

3. Codifica en binario y en hexadecimal la instrucción «**ldrh** r4, [r0, r1]» sabiendo que el formato de instrucción utilizado para dicha instrucción es el que aparece a continuación.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	H	S	1	Ro			Rb			Rd		

H y **S** Identifican la instrucción:

H	S	Instrucción
0	0	« strh rd, [rb, ro]»
0	1	« ldrsb rd, [rb, ro]»
1	0	« ldrh rd, [rb, ro]»
1	1	« ldrsh rd, [rb, ro]»

Ro Registro desplazamiento.

Rb Registro base.

Rd Registro fuente/destino.

El contenido en binario de cada uno de los campos, de izquierda a derecha, será:

- **0101**, que es fijo y nos indica de qué tipo de instrucción se trata.
- **10** en el campo **HS**, pues se trata de una instrucción «**ldrh**».
- **1**, que es fijo y forma parte del código de operación.
- **001** en el campo **Ro**, que codifica el registro desplazamiento, «**r1**».
- **000** en el campo **Rb**, que codifica el registro base, «**r0**».
- **100** en el campo **Rd**, que codifica el registro destino, «**r4**».

Juntando los campos se tiene:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	0	1	0	0	1	0	0	0	1	0	0

Que en hexadecimal es 0x5A44.

4. Describe el modo de direccionamiento de cada uno de los operandos de la instrucción «**ldrh** r4, [r0, r1]».

La instrucción propuesta tiene dos operandos, uno fuente y uno destino, especificado cada uno de ellos con su modo de direccionamiento.

El operando fuente es la dirección de memoria dada por la suma de los registros «**r0**» y «**r1**», por lo que se usa el modo de direccionamiento «indirecto con registro de desplazamiento» (también llamado «base más índice» o «indexado», en el caso de que «**r0**» contenga la dirección base y «**r1**», el índice).

El operando destino, «**r4**», utiliza el modo directo a registro. En este caso se indica que el contenido de la dirección de memoria se debe cargar en el registro «**r4**».



5. Dado el siguiente programa en ensamblador Thumb de ARM, describe su funcionamiento comentando cada una de sus líneas y anota a continuación el contenido final de los registros «r2» y «r3» y los contenidos de las direcciones de memoria desde la 0x2007 0004 (etiquetada como «vec») hasta la 0x2007 0018.

```
1      .data
2 tam:  .word 6           @ Inicializa una palabra con el valor 6
3 vec:  .word 1, 2, 3, 4, 5, 6 @ Inicializa un vector de palabras
4
5      .text
6 main: ldr r0, =vec      @ r0 apunta al comienzo del vector
7       ldr r1, =tam      @ r1 <- dirección tam
8       ldr r1, [r1]      @ r1 <- valor de tam (6)
9       bl pv            @ llama a la subrutina pv
10      wfi              @ termina la ejecución
11
12 pv:   sub r1, #1       @ r1 <- r1 - 1 (índice último elemento)
13       lsl r1, #2       @ r1 <- índice * 4
14       add r1, r0       @ r1 <- base + índice * 4 (dir. últ. elemento)
15 pv_b: cmp r0, r1       @ si r0 >= r1, salta a pv_f
16       bge pv_f         @ si r0 >= r1, salta a pv_f
17       ldr r2, [r0]     @ r2 <- contenido de memoria[r0]
18       ldr r3, [r1]     @ r3 <- contenido de memoria[r1]
19       str r2, [r1]     @ contenido de memoria[r1] <- r2
20       str r3, [r0]     @ contenido de memoria[r0] <- r3
21       add r0, #4       @ r0 <- r0 + 4
22       sub r1, #4       @ r1 <- r1 - 4
23       b pv_b          @ salta a pv_b
24 pv_f: mov pc, lr       @ vuelve de la subrutina
25
26      .end
```

El programa invierte un vector comenzando por su primer y último elemento, siguiendo con el segundo y penúltimo, y así sucesivamente hasta que se invierte por completo. En los registros y direcciones de memoria indicadas habrán los siguientes valores una vez se haya ejecutado el programa.

r2	<table border="1"><tr><td>3</td></tr></table>	3	0x2007 0004	<table border="1"><tr><td>6</td></tr></table>	6
3					
6					
r3	<table border="1"><tr><td>4</td></tr></table>	4	0x2007 0008	<table border="1"><tr><td>5</td></tr></table>	5
4					
5					
		0x2007 000C	<table border="1"><tr><td>4</td></tr></table>	4	
4					
		0x2007 0010	<table border="1"><tr><td>3</td></tr></table>	3	
3					
		0x2007 0014	<table border="1"><tr><td>2</td></tr></table>	2	
2					
		0x2007 0018	<table border="1"><tr><td>1</td></tr></table>	1	
1					