



# IS-603 Arquitectura de Computadoras

## I Parcial - Instrucciones de Carga de datos

IS-UNAH

I PAC 2024



# Contenido

## ① Instrucciones de carga

- Carga de palabras

- Carga de medias palabras

- Carga de bytes

## ② Instrucciones de almacenamiento

- Almacenamiento de palabras

- Almacenamiento de medias palabras

- Almacenamiento de bytes

## ③ Ejercicios





# Carga de datos constantes

- **mov** *rd*, #*Inm8* :  $rd \leftarrow Inm8$
- **ldr** *rd*, =*Inm32* :  $rd \leftarrow Inm32$



# Ejemplo de carga de constantes

```
1      .data
2 word1: .word 0x10203040
3 word2: .word 0x11213141
4 word3: .word 0x12223242
5
6      .text
7 main: ldr r0, =word1
8       ldr r1, =word2
9       ldr r2, =word3
10      wfi
```

Las instrucciones cargan en un registro, la dirección de memoria a la que apunta la etiqueta. Así, r0 tiene la dirección de memoria a la que apunta word1, que es la 0x2007 0000.



r0

r1

r2

r3

r4

r5

r6

r7

20070000	20070004	20070008					
----------	----------	----------	--	--	--	--	--

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x20070000	40	30	20	10	41	31	21	11	42	32	22	12				
0x20070010																
0x20070020																
0x20070030																
0x20070040																



# Carga de constantes

- **ldr** rd, [rb]
- **ldr** rd, [rb, #Offset5] <sup>a</sup>
- **ldr** rd, [rb, ro] <sup>b</sup>

---

<sup>a</sup>¿Qué valores se pueden codificar con #offset5 y ldr?

<sup>b</sup>¿Qué modos de direccionamiento se aplica en cada instrucción?



# Ejemplo de carga de palabras

```
1          .data
2 word1:   .word 0x10203040
3 word2:   .word 0x11213141
4 word3:   .word 0x12223242
5
6          .text
7 main:    ldr r0, =word1
8          mov r1, #8
9          ldr r2, [r0]
10         ldr r3, [r0, #4]
11         ldr r4, [r0, r1]
12         wfi
```





# Carga de medias palabras

Sin extensión de signo	Con extensión de signo
<b>ldrh</b> rd, [rb]	<b>ldrh</b> rd, [rb] <b>sxth</b> rd, rd
<b>ldrh</b> rd, [rb, #offset5 <sup>c</sup> ]	<b>ldrh</b> rd, [rb, #offset5] <b>sxth</b> rd, rd
<b>ldrh</b> rd, [rb, ro]	<b>ldrsh</b> rd, [rb, ro]

<sup>c</sup>¿Qué valores de offset se codifican en el caso de medias palabras?



```
1          .data
2  half1:  .hword -15
3  half2:  .hword 20
4  half3:  .hword 40
5
6          .text
7  main:   ldr r0, =half1
8          mov r1, #4
9          @ Sin extension de signo
10         ldrh r2, [r0]
11         ldrh r3, [r0,#2]
12         ldrh r4, [r0,r1]
13         @ Con extension de signo
14         ldrh r5, [r0]
15         sxth r5, r5
16         ldrsh r6, [r0,r1]
17  stop:   wfi
```



r0            r1            r2            r3            r4            r5            r6            r7

20070000	00000004	0000FFF1	00000014	00000028	FFFFFFFF	00000028	
----------	----------	----------	----------	----------	----------	----------	--

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x20070000	F1	FF	14	00	28	00										
0x20070010																
0x20070020																
0x20070030																
0x20070040																



# Carga de bytes

Sin extensión de signo	Con extensión de signo
<b>ldrb</b> rd, [rb]	<b>ldrb</b> rd, [rb] <b>sxtb</b> rd, rd
<b>ldrb</b> rd, [rb, #offset5 <sup>d</sup> ]	<b>ldrb</b> rd, [rb, #offset5] <b>sxtb</b> rd, rd
<b>ldrb</b> rd, [rb, ro]	<b>ldrsb</b> rd, [rb, ro]

<sup>d</sup>¿Qué valores de offset se codifican en el caso de bytes?



```
1
2          .data
3 byte1:   .byte -15
4 byte2:   .byte 20
5 byte3:   .byte 40
6
7          .text
8 main:    ldr r0, =byte1
9          mov r1, #2
10         @ Sin extensión de signo
11         ldrb r2, [r0]
12         ldrb r3, [r0,#1]
13         ldrb r4, [r0,r1]
14         @ Con extensión de signo
15         ldrb r5, [r0]
16         sxtb r5, r5
17         ldrsb r6, [r0,r1]
18 stop:    wfi
```



r0

r1

r2

r3

r4

r5

r6

r7

20070000	00000002	000000F1	00000014	00000028	FFFFFFFF	00000028	
----------	----------	----------	----------	----------	----------	----------	--

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x20070000	F1	14	28													
0x20070010																
0x20070020																
0x20070030																
0x20070040																





# Almacenamiento de palabras

- **str** rd, [rb]
- **str** rd, [rb, #Offset5]
- **str** rd, [rb, ro]





# Almacenamiento de medias palabras

- **strh** rd, [rb]
- **strh** rd, [rb, #Offset5]
- **strh** rd, [rb, ro]



# Almacenamiento de bytes

- **strb** rd, [rb]
- **strb** rd, [rb, #Offset5]
- **strb** rd, [rb, ro]



# Ejemplo de almacenamiento de palabras

```
1                                     .data
2 hword1: .space 2
3 hword2: .space 2
4 hword3: .space 2
5
6                                     .text
7 main:  ldr r0, =hword1
8        mov r1, #4
9        mov r2, #10
10       strh r2, [r0]
11       strh r2, [r0, #2]
12       strh r2, [r0, r1]
13 stop:  wfi
```



```
1      .data
2  X:      .word 1
3  E:      .word 1
4  LIM:    .word 100
5
6      .text
7  main:   ldr r0, =X
8          ldr r0, [r0]
9          ldr r1, =E
10         ldr r1, [r1]
11         ldr r2, =LIM
12         ldr r2, [r2]
13         cmp r0, r2
14         lsl r3, r1, #1
15         add r0, r0, r3
16         add r1, r1, #1
17         ldr r4, =X
18         str r0, [r4]
19         ldr r4, =E
20         str r1, [r4]
21  wfi
```

¿Cuál es el valor de r0 y r1 al finalizar el programa? Dibuje la memoria.





# Ejercicios

Desarrolle los siguientes ejercicios utilizando el simulador. Trate de bosquejarlos en papel y entender el ejercicio antes de ejecutarlo y comprobar sus resultados.

- ① 4.11
- ② 4.12
- ③ 4.15
- ④ 4.19
- ⑤ 4.20
- ⑥ 4.23

