



Manual de Prácticas Microprocesadores

División: Ingeniería Eléctrica

Departamento: Ingeniería Electrónica

Programación en lenguaje Assembly (Ensamblador) 2

N.º de práctica: 05

Nombre completo del alumno		Firma
nombre		
N.º de brigada:	Fecha de elaboración	Grupo:



Manual de Prácticas Microprocesadores

División: Ingeniería Eléctrica

Departamento: Ingeniería Electrónica

1. Seguridad en la Ejecución

	Peligro o fuente de energía	Riesgo asociado
1	Manejo de Corriente Alterna	Electrochoque
2	Manejo de corriente Continua	Daño al equipo

2. Objetivos de aprendizaje.

El alumno aprenderá a utilizar las instrucciones de operaciones aritméticas y lógicas, así como emplear saltos condicionales y llamados a subrutinas.

3. Material y equipo.

Calculadora en modo “Programador”. Tarjeta de desarrollo y ambiente IDE CCS.

4. Actividad previa.

- ¿Cómo se obtiene la representación binaria de un número negativo en el formato de complemento a 2?
- ¿Cuál es la longitud en bytes del resultado de una multiplicación no signada (unsigned multiplication) al multiplicar: a) dos valores de 16 bits cada uno? b) dos valores de 32 bits cada uno?
- ¿Qué hace el salto condicional “BEQ etiqueta1” ?
- ¿Qué hace el salto “BL etiqueta2” ?
- Para cada uno de los dos saltos anteriores, ¿se debe regresar al lugar del salto?, ¿cómo?
- Examine el comportamiento de los segmentos del código proporcionado. Con ayuda de la calculadora, compruebe el resultado de las operaciones.

5. Desarrollo.

Escriba un programa que funcione como una calculadora de números enteros de hasta 32 bits, con operaciones de **Suma, Resta, Multiplicación, AND, OR, XOR**.

Para elegir la operación a realizar, **desde el depurador** se escribirá un valor en **R0 del 1 al 6 (1:suma, 2:resta, etc.)**. El programa debe hacer llamados a **dos subrutinas**: una para carga de operandos y otra para guardar resultado.

Los dos operandos se encuentran definidos en memoria **RAM** y se deben colocar en **R1 y R2**. **El resultado** se escribe también en memoria **RAM**, usando **R3**.

Por tanto se debe hacer acceso de lectura y escritura (LDR, STR).

Los operandos se pueden modificar en cualquier sobre-escribiendo su valor en memoria RAM.

6. Cuestionario.

- i. Después de ejecutar la instrucción **SUBS**, ¿qué signo tiene el resultado cuando las bandera **C = 0**, y cuando **C=1**? En cada caso, qué valor debe tener la bandera **N**?

Para un valor en un Registro, en lenguaje ensamblador...

- ii. ¿Cómo se enciende un bit específico sin afectar a los demás? (2 maneras)
- iii. ¿Cómo se apaga un bit específico sin afectar a los demás? (2 maneras)
- iv. ¿Cómo se invierte el estado de un bit específico?
- v. Al emplear instrucción de comparación **CMP**, ¿Qué nos dice la bandera **Z**?
- vi. Se quiere saber si los 4 LSB de un byte son '0', ¿cómo emplearía la instrucción **TST** y qué condición/bandera se debe cumplir/establecer?
- vii. Se quiere saber si los 4 MSB de un byte son '1', ¿cómo emplearía la instrucción **TEQ** y qué condición/bandera se debe cumplir/establecer?
- viii. Para qué sirve la directiva **.retain**

7. Conclusiones.

De cada estudiante y de carácter obligatorio en el informe correspondiente.

8. Bibliografía