 STAA SCDR

C15
 BRCLR \$00,X,#00 C15

 LDX #SCSR
 LDAA SCSR
 LDAA # \$20
 STAA SCDR

C16
 BRCLR \$00,X,#00 C16

 LDX #SCSR
 LDAA SCSR
 LDAA #'L
 STAA SCDR

C17
 BRCLR \$00,X,#00 C17

 LDX #SCSR
 LDAA SCSR
 LDAA #'U
 STAA SCDR

C18
 BRCLR \$00,X,#00 C18

 LDX #SCSR
 LDAA SCSR
 LDAA #'N
 STAA SCDR

C19
 BRCLR \$00,X,#00 C19

 LDX #SCSR
 LDAA SCSR
 LDAA #'A
 STAA SCDR

FIN

BRCLR \$00,X,#00 FIN
RTS

4. *¿Cuánto tiempo demora el envío de un byte a una velocidad de 9600 baudios en segundos y en ciclos de reloj, si se sabe que el procesador cuenta con un cristal de cuarzo de 8MHz?*

La velocidad de transmisión de 9600 baudios. Entonces, para obtener cuántos segundos tarda en transmitirse un byte tenemos que:

$$t_{byte} = \frac{1}{9600} = 104.167 [\mu s]$$

Si tenemos un cristal de 8 [MHz], el reloj interno es de:

$$E = \frac{8M}{4} = 2 [MHz]$$

Sabiendo que el inverso de la frecuencia es el tiempo, entonces:

$$t = \frac{1}{2M} = 0.5 [\mu s]$$

Por último, para obtener los ciclos de reloj que demorará el MC68HC11 se tiene que:

$$C_{reloj} = \frac{104.167}{0.5} = 208.334 \approx 209 \text{ ciclos}$$

5. *Escriba el código completo de un programa que transmita la palabra "ÉXITO" vía puerto serial cada que reciba un carácter "E" también por la vía del puerto serial. Considere que el sistema tiene que operar a 9600 baudios y que cuenta con un cristal de cuarzo de 8MHz.*

SCDR EQU \$102F
SCSR EQU \$102E
BAUD EQU \$102B
SCCR2 EQU \$102D
SCCR1 EQU \$102C
ORDN EQU \$0000
DDRD EQU \$1009
HPRIO EQU \$103C

ORG \$8000

INICIO

LDS #\$00FF
JSR SERIAL

CICLO

WAI
JMP CICLO

EXITO

LDX #SCSR
LDAA SCSR
LDAA #'E
STAA SCDR

C1
 BRCLR \$00,X,#00 C1
 LDX #SCSR
 LDAA SCSR
 LDAA #'X
 STAA SCDR

C2
 BRCLR \$00,X,#00 C2
 LDX #SCSR
 LDAA SCSR
 LDAA #'I
 STAA SCDR

C3
 BRCLR \$00,X,#00 C3
 LDX #SCSR
 LDAA SCSR
 LDAA #'T
 STAA SCDR

C4
 BRCLR \$00,X,#00 C4
 LDX #SCSR
 LDAA SCSR
 LDAA #'O
 STAA SCDR

FIN
 BRCLR \$00,X,#00 FIN
 RTS

SERIAL

LDD \$302C	*CONFIGURA EL PUERTO SERIAL.
STAA BAUD	*BAUD 9600 PARA CRISTAL DE 8MHz.
STAB SCCR2	*HABILITA RX Y TX, PERO INTERRUPCION SOLO RX.
LDAA #\$00	
STAA SCCR1	*8 BITS.

LDAA #\$FE	*CONFIG PUERTO D COMO SALIDAS (EXCEPTO PD0)
STAA DDRD	*SEA ENABLE DEL DISPLAY PD4 Y RS PD3

LDAA #\$04
 STAA HPRIO

LDAA #\$00
 TAP
 RTS

 * ATENCION A INTERRUPCION SERIAL

ORG \$F100

LDAA SCSR

```

LDAA SCDR
CMPA #E
BEQ  EXITO

RTI

*****
* VECTOR INTERRUPCION REAL TIME
*****

ORG  $FFD6
FCB  $F1,$00

*****
*RESET
*****

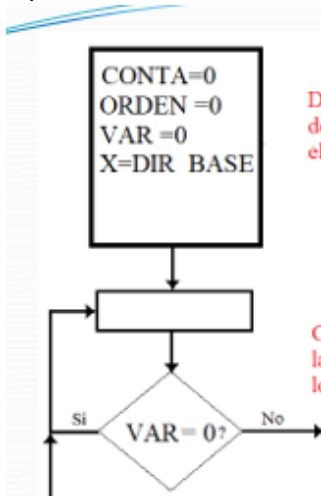
ORG  $FFFE
RESETFCB  $80,$00
*****

END  $8000

```

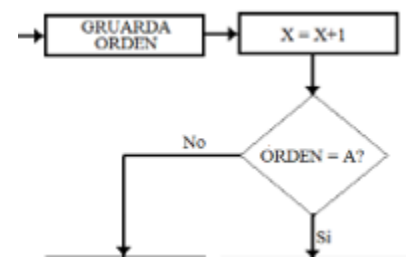
6. Compare el diagrama de flujo que aparece en esta presentación con el código del programa denominado "down.asc" que se anexó en el correo electrónico.

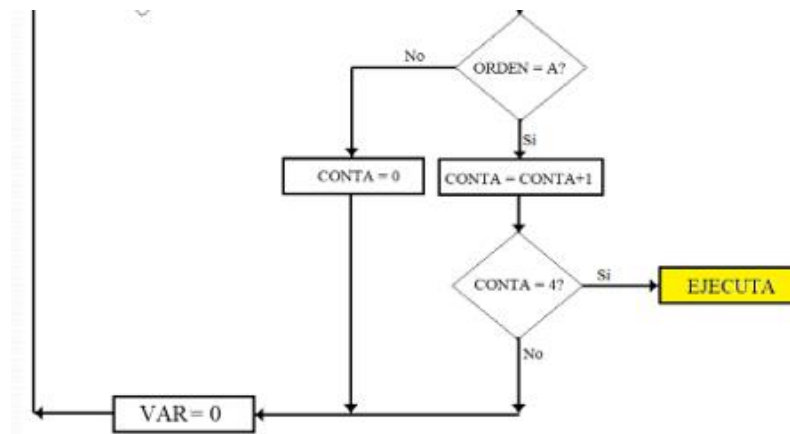
Como se puede observar en el diagrama de flujo, este corresponde a la parte donde se va a empezar a realizar la lectura del programa, es decir, una vez que ya se haya validado la parte correspondiente a la palabra "START", la cual define el inicio de la transmisión de datos.



En el programa, corresponde a la subrutina de ejecución llamada LOOP. Esta subrutina inicia con la instrucción LDAA VAR, la cual cargará en la memoria A el valor que se encuentra en la dirección de memoria que almacena VAR. Posteriormente, si este valor es cero, la instrucción BEQ volverá a LOOP, en caso contrario, continuará con las siguientes instrucciones. En un inicio, el valor al que apunta VAR es 0, por lo tanto, solo se saldrá de este LOOP cuando se interrumpa el programa debido a una recepción de información, porque cuando sucede esto, el valor al que apunta VAR ahora vale -1. Una vez que se sale de este ciclo, se vuelve a dejar en cero el valor al que apunta VAR con la instrucción CLR VAR, porque en instrucciones posteriores se volverá al inicio de la subrutina LOOP.

En esta parte se inicia con la instrucción LDAB ORDEN, STAB \$00,X e INX, las cuales sirven para que se vaya guardando el programa a partir de la dirección a la que apunta la memoria X, la cual inicia con un valor de \$0030 y que posteriormente irá incrementando. Por último, con CMPB se verifica si el contenido en el acumulador B es igual al código ASCII de la letra "A".





Para esta última sección, en caso de que la validación anterior sea falsa, se va a ir a la subrutina BORRA_A mediante la instrucción BNE BORRA_A. En la subrutina BORRA_A se tiene la instrucción CLR CONTA, la cual hace que la localidad a la que apunta la variable CONTA vuelva a ser cero.

En caso de que la validación sea verdadera, se incrementa el valor al que apunta la variable CONTA en uno con la instrucción INC CONTA. Este valor se carga en memoria con la instrucción LDAA CONTA y se verifica si es igual a 4 con CMPA #04, ya que esto significa que la transmisión de datos ha terminado y se puede ejecutar el programa con JMP DIR_BASE.

En caso de que no sea igual a 4, se vuelve a ejecutar la subrutina LOOP con BNE LOOP, y se repite el proceso anteriormente mencionado.

7. Simule el programa down.asc haciendo uso del simulador visto en clase THRSim11.exe (Para ello es necesario simular la transmisión de una cadena de caracteres, a través de la función "Serial transfer" que se encuentra en el menú de memoria, y describa el comportamiento del programa brevemente).

Al ejecutar el programa, la ejecución es un tanto rápida, por lo que no se aprecia mucho el inicio del programa donde se configura el puerto serial asíncrono. Se observa cómo se llega a la subrutina de ciclo y se mantiene el programa en esta, hasta que se trasmite algo por el puerto serial asíncrono, como se observa a continuación:

```

C:\Users\Alejandro\Documents\Tareas\Estructura y Programacion de Computadoras\down.LST

$8006 7F 00 01    CLR U1
$8009 7F 00 02    CLR U2
$800c 7F 00 03    CLR U3
$800f 7F 00 04    CLR U4

      CICLATE
$8012 86 3F      LDAA #'?
$8014 97 00      STAA ORDEN

      CICLO
$8016 96 00      LDAA ORDEN
$8018 81 3F      CMPA #'?
$801a 27 fa      BEQ  CICLO

$801c d6 01      LDAB U1
$801e 26 15      BNE  SIGUET

      BORRA
$8020 7F 00 01    CLR U1
$8023 7F 00 02    CLR U2
$8026 7F 00 03    CLR U3
$8029 7F 00 04    CLR U4
  
```



Conforme se van transmitiendo la información del puerto serial asíncrono, se observa como aquello que falta por transmitir se encuentra en color azul.



Una vez transmitido, se ve como el programa pasa a la subrutina LOOP, donde empieza a guardar la información revisada, para que finalmente todo termine cuando se pone AAAA. Esto es lo que se muestra cuando termina la ejecución del programa.

List window warning



The PC is placed on an address outside the list window.
The disassembler window is made active so you can still trace the program counter.

Aceptar

8. *Modifique la secuencia del programa que se transmite de modo que se quede permanentemente funcionando en RAM.*

Para el programa que se transmite, se debe de poner una subrutina cuyo nombre sea INICIO y que esté en la parte inicial del programa para que, posteriormente, al final del programa se tenga la instrucción JMP INICIO, con la finalidad de que se tenga un ciclo que abarque todo el programa, y, por ende, se quede funcionando permanentemente el programa en RAM.

9. *Modifique el programa en ensamblador para que acomode las cifras en el sentido opuesto, es decir de mayor a menor.*

El cambio que se debe de realizar es en la subrutina de COMPARA y en la subrutina de INTERCAMBIA. En la rutina de COMPARA se tiene la comparación de si el iésimo elemento es mayor al iésimo elemento más uno. Esto se realiza restándole al contenido del acumulador A el contenido del acumulador B. En el acumulador A ahora se cargará el elemento iésimo más uno y en el acumulador B el elemento iésimo, para tener la comparación contraria, que sería si el elemento iésimo es menor al iésimo elemento más uno.

COMPARA

LDAA \$00,Y
LDAB \$01,Y



COMPARA

LDAB \$00,Y
LDAA \$01,Y

Ahora, para la subrutina INTERCAMBIA, igual se cambiará el orden de las instrucciones, con la finalidad de que el elemento mayor quede a la izquierda y el elemento menor a la derecha.

INTERCAMBIA

STAB \$00,Y
STAA \$01,Y



INTERCAMBIA

STAA \$00,Y
STAB \$01,Y