PRACTICA 1

Subrutinas y pasaje de parámetros

Objetivos: Comprender la utilidad de las subrutinas y la comunicación con el programa principal a través de una pila. Escribir programas en el lenguaje assembly del simulador MSX88. Ejecutarlos y verificar los resultados, analizando el flujo de información entre los distintos componentes del sistema.

1) Multiplicación de números sin signo.

Escribir un programa que calcule el producto entre dos números sin signo almacenados en la memoria del microprocesador:

1.1) Sin hacer llamados a subrutinas, resolviendo el problema desde el programa principal;

1.2) Llamando a una subrutina MUL para efectuar la operación, pasando los parámetros por valor desde el programa principal a través de registros;

1.3) Llamando a una subrutina MUL, pasando los parámetros por referencia desde el programa principal a través de registros.

1.1)

```
      ; Memoria de Datos
      ORG 1000H
NUM1  DB    5H
NUM2  DB    3H
      ; Memoria de Instrucciones
      ORG 2000H
      MOV   AL, NUM1
      CMP   AL, 0
      JZ    FIN
      MOV   AH, 0
      MOV   DX, 0
      MOV   CL, NUM2
LOOP: CMP   CL, 0
      JZ    FIN
      ADD   DX, AX
      DEC   CL
      JMP   LOOP
FIN:  HLT
      END
```

1.2)

```
      ; Memoria de Datos
      ORG 1000H
NUM1  DB    5H
NUM2  DB    3H
      ; Memoria de Instrucciones
      ORG 3000H   ; Subrutina MUL
MUL:  CMP   AL, 0
      JZ    FIN
      CMP   CL, 0
      JZ    FIN
      MOV   AH, 0
      MOV   DX, 0
LAZO: ADD   DX, AX
      DEC   CL
      JNZ   LAZO
FIN:  RET

      ORG 2000H   ; Programa principal
      MOV   AL, NUM1
      MOV   CL, NUM2
      CALL  MUL
      HLT
      END
```

1.3)

```
      ; Memoria de datos
      ORG 1000H
NUM1  DW    5H    ; NUM1 y NUM2 deben ser mayores que cero
NUM2  DW    3H

      ; Memoria de Instrucciones
      ORG 3000H   ; Subrutina MUL
MUL:  MOV   DX, 0
LAZO: MOV   BX, AX
      ADD   DX, [BX]
      PUSH  DX
      MOV   BX, CX
      MOV   DX, [BX]
      DEC   DX
      MOV   [BX], DX
      POP   DX
      JNZ   LAZO
      RET
```

```
    ORG 2000H   ; Programa principal
    MOV   AX, OFFSET NUM1
    MOV   CX, OFFSET NUM2
    CALL  MUL
    HLT
    END
```

Explicar detalladamente:

a) Todas las acciones que tienen lugar al ejecutarse la instrucción CALL MUL.
b) Todas las acciones que tienen lugar al ejecutarse las instrucciones PUSH DX y POP DX.
c) ¿Qué operación se realiza con la instrucción RET?

2) Escribir un programa que calcule el producto entre dos números sin signo almacenados en la memoria del microprocesador llamando a una subrutina MUL, pero en este caso pasando los parámetros por valor y por referencia a través de la pila.

```
        ORG   1000H; Memoria de datos
NUM1    DW    5H
NUM2    DW    3H
RES     DW    ?

        ORG   3000H ; Subrutina MUL
MUL:    PUSH  BX
        MOV   BX, SP
        PUSH  CX
        PUSH  AX
        PUSH  DX
        ADD   BX, 6
        MOV   CX, [BX]
        ADD   BX, 2
        MOV   AX, [BX]
SUMA:   ADD   DX, AX
        DEC   CX
        JNZ   SUMA
        SUB   BX, 4
        MOV   AX, [BX]
        MOV   BX, AX
        MOV   [BX], DX
        POP   DX
        POP   AX
        POP   CX
        POP   BX
        RET

        ORG   2000H ; Programa principal
        MOV   AX, NUM1
        PUSH  AX
        MOV   AX, NUM2
        PUSH  AX
        MOV   AX, OFFSET RES
        PUSH  AX
        MOV   DX, 0
        CALL  MUL
        POP   AX
        POP   AX
        POP   AX
        HLT
        END
```

Analizar el diagrama esquemático de la pila y verificar con el simulador:

DIRECCIÓN DE MEMORIA	CONTENIDO
7FF0H	DL
	DH
7FF2H	AL
	AH
7FF4H	CL
	CH
7FF6H	BL
	BH
7FF8H	IP RET. L
	IP RET. H
7FFAH	DIR. RES L
	DIR. RES H
7FFCH	NUM2 L
	NUM2 H
7FFEH	NUM1 L
	NUM1 H
8000H	——

Responder brevemente:

a) ¿Cuál es el modo de direccionamiento de la instrucción MOV AX, [BX]? ¿Qué se copia en el registro AX en este caso?
b) ¿Qué función cumple el registro temporal **ri** que aparece al ejecutarse una instrucción como la anterior?
c) ¿Qué se guarda en AX al ejecutarse MOV AX, OFFSET RES?
d) ¿Cómo se pasa la variable RES a la pila, por valor o por referencia? ¿Qué ventaja tiene esto?
e) ¿Cómo trabajan las instrucciones PUSH y POP?

Observaciones:

- Los contenidos de los registros AX, BX, CX y DX antes y después de ejecutarse la subrutina son iguales, dado que al comienzo se almacenan en la pila para poder utilizarlos sin perder la información que contenían antes del llamado. Al finalizar la subrutina, los contenidos de estos registros son restablecidos desde la pila.
- El programa anterior sólo puede aplicarse al producto de dos números mayores que cero.

Ejemplo de uso de la pila:

```
ORG 2000H
MOV AX, 1111H
MOV BX, 2222H
PUSH AX
ADD AX, BX
MOV CX, AX
POP AX
HLT
END
```

3) Suma de números de 32 bits. Escribir un programa que calcule la suma de dos números de 32 bits almacenados en la memoria del microprocesador:
   a) Sin hacer llamados a subrutinas, resolviendo el problema desde el programa principal;
   b) Llamando a una subrutina SUM32 para efectuar la operación, pasando los parámetros por valor desde el programa principal a través de registros;
   c) Llamando a una subrutina SUM32, pasando los parámetros por referencia desde el programa principal a través de registros.
   d) Llamando a una subrutina SUM32, pero en este caso pasando los parámetros por valor y por referencia a través de la pila.

4) Escribir una subrutina ROTARIZ que haga una rotación hacia la izquierda de los bits de un byte almacenado en la memoria del microprocesador. Dicho byte y el número de posiciones a rotar deben pasarse por valor desde el programa principal a la subrutina a través de registros.

5) Escribir una subrutina CONCAR que cuente el número de caracteres de una cadena de caracteres terminada en cero (00H) almacenada en la memoria del microprocesador. La cadena se pasa a la subrutina por referencia vía registro.

6) Escribir una subrutina SWAP que intercambie dos datos de 16 bits almacenados en memoria. Los parámetros deben ser pasados por referencia desde el programa principal a través de la pila.

7) Modificar la subrutina del ejercicio 5 para que cuente la cantidad de veces que se repite un dado caracter en una cadena. Además, la subrutina debe cambiar el caracter especificado por una "X". El caracter a buscar se debe pasar por valor mientras que la cadena a analizar por referencia a través de la pila.

8) Usando la subrutina ROTARIZ del ejercicio 4, escriba una subrutina ROTARDER que haga una rotación hacia la derecha de un byte almacenado en la memoria del microprocesador. Dicho byte y el número de posiciones a rotar deben pasarse por valor desde el programa principal a la subrutina a través de registros.

9) Escriba la subrutina ROTARDER del ejercicio anterior, pero sin usar la subrutina ROTARIZ. Compare que ventajas tiene cada una de las soluciones.

10) Escriba la subrutina RESTO que calcule el resto de la división entre 2 números positivos. Dichos números deben pasarse por valor desde el programa principal a la subrutina a través de registros.

11) Escriba la subrutina ES_VOCAL, que determina si un caracter es vocal o no. La rutina debe recibir el caracter por valor, y debe retornar, vía registro, el valor 0FFH si el caracter es una vocal, o 00H en caso contrario.

12) Usando la subrutina anterior escribir la subrutina VOCALES, que recibe una cadena por referencia, y devuelve, en un registro, la cantidad de vocales que tiene esa cadena

13) Analizar el funcionamiento de la siguiente subrutina y su programa principal:

```
     ORG 3000H
MUL: CMP AX, 0
     JZ  FIN
     ADD CX, AX
     DEC AX
     CALL MUL
FIN: RET
```

```
ORG 2000H
MOV CX, 0
MOV AX, 3
CALL MUL
HLT
END
```

   a) ¿Qué hace la subrutina?
   b) ¿Cuál será el valor final de CX?
   c) Dibujar las posiciones de memoria de la pila, anotando que valores va tomando
   d) ¿Cuál será la limitación para determinar el valor mas grande que se le puede pasar a la subrutina a través de AX?