

Trabajo Práctico 3

A- Lenguajes de Programación

✓ **A.01** Clasifique los lenguajes de programación y describa las características de cada categoría (lenguaje de máquina, lenguaje de bajo nivel, lenguajes de alto nivel).

✓ **A.02** Describa la tarea de un compilador y la tarea de un intérprete. ¿En qué se diferencian?

✓ **A.03** Describa la tarea del ensamblador. ¿En qué se diferencia de un compilador?

✓ **A.04** Explique la utilidad de las siguientes palabras reservadas

EQU

RMB

ORG

DB

DW

FCC

FCB

¿En qué se diferencian de las instrucciones ASM?

✓ **A.05** Explique el concepto de "etiqueta" en un programa ASM.

A.06 Explique si es posible pasar de un programa en lenguaje de máquina a su representación en ASM. En caso de serlo indique los pasos a realizar. ¿Ocurre lo mismo pasando de lenguaje de máquina a un lenguaje de alto nivel?

A.07 Explique que es una subrutina. Explique cómo es que se vuelve de una subrutina al flujo de programa original al momento de llamar a una subrutina. Indique si es posible que una subrutina se llame a sí misma (recursividad) y qué problemas podrían ocurrir (en particular el desborde de pila).

B- ASM

B.01 Indique que hace el procesador 68HC11 cuando ejecuta las siguientes instrucciones. Indique el modo de direccionamiento (motorola y genérico). Indique si la instrucción es válida y en caso de no serlo indique el problema. En el caso de ser válida codifique la misma en lenguaje de máquina e indique si afecta los bits del CCR (y en qué forma).

LDAB #\$25
LDAB \$25
LDAB 25
LDAB \$2525
LDAB #\$2525
LDAB \$25,X
LDD \$225,X
LDAA #\$2525
ADDA #\$EA
ADDA \$EA
ADDA \$EA,X
ADDA \$46EA
ADDA \$EEAA,X
CBA
BCC \$10
BCC #\$10
BCC Etiqueta
BGE \$F7
BSR \$65

B.02 Para una unidad de procesos 68HC11, indicar los contenidos de los acumuladores A y B, y de las direcciones de memoria 00F5, 00F7 y 00F8 después de la ejecución de las siguientes instrucciones:

INICIO	CLRA	
	LDAB	\$F5
Otro	ADDA	\$F7
	DECB	
	BNE	otro
	STAA	\$F8
FIN	BRA	FIN

si los contenidos iniciales de las direcciones de memoria son respectivamente

00F5: 08

00F7: 12

Indicar qué hace el programa.

B.03 Escriba un programa para el 68HC11 simulado en THRSIM11, donde utilice los switches y LEDS. En los switches se lee un valor entre 00 y FF. Si el valor es 0, no se enciende ningún led. Si el valor está entre 1~32 se enciende el led más a la derecha. Si el valor está entre 33~64 encienda los dos leds más hacia la derecha. Si el valor está entre 65 ~96 encienda los tres leds más a la derecha... y así. Esto utiliza los leds tipo vumetro basado en el valor de entrada. Existe un video de ayuda indicando cómo utilizar los LEDS y switches.



B.04 Escriba un programa para el 68HC11 que resuelva la ecuación. La variable A existe en RWM y debe tener tamaño suficiente como para almacenar el resultado. El resto (b,c,d,e y f) son de 8 bits sin signo y pueden estar definidos como constantes o como valores almacenados en ROM.

$$a = \frac{b^2 - c^3}{(d - 2) * (e + f)}$$

Recomendamos resolverlo de manera estructurada y luego como una arquitectura orientada al stack (en este último caso puede guardar solo los 8 bits menos significativos del resultado).

B.05 Escriba una subrutina para el HC11 (llamada CANTNEGFIJA) que reciba en IX la dirección de comienzo de un vector. El mismo tiene una longitud fija de 20 elementos. Los elementos que componen el vector son números signados en CB (-32768 hasta 32767). La subrutina debe retornar la cantidad de números negativos del vector en algún registro.

B.06 Escriba una subrutina para el HC11 (llamada CANTNEGVAR) que reciba en IX la dirección de comienzo de un vector. El mismo tiene una cantidad variable de elementos y se indica el fin del mismo con un elemento 0. Los elementos que componen el vector son números signados en CB (-32768 hasta 32767). La subrutina debe retornar la cantidad de números negativos del vector en algún registro.

B.07 Repita los ejercicios A05 y A06 pero contando números positivos.

B.08 Escriba una subrutina para el HC11 (llamada PROMPOS) que reciba en IX la dirección de comienzo de un vector. El mismo tiene longitud variable (máximo 128 elementos, mínimo 1) y el fin del mismo se indica guardando un 0. Los elementos del vector son números signados en CB

(-128 a 127). La subrutina debe calcular el promedio de los valores positivos y devolver el mismo mediante un acumulador.

B.09 Escriba una subrutina para el HC11 (llamada BUSCAMIN) que reciba en IX la dirección de comienzo de un vector. El mismo tiene longitud variable y el fin del mismo se indica guardando un 0. Los elementos que componen el vector son números signados en CB (-32768 hasta 32767). La subrutina debe retornar el menor elemento encontrado en el vector en un registro.

B.10 Escriba una subrutina para el HC11 (llamada POSMIN) que reciba en IX la dirección de comienzo de un vector. El mismo tiene longitud variable y el fin del mismo se indica guardando un 0. Los elementos que componen el vector son números signados en CB (-32768 hasta 32767). La subrutina debe retornar la dirección del menor elemento encontrado en el vector en un registro.

B.11 Escriba una subrutina para el HC11 (llamada COMP2BEB) que reciba en IX la dirección más baja donde se encuentra almacenado un número en formato big-endian. La cantidad de bytes que componen al mismo se informa en el registro B (no mayor a 64). Debe complementar a la base el número.

B.12 Escriba una subrutina para el HC11 (llamada COMP2LEB) que reciba en IX la dirección más baja donde se encuentra almacenado un número en formato little-endian. La cantidad de bytes que componen al mismo se informa en el registro B (no mayor a 64). Debe complementar a la base el número.

B.13 Escriba una rutina para el HC11 (llamada CONTCANT) que reciba en IX la dirección de comienzo de un vector. El mismo es de tamaño variable, siendo el último elemento un 0. Cada elemento es de 16 bits. La subrutina recibe en el registro IY un valor de 16 bits. Debe retornar en el registro IX la cantidad de veces que aparece el valor informado en IY dentro del vector.

B.14 Escriba una rutina para el HC11 (llamada CONTNCANT) que reciba en IX la dirección de comienzo de un vector. El mismo es de tamaño variable, siendo el último elemento un 0. Cada elemento es de 16 bits. La subrutina recibe en el registro IY un valor de 16 bits. Debe retornar en el registro IX la cantidad de veces que NO aparece el valor informado en IY dentro del vector.

B.15 Escriba una rutina para el HC11 (llamada INVIERTE) que reciba en IX la dirección de comienzo de un vector. El mismo es de tamaño variable (máximo 64 elementos), siendo el último elemento un 0. Cada elemento es de 16 bits. La subrutina recibe en IY una dirección en RWM en donde deben escribirse los elementos del vector pero en orden inverso. Ej: si el vector tiene como elementos 1,2,3,0, debe escribir 3,2,1,0.

B.16 Escriba una rutina para el HC11 (llamada ALIGN4) que reciba en IX la dirección de comienzo de un vector. El mismo es de tamaño variable siendo el último elemento un 0. Cada elemento es una dirección de 16 bits tomada de una computadora que direcciona al byte con tamaño de

palabra 4 bytes. Retorne en el registro IY la cantidad de direcciones que se encuentran alineadas para lecturas de palabras completas.

B.17 Escriba una rutina para el HC11 (llamada C24IMPBE) que reciba en IX la dirección de comienzo de un vector. El mismo es de tamaño variable siendo el último elemento un 0. Cada elemento es un número de 24 bits almacenado en big-endian. La subrutina debe retornar en el registro IY la cantidad de elementos impares que contiene el vector.

B.18 Escriba una rutina para el HC11 (llamada CONTMAY) que reciba en IX la dirección de comienzo de una cadena de hasta 1000 caracteres (ASCII 8 bits) que termina con NULL (0x00). Retorne en un registro la cantidad de mayúsculas.

B.19 Escriba una rutina para el HC11 (llamada CONTNUM) que reciba en IX la dirección de comienzo de una cadena de hasta 1000 caracteres (ASCII 8 bits) que termina con NULL (0x00). Retorne en un registro la cantidad de caracteres numéricos.

B.20 Escriba una rutina para el HC11 (llamada CONTALPHA) que reciba en IX la dirección de comienzo de una cadena de hasta 1000 caracteres (ASCII 8 bits) que termina con NULL (0x00). Retorne en un registro la cantidad de caracteres alfanuméricos.

B.21 Escriba una subrutina para el procesador HC11 (llamada ITOAU16BE (integer to ASCII)) que reciba en IX la dirección de comienzo de un número de 16 bits sin signo, almacenado en BIG ENDIAN y que lo convierta a una cadena de caracteres ASCII, a partir de la dirección indicada por el registro IY.

B.22 Escriba una subrutina para el procesador HC11 (llamada ATOIS16BE (ASCII to integer)) que reciba en IX la dirección de comienzo de una cadena de caracteres ASCII (cuyo primer caracter puede ser el signo '-'), y que convierta a un entero con signo de 16 bits almacenado en BIG ENDIAN, a partir de la dirección indicada por el registro IY. Los negativos deben representarse en CB.

B.23 Escriba una rutina para el procesador HC11 (llamada SUMAFLOAT) que reciba dos números codificados en IEEE754 simple precisión (las direcciones del bit más significativo de ambos apuntados por IX e IY respectivamente). Debe realizar la suma de ambos números y devolver el resultado mediante el stack (De forma que quien llame a la rutina obtenga en 4 lecturas del stack el valor). Puede ignorar los casos especiales de exponente igual a 00000000 y 11111111.

B.24 Desensamblar el siguiente código de máquina del procesador HC11 teniendo en cuenta que el área de memoria comienza en la dirección \$8000. En los saltos relativos (si los hubiera) colocar la etiqueta correspondiente. El valor inicial del PC = \$8000.

8000 7F 00 02 7F 00 03 7F 00 04 7F 00 05 96 00 D6 01 1B 24 03 7C 00 02 97 03 96 00
9B 01 24 03 7C 00 04 97 05 20 FE

B.25 Considere el siguiente fragmento de memoria de un microcontrolador Motorola 68HC11, del que además se indica el estado inicial de los registros: IX = 2500; PC = 4000; D= 0519

Dirección	Contenido
2500	4C
2501	08
2502	5C
2503	18
2504	08
....	
3FFF	08
4000	A6
4001	01
4002	08
4003	28
4004	02
4005	C6
4006	08
4007	4C

Determine los valores finales de cada registro (IX, PC, A, B) luego de ejecutarse la porción de código indicada:

IX _____

PC _____

A _____

B _____

B.26 Un microcontrolador 68HC11 presenta los siguientes valores en su estado funcional antes de comenzar a ejecutar la siguiente instrucción:

PC: C000; SP=00FF; IX=2000; IY=3000; A=01; B=02

El contenido de la memoria a partir de la dirección C000 es el siguiente:

7f 00 01 7f 00 02 96 00 80 64 25 05 7c 00 01 20

f7 8b 64 80 0a 25 05 7c 00 02 20 f7 8b 0a 97 03

20 fe

Desensamble el programa (una línea por instrucción, no por byte) y complete con las etiquetas que considere necesarias.

B.27 *Escriba un programa ASM para el 68HC11 donde el origen sea \$C000 y con la instrucción DB almacene los bytes que forman el programa del punto B26. Luego cargue el vector de reset con la dirección \$C000 y ejecute el programa. Utilice la ventana de dissassembler para seguir la ejecución del mismo.*