

PRÁCTICA 1

MODOS DE DIRECCIONAMIENTO. ENSAMBLADOR

INTRODUCCIÓN

Esta práctica aborda el estudio de algunos modos de direccionamientos utilizados en la programación del microcontrolador LPC1768 de la familia Cortex M3 (ARM). Además, con ella se pretende una mejor comprensión del funcionamiento de los microprocesadores, a través de su programación en ensamblador.

La práctica está dividida en dos actividades. La Actividad 1.1 es no presencial y en ella se suministra un programa de ejemplo donde el alumno debe analizar los modos de direccionamiento de diferentes instrucciones y comprobar los valores que toman los registros y la memoria en cada una de ellas. También se pide realizar una pequeña modificación sobre otro programa.

La Actividad 1.2 es presencial en el laboratorio y en ella los alumnos llevarán a cabo el desarrollo de aplicaciones sencillas, basándose en el ejemplo facilitado en la Actividad 1.1.

En el Anexo I se explica cómo crear un programa en ensamblador para el LPC1768 con el entorno Keil μ Vision[®]4 y el Anexo II donde se resumen las principales directivas de ensamblador.

1.1. Actividad 1.1. Modos de direccionamiento I (No presencial)

En el código fuente del Programa 1.1 se utilizan instrucciones en ensamblador que permiten analizar y entender el funcionamiento de algunos de los modos de direccionamiento vistos en las clases de teoría.

Tenga en cuenta que al ensamblar el programa el área de código (RESET) se almacenará a partir de la dirección 0x0000_0000, a continuación el área de datos constantes (Constantes) y el área de datos variables (Variables) a partir de la dirección 0x1000_0000.

```

1  ;
2  ; Programa para el estudio de los modos
3  ; de direccionamiento en ensamblador
4
5  STACK_TOP EQU 0x10004000
6
7  AREA RESET, CODE ; La directiva ÁREA indica al
8                    ; ensamblador crear una nueva sección
9                    ; de código o de datos.
10                   ; Las secciones son independientes
11                   ; e indivisibles para el linker
12  DCD STACK_TOP    ; La directiva DCD asigna
13                   ; una o más palabras de memoria,
14                   ; alineados en grupos de cuatro bytes,
15                   ; y define el contenido inicial de la memoria.
16
17  DCD inicio
18
19  SPACE (0x400-8) ; Dejamos libre el primer Kbyte de memoria
20                 ; El programa comenzará en la dirección 0x400
21
22  ENTRY
23
24 inicio          ; Inicio del programa principal
25
26  MOV r0,#9
27  MOV r1,#0
28  ADD r1,#4
29  MOV r2,#0xA
30  MOV r5,#0x10000000
31  MOV r4,r5
32
33  STR r0,[r5]
34  LDR r3,[r5]
35  STR r0,[r5,#4]
36  LDR r3,[r5,#4]!
37
38  STR r1,[r5,r1]
39  LDR r3,[r5,r1]
40
41  STR r4,[r5],#2
42
43  LDMIA r4,{r0-r2}
44
45  LDR r0,=Lista
46  LDR r2,=Num
47  LDRB r1,[r2]
48  LDR r2,=Suma
49  MOV r3,#0x00
50  MOV r5,#0x4
51
52 lazo
53     LDR r4,[r0]
54     ADD r0,r5
55     ADD r3,r4
56     SUBS r1,#1
57     BNE lazo
58
59     STR r3,[r2]
60
61 deadloop
62     B deadloop

```

```
62
63      AREA Constantes, DATA, READONLY
64  Lista DCD 0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A
65  Num   DCB 10
66
67      AREA Variables, DATA
68  Espacio SPACE 40
69  Suma    SPACE 4
70  Numeros SPACE 80
71
72      END
73
```

PROGRAMA 1.1. INSTRUCCIONES CON DIFERENTES MODOS DE DIRECCIONAMIENTO.

1.1.1. Tareas a realizar en esta actividad

Cree un proyecto con el entorno Keil μ Vision[®]4 y ensamble el código fuente del Programa 1.1 siguiendo los pasos especificados en el Anexo I. A continuación realice los siguientes apartados:

- 1. Ejecute el Programa 1.1 paso a paso. Indique, para cada una de las líneas indicadas en la Tabla 1.1, los modos de direccionamiento utilizado, el registro o posición de memoria que se modifica y el valor que toma.

Tabla 1.1. Modos de direccionamiento y valores de los registros				
Línea	Modo de direccionamiento		Registro/Memoria modificado	Valor
	Destino	Fuente		
26				
32				
35				
38				
40				
46				

- 2. Adjunte una captura de pantalla de la ventana *Memory* en la que se muestren al menos los primeros 48 bytes de memoria, organizada en 4 bytes por fila, a partir de la dirección dada por la etiqueta Espacio, una vez ejecutado el programa.



3. Sustituya las dos instrucciones de las líneas 52 y 53 por una sola instrucción que haga la misma operación.



4. Complete el Programa 1.2 mostrado a continuación para que el bucle almacene los valores del 1 al 20 en tamaño *word* a partir de la dirección dada por la etiqueta Numeros. Los valores se deben almacenar en memoria como se muestra en la Tabla 1.2.

```
STACK_TOP    EQU 0x10004000
              AREA RESET, CODE
              DCD STACK_TOP
              DCD inicio
              SPACE (0x400-8)
              ENTRY

inicio        LDR r0,=Numeros
              MOV r1,#20

; Bucle a completar
lazo
              BNE lazo      ; Fin del bucle

deadloop      B deadloop

              AREA Variables, DATA
Numeros       SPACE 20*4

              END
```

PROGRAMA 1.2. CÓDIGO EN ENSAMBLADOR A MODIFICAR.

Tabla 1.2. Valores almacenados en memoria					
Dir	Numeros	Numeros+0x4	Numeros+0x8	Numeros+0xC
Valor	0x00000001	0x00000002	0x00000003	0x00000004

5. Adjunte una captura de la ventana *Memory* en la que se vean las posiciones que se han modificado una vez ejecutado el Programa 1.2.

6. Indique los problemas que se ha encontrado durante la realización de esta actividad.

1.2. Actividad 1.2. Modos de direccionamiento II (Presencial)

En la Actividad 1.2, que se llevará a cabo en el laboratorio de forma presencial, se realizarán modificaciones al programa en ensamblador estudiado en la Actividad 1.1. Los pasos que hay que seguir en esta actividad son los siguientes:

- Simular y depurar en el laboratorio los ejemplos propuestos.
 - Tomar evidencias de los resultados (valores numéricos, captura de pantallas, etc.).
 - **Mostrar en el laboratorio que ha realizado la actividad con aprovechamiento.**
1. A partir del Programa 1.1, determine la duración de las siguientes instrucciones:
 - `MOV r2, #0xA`
 - `LDR r3, [r5, r1]`
 - `LDR r2, =Suma`
 - `SUBS r1, #1`
 - `BNE lazo` (cuando salta a la etiqueta lazo)
 - `BNE lazo` (cuando ejecuta la siguiente instrucción –secuencialmente–, es decir, `STR r3, [r2]`).
 2. A partir del Programa 1.2, realice las siguientes actividades:
 - Modifique el programa para que desplace un '1' por los 32 bits del registro R0, desde el bit 0 al 31. Cuando el '1' alcanza la posición 31 el programa se debe quedar parado en un bucle infinito. **Nota:** La instrucción de desplazamiento a izquierdas es LSL.

- Modifique el programa para que la duración del '1' en cada posición sea igual al número de su puesto de trabajo en el laboratorio multiplicado por 2 μ s. Haga uso de un bucle y del conocimiento que ya tiene de la duración de ejecución de algunas instrucciones.



- Realice un desplazamiento circular, de forma que cuando el '1' llegue al bit 31 se repita el proceso comenzando de nuevo por el bit 0.

