

Introducción a la Arquitectura de Sistemas  
Guía de Trabajos Prácticos N° 4

Programación en lenguaje ensamblador CODE-2

1. Escribir una subrutina que comience en la posición 100 h de la memoria que determine si el valor almacenado en R9 es negativo. En caso de serlo, que lo reemplace por su valor absoluto.
2. Escribir una subrutina que comience en la posición 120 h de la memoria que calcule un promedio entero entre los valores de los registros R9 y R10 y almacene el resultado en R11.
3. Implementar las operaciones NOT, OR, NOR, AND y XOR en sendas subrutinas que tomen como entradas los registros R9 y R10, y dejen su salida en R11. Se puede utilizar R12 y R15 como registros auxiliares.
4. Implementar una suma de 32 bits en una subrutina que tome los valores de R7 y R8 como la parte alta y baja respectivamente del primer operando, y los registros R9 y R10 como el segundo operando. El resultado será devuelto en los registros R11 y R12 (parte alta y baja respectivamente). El flag de acarreo de la ALU debe volver seteado si se produjo acarreo al sumar las partes altas.
5. Dado el siguiente fragmento de código ensamblador CODE2 que involucra las variables A, B y C almacenadas en las posiciones 2002 h, 2003 h y 2004 h resuelva los siguientes ítems:
  - a) Describa con sus palabras o con pseudocódigo la función. Eleve el nivel de abstracción tanto como pueda. Es una subrutina o un programa?
  - b) Indique el tipo de formato de cada instrucción y realice el volcado de memoria del programa completo.
  - c) Escriba las microinstrucciones de la instrucción 'ld' y 'shra'.

1er caso	2do caso
0010 llo, Rd, 02	0010 llo, Rd, 02
0011 lhi Rd, 20	0011 lhi Rd, 20
0012 ld R1, 0	0012 ld R1, 0
0013 ld R2, 1	0013 ld R2, 1
0014 adds R3, R1, R2	0014 adds R3, R1, R2
0015 shra R3	0015 shl R3
0016 st 2, R3	0016 shl R3
0017 HALT	0017 st 2, R3
	0018 RET

6. Escribir un programa que lea 15 valores del puerto de entrada 1F<sub>n</sub> y que arme con ellos un arreglo que comience en la posición 1000<sub>n</sub> de la memoria. Agregar un 0 como último elemento.
7. Escribir un programa que recorra el arreglo que comienza en la posición 1020 h de la memoria y transforme todos los elementos negativos en positivos. Utilice la subrutina escrita en el ejercicio 1. Se asume que el arreglo finaliza cuando se encuentra el primer elemento con valor 0.
8. Escribir un programa que calcule el promedio elemento por elemento de los arreglos que comienzan en la posición 1000 h y 1020 h de la memoria respectivamente, y el resultado lo almacene en un nuevo arreglo que comienza en la posición 1040 h de la memoria. Todos los arreglos deben tener 0 como último valor, y el proceso debe concluir cuando se encuentre un 0 en cualquiera de los dos arreglos de entrada.
9. Escribir un programa que se copie a sí mismo 40 h posiciones de memoria más abajo de donde comienza, y luego haga un salto al punto donde comienza la copia. ¿Cuándo finaliza de ejecutarse este programa? ¿Qué tipo de programas actúan de esta manera?
10. Tenemos en memoria un conjunto con 0Fh datos, se puede considerar como una tabla. El primer dato (inicio de la tabla) se encuentra en la posición 00C0h. Hacer un programa que vaya leyendo los datos de la tabla, cuente cuantos números positivos, negativos y ceros hay en el conjunto. Cuando llegue al final, debe mostrar por OP1, primero cuantos números negativos había, a continuación cuantos ceros y por ultimo cuantos positivos había. Cargar el programa a partir de la posición 0010h.
11. SOLO PARA INGENIERIA...Escribir un programa que permita recorrer una lista vinculada, sumando todos los elementos de la misma y enviando al final el resultado por el puerto de salida 14. Cada nodo de la lista está formado por dos elementos, ocupando un total de 4 bytes de memoria. El primer elemento es un número entero de 2 bytes que corresponde al valor del nodo. El segundo elemento es un puntero de 2 bytes que indica la posición en la memoria del siguiente nodo de la lista. El último nodo tendrá un puntero nulo (el puntero indica 0). Asumir que el primer nodo de la lista se encuentra en la posición 2000 de la memoria.