

## PRÁCTICA 2: MANEJO DEL REPERTORIO DE INSTRUCCIONES 8086

### OBJETIVOS:

1. Conocer la estructura de la memoria basada en la segmentación del i8086
2. Conocer los tipos de datos básicos en la arquitectura 8086: byte y word
3. Conocer los formatos de instrucción del procesador i8086: sin operando, mono-operando, bi-operando (origen y destino)
4. Conocer las instrucciones de mayor uso del 8086 como son: aritméticos, lógicas, condicionales, saltos y desplazamientos

### CONTENIDOS:

1. Segmentos en el procesador I8086
2. Tipos de datos básicos en la arquitectura 8086
3. Instrucciones: Formatos y tipos
4. Operadores aritméticos básicos: ADD, ADC, SUB, SBB, MUL, IMUL, DIV, IDIV, INC, DEC
5. Operadores lógicos básicos: NEG, NOT, AND, OR, XOR
6. Operadores condicionales básicos y otros: CMP, XCHG
7. Etiquetas e instrucciones de saltos básicas: JMP, JE, JG, JGE, JL, JLE, JNE, LOOP
8. Operadores de desplazamiento: SHR, SHL, ROR, ROL

### EJEMPLOS:

1. Mover al registro AX el valor 5 y sumar AX con el valor inmediato 9
2. Negar el contenido del registro AX y sumarlo consigo mismo
3. Intercambiar los valores de las posiciones de memoria 0 y 1.
4. Mover a la posición de memoria indicada por la variable **vector** (definida en el segmento de datos) el contenido de AL y restar a BL el contenido de **vector**
5. Mover a AL el valor 25 y multiplicar AL por el contenido de la variable **dato** (definida e inicializada en el segmento de datos al valor 2)
6. Dividir AX entre la variable **numero** (definida e inicializada en el segmentos de datos a valor 10). Si quisiéramos dividir entre dos, ¿qué instrucción podemos utilizar que no sea DIV?.
7. Poner a cero el contenido del registro CX sin usar MOV.
8. Introducir en BX el valor 100h. Guardar en la dirección de memoria que apunta BX y siguientes los caracteres "a", "b", "c", "d", "e", "f", "g", "h", "i", "j". Usar la instrucción LOOP.
9. Definir en el segmento de datos una variable de tipo byte **letras**, donde guardaremos los caracteres del ejercicio anterior.

### EJERCICIOS:

1. Utilizar las instrucciones lógicas para: hallar el c-1, poner a cero, poner a uno, enmascarar los 4 bits mas significativos, de una variable de tipo byte definida e inicializada al valor 0AFEDh en el segmento de datos.
2. Realizar un programa en ensamblador que calcule la media aritmética de tres valores de tipo *byte* definidos e inicializados en el segmento de datos, con los nombres: **dat1**, **dat2** y **dat3**
3. Modificar el programa anterior para que calcule la media aritmética de N números de tipo byte guardados en la variable **valores** en el segmento de datos. El valor de N estará guardado en la variable **num** definida en el segmento de datos.
4. Utilizando **LOOP**, realizar un bucle que multiplique un número **num1** por si mismo el número de veces indicado en la variable **veces**.
5. Realizar un programa en ensamblador que realice la tabla de multiplicar de todos los números entre el 1 y el 10. Guardar el resultado en la variable **tabla** inicializada en el segmento de datos al valor 0. Definir las variables necesarias en el segmento de datos.
6. Dado un número cualquiera almacenado en la variable **numero**, calcular todos sus divisores. Los divisores serán almacenados en memoria a partir de la posición de memoria 0. (Se puede usar una variable, por ejemplo, **divisores**, pero esta debe apuntar a la dirección 0 del segmento de datos)