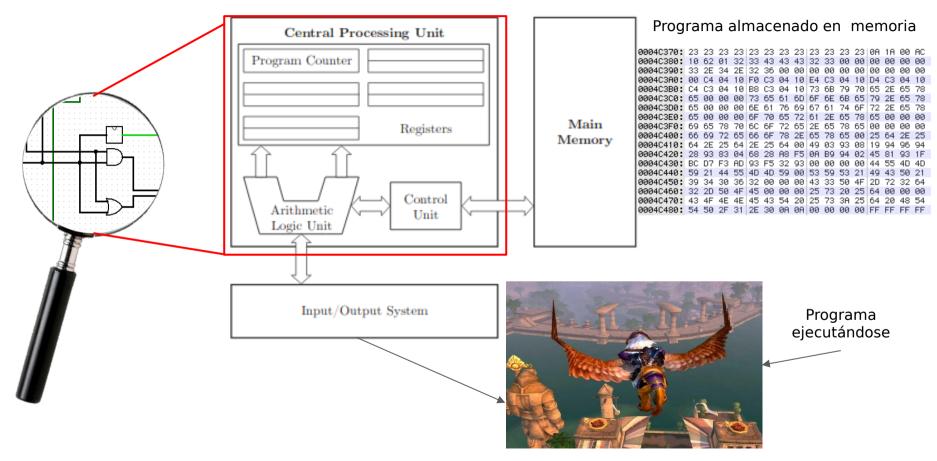
### Organización del Computador I

# Ciclo de Instrucción Codificación de instrucciones memoria

#### 1c2022

Universidad de Buenos Aires - FCEyN - Departamento de Computación

## ¿Qué buscamos comprender?

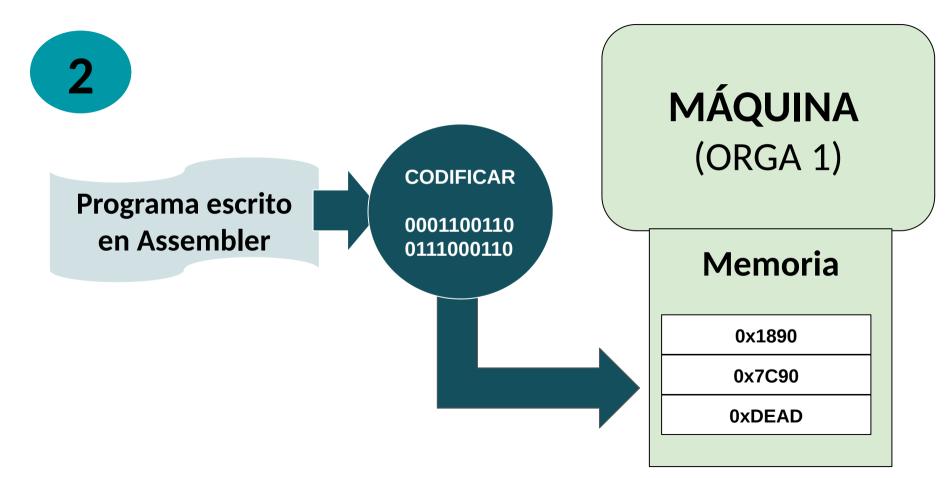


¿Qué vamos a ver hoy? **Usaremos la** cartilla o "manual" de ORGA 1 CODIFICAR Programa escrito 0001100110 en Assembler 0111000110

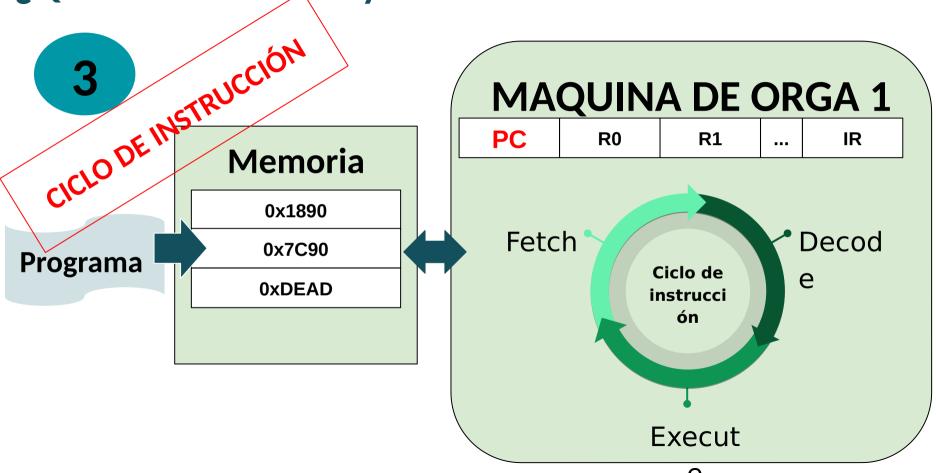
MOV R1, 0x02

ADD R0, 0x01

## ¿Qué vamos a ver hoy?



## ¿Qué vamos a ver hoy?





Codifique el siguiente programa que se carga en la posición de memoria 0x0000:

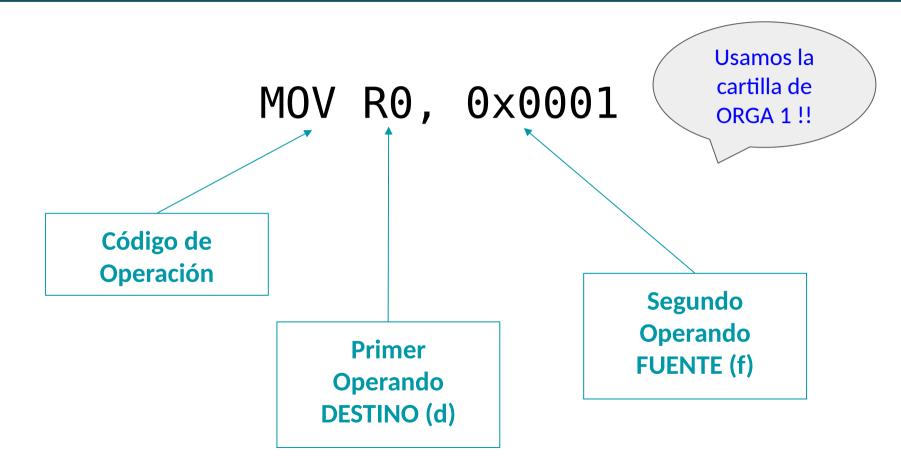
```
main: MOV R0, 0x0001
ADD R0, [inc]
JMP main
inc: DW 0x0002
```



Veamos la primer instrucción del programa:

MOV R0,  $0\times0001$ 







## MOV RO, $0\times0001$

#### Tipo de instrucción

```
Formato de instrucción

Tipo 1: Instrucciones de dos operandos

4 bits 6 bits 6 bits 16 bits 16 bits

cod. op. destino fuente constante destino (opcional) constante fuente (opcional)
```



## MOV R0, $0\times0001$

#### **Formato**

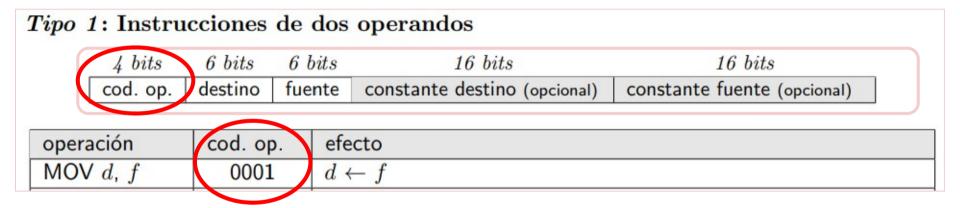
#### Tipo 1: Instrucciones de dos operandos

4 bits	6 bits	6 bits	$16 \ bits$	$16 \ bits$
cod. op.	destino	fuente	constante destino (opcional)	constante fuente (opcional)

operación	cod. op.	efecto
MOV $d$ , $f$	0001	$d \leftarrow f$

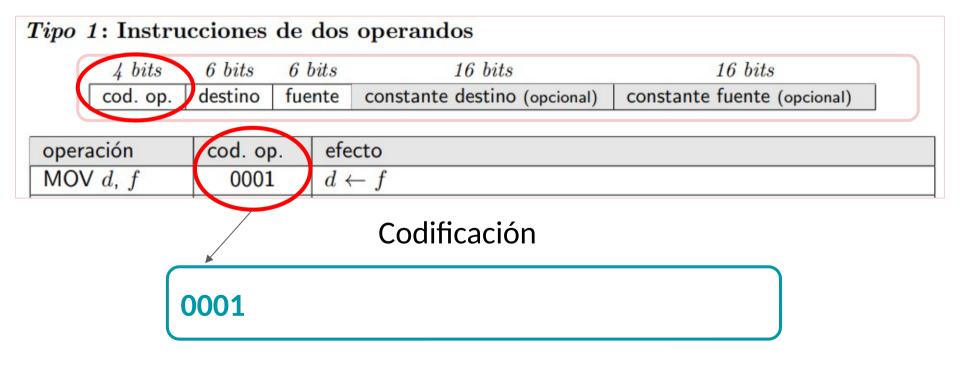


## MOV RO, $0\times0001$



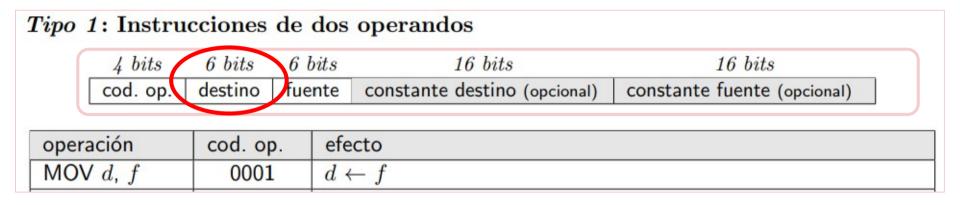


## MOV RO, $0\times0001$





## MOV RO, $0\times0001$



#### Codificación

0001



## MOV RO, $0\times0001$

#### Tipo 1: Instrucciones de dos operandos

4 bits 6 bits 6 bits 16 bits 16 bits 16 bits cod. op. destino fuente constante destino (opcional) constante fuente (opcional)

operación	cod. op.	efecto
MOV $d$ , $f$	0001	$d \leftarrow f$

Formato de operandos destino y fuente.

Modo	Codificación	Resultado
Inmediato	000000	c16
Directo	001000	[c16]
Indirecto	011000	[[c16]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Rrrr]
Indexado	111rrr	[Rrrr + c16]

c16 es una constante de 16 bits.

Rrrr es el registro indicado por los últimos tres bits del código de operando.

Las instrucciones que tienen como destino un operando de tipo *inmediato* son consideradas como inválidas por el procesador, excepto el CMP.

Indirecto

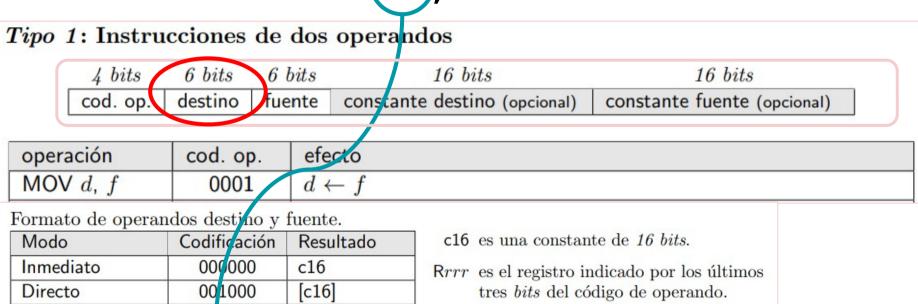
Registro

Indexado

Indirecto registro



## 0×0001



[Rrrr + c16]

[[c16]]

Rrrr

 $|\mathsf{K}rrr|$ 

011000

100rrr

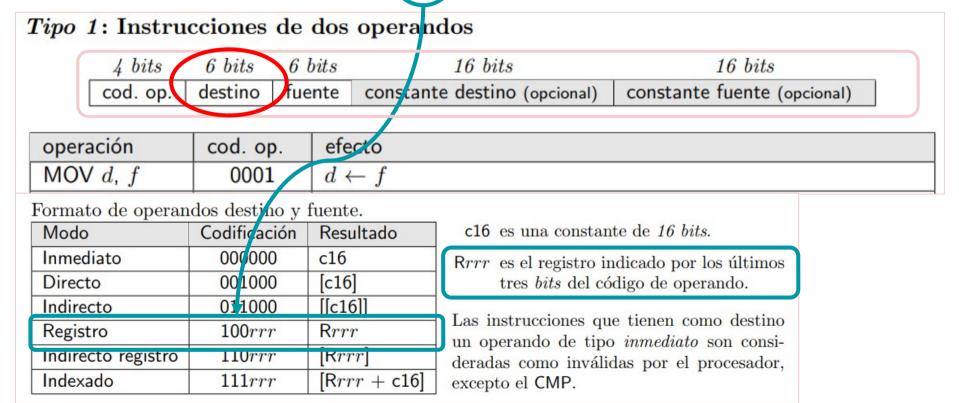
110rrr

111rrr

Las instrucciones que tienen como destino un operando de tipo inmediato son consideradas como inválidas por el procesador, excepto el CMP.



## MOV (RO), 0x0001







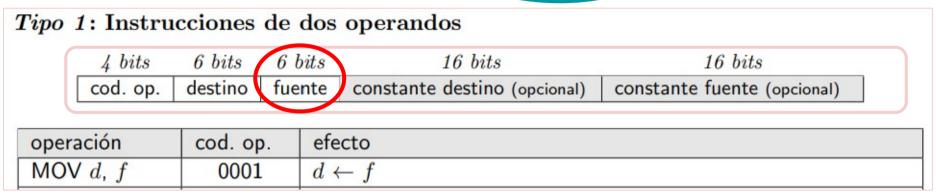


Codificación

0001 100000







#### Codificación

0001 100000



## MOV RO, 0x0001

#### Tipo 1: Instrucciones de dos operandos

4 bits 6 bits 6 bits 16 bits

cod. op. destino fuente constante destino (opcional) constante fuente (opcional)

operación	cod. op.	efecto	
MOV $d$ , $f$	0001	$d \leftarrow f$	

Formato de operandos destino y fuente.

Modo	Codificación	Resultado
Inmediato	000000	c16
Directo	001000	[C10]
Indirecto	011000	[[c16]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Rrrr]
Indexado	111rrr	[Rrrr + c16]

c16 es una constante de 16 bits.

\*\*Trr es el registro indicado por los últimos tres bits del código de operando.

Las instrucciones que tienen como destino un operando de tipo *inmediato* son consideradas como inválidas por el procesador, excepto el CMP.



## MOV RO, 0x0001

#### Tipo 1: Instrucciones de dos operandos

4 bits 6 bits 6 bits 16 bits

cod. op. destino (ruente constante destino (opcional) constante fuente (opcional)

operación	cod.	op.	efecto
MOV $d$ , $f$	000	)1	$d \leftarrow f$

Formato de operandos destino y fuente.

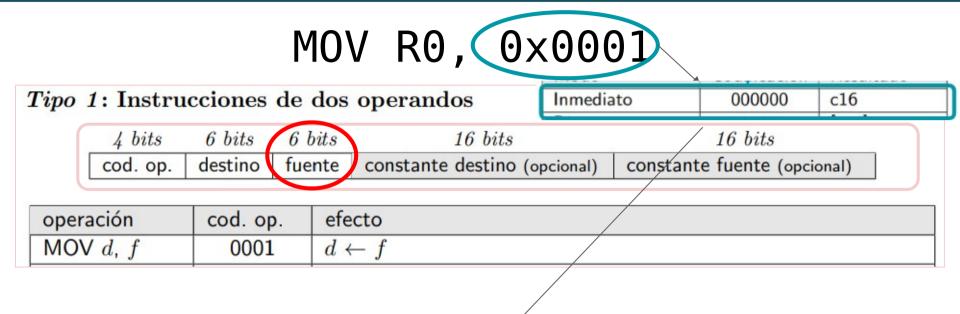
Modo	Codificación	Resultado
Inmediato	000000	c16
Directo	001000	[C10]
Indirecto	011000	[[c16]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Rrrr]
Indexado	111rrr	[Rrrr + c16]

c16 es una constante de 16 bits.

Rrrr es el registro indicado por los últimos tres bits del código de operando.

Las instrucciones que tienen como destino un operando de tipo *inmediato* son consideradas como inválidas por el procesador, excepto el CMP.



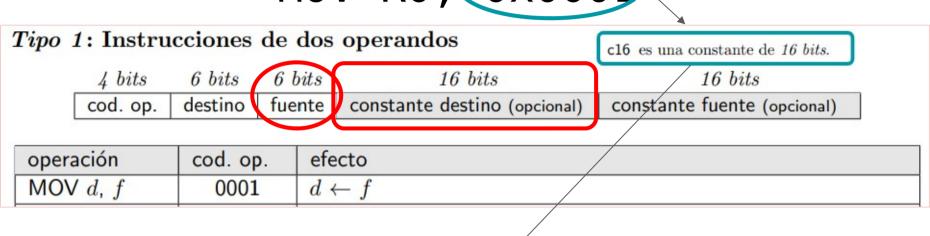


Codificación

0001 100000 000000







Codificación

0001 100000 000000 00000000000000000001



## MOV R0, $0\times0001$

Codificación



MOV RO,  $0\times0001$ 

Codificación

0001 1000 0000 0000 0000 0000 0000 0001

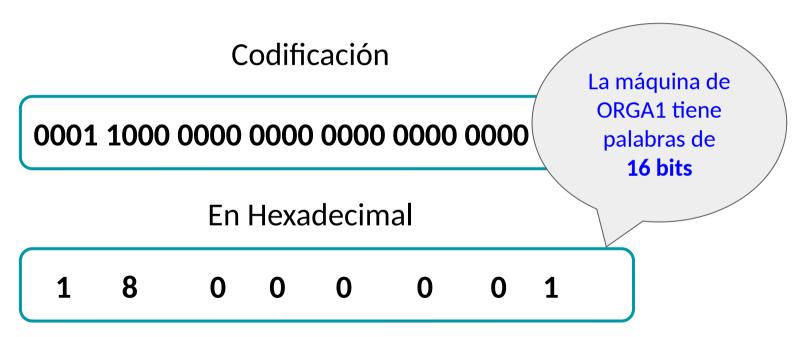
**En Hexadecimal** 

1 8 0 0 0 0 1

## ¿Cómo lo guarda en memoria?

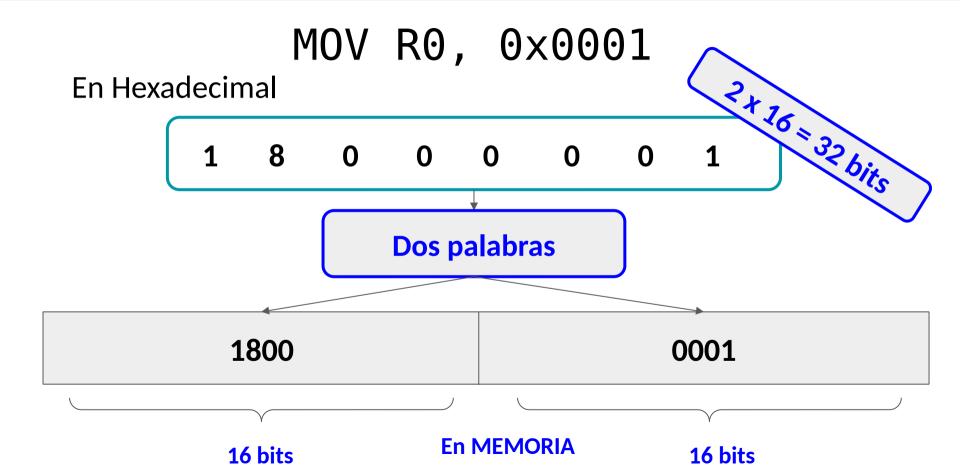


## MOV R0, $0\times0001$



## ¿Cómo lo guarda en memoria?







main: MOV R0, 0x0001

ADD R0, [inc]

JMP main

inc: DW 0x0002

Dirección de la memoria (16 bits)

Contenido de la memoria

#### **En MEMORIA**

0000	0001	0002	0003	0004	0005	0006
0007	0008	0009	000A	000B	000C	000D



main: MOV R0, 0x0001 🗹

ADD R0, [inc]

JMP main

inc: DW 0x0002

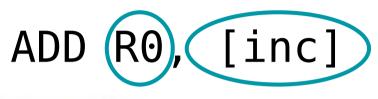
Recordemos que este programa se carga en la dirección 0x0000 (por el enunciado)

#### **En MEMORIA**

0000	0001	0002	0003	0004	0005	0006
1800	0001					
0007	8000	0009	000A	000B	000C	000D



Veamos la siguiente instrucción de un programa:



2 Operandos

Tipo 1: Instrucciones de dos operandos

4 bits	6 bits	6 bits	16 bits	16 bits
cod. op.	destino	fuente	constante destino (opcional)	constante fuente (opcional)

operación	cod. op.	efecto
MOV $d$ , $f$	0001	$d \leftarrow f$
ADD $d$ , $f$	0010	$d \leftarrow d + f$ (suma binaria)
CIII I f	0011	J. J f (

Código de operación

0010



Veamos la siguiente instrucción de un programa:



Formato de operandos destino y fuente.

Modo	Codificación	Resultado
Inmediato	000000	c16
Directo	001000	[c16]
Indirecto	011000	[[c16]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Krrr]
Indexado	111rrr	[Rrrr + c16]

c16 es una constante de 16 bits.

Rrrr es el registro indicado por los últimos tres bits del código de operando.

Las instrucciones que tienen como destino un operando de tipo *inmediato* son consideradas como inválidas por el procesador, excepto el CMP.

Primer Operando

0010 100000



Veamos la siguiente instrucción de un programa:

ADD R0, [inc]

Formato de operandos destino y fuente.

Modo	Codificación	Resultado
Inmediato	000000	c16
Directo	001000	[c16]
Indirecto	011000	[[C10]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Rrrr]
Indexado	111rrr	[Rrrr + c16]

c16 es una constante de 16 bits.

Rrrr es el registro indicado por los ultimos tres bits del código de operando.

Las instrucciones que tienen como destino un operando de tipo *inmediato* son consideradas como inválidas por el procesador, excepto el CMP.

Segundo Operando

0010 100000 001000



Veamos la siguiente instrucción de un programa:



Formato de operandos destino y fuente. Modo Codificación Resultado Inmediato 000000 c16 Directo 001000 [c16]Indirecto 011000 [[CTD]] Registro 100rrrRrrr[Rrrr]Indirecto registro 110rrr

111rrr

c16 es una constante de 16 bits.

tres

Las instru

un operan

deradas co

excepto el

Krrr es el registro indicado por los ultimos

main: MOV R0, 0x0001

ADD R0, [inc]

JMP main

inc: DW 0x0002

Segundo Operando

Indexado

0010 100000 001000 c16 inc?

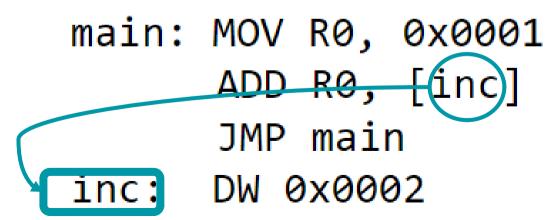
[Rrrr + c16]

"inc" es una etiqueta!!

## **Etiquetas**



- Las etiquetas son renombres de direcciones de memoria.
- Al momento de programar en Assembler, es imposible saber en qué dirección de memoria se va a guardar la información del programa.
- Las etiquetas nos ayudan a que podamos referirnos a posiciones de memorias relativas al inicio del programa y son resueltas por el ensamblador



## Cálculo de etiquetas



• ¿En qué posición de memoria va a quedar cada etiqueta? Calculemos cuántas palabras (16 bits en la máquina de Orga1) ocupa cada instrucción.

0x00000 main: MOV R0, 0x00001 2 palabras
0x00002 ADD R0, [inc] 2 palabras
JMP main
inc: DW 0x0002

## Cálculo de etiquetas



• ¿En qué posición de memoria va a quedar cada etiqueta? Calculemos cuántas palabras (16 bits en la máquina de Orga1) ocupa cada instrucción.

0x0000	main:	MOV R0, 0x0001	2 palabras
0x0002		ADD R0, [inc]	2 palabras
0x0004		JMP main	2 palabras
0x0006	inc:	DW 0x0002	1 palabra

## Cálculo de etiquetas



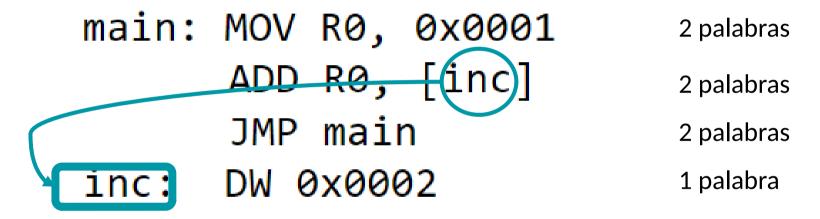
- La posición va a ser relativa el inicio del programa
- Si el programa empieza en la posición 0x0000, las direcciones de las etiquetas serán:

0x0000	main:	MOV R0, 0x0001	2 palabras
0x0002		ADD R0, [inc]	2 palabras
0x0004		JMP main	2 palabras
0x0006	inc:	DW 0x0002	1 palabra

## Cálculo de etiquetas



 Si cambiamos que el programa se cargue en la posición de memoria 0x0005, las etiquetas serías:



main 0x0005 inc 0x000B



Veamos la siguiente instrucción de un programa:

# ADD RO,



Formato de operandos destino y fuente.

Modo	Codificación	Resultado
Inmediato	000000	c16
Directo	001000	[c16]
Indirecto	011000	[[C10]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Rrrr]
Indexado	111rrr	[Rrrr + c16]

c16 es una constante de 16 bits.

Krrr es el registro indicado por los ultimos tres bits del código de operando.

Las instrucciones que tienen como destino un operando de tipo *inmediato* son consideradas como inválidas por el procesador, excepto el CMP.

"inc" 0x0006 en binario

Segundo Operando

0010 100000 001000 000000000000110



Veamos la siguiente instrucción de un programa:

ADD R0,  $[0 \times 0006]$ 

0010 1000 0000 1000 0000 0000 0000 0110

2808 0006

16 bits 16 bits



main: MOV R0, 0x0001 🗹

ADD R0, [inc] 🗹

JMP main

inc: DW 0x0002

0000	0001	0002	0003	0004	0005	0006
1800	0001	2808	0006			
0007	8000	0009	000A	000B	000C	000D

# **Actividad 1**

main: MOV R0, 0x0001 ☑
ADD R0, [inc] ☑
JMP main

inc: DW 0x0002

- 1. Codifiquemos JMP main
- 2. ¿Cuantas celdas de memoria de 16 bits ocupa toda la instrucción en memoria?
- 3. ¿Qué contenido tendría cada celda de memoria?



Veamos la siguiente instrucción de un programa:

#### JMP 0x0000

Tipo 2b: Instrucciones de un operando fuente.

4 bits	6 bits	6 bits	$16 \ bits$
cod. op.	000000	fuente	constante fuente (opcional)

operación	cod. op.	efecto
$JMP\ f$	1010	$PC \leftarrow f$

Etiqueta 'main' posición 0x0000



main: MOV R0, 0x0001 🗹

ADD R0, [inc] 🗹

JMP main

inc: DW 0x0002

0000	0001	0002	0003	0004	0005	0006
1800	0001	2808	0006	A000	0000	
0007	8000	0009	000A	000B	000C	000D



Veamos la siguiente instrucción de un programa:

DW 0x0002

DIRECTIVA	Directiva	Efecto	
ENSAMBLADOR	DW c16	Asigna en la posición correspondiente la constante c16	]

0002

16 bits



main: MOV R0, 0x0001 🗹

ADD R0, [inc]

JMP main



inc: DW 0x0002



0000	0001	0002	0003	0004	0005	0006
1800	0001	2808	0006	A000	0000	0002
0007	8000	0009	000A	000B	000C	000D

#### Cálculo de desplazamiento de saltos condicionales

#### Cálculo de saltos condicionales



```
main: MOV R1,[[et1]]
      ADD [R1],0x6000
      JMP main
 et1: DW 0x0007
      DW 0x0004
loop: CMP R1,0\times0001
      JL fin
      JGE main
 fin: RET
```



Los saltos condicionales son relativos

Esto significa que las etiquetas de los saltos son reemplazadas por el desplazamiento necesario para "llegar" desde la dirección en la que estamos parados, hasta la de la etiqueta de destino.

**Desplazamiento = Dirección Etiqueta - Dirección instrucción (post fetch)** 



#### Los saltos condicionales son relativos

Esto significa que las etiquetas de los saltos son reemplazadas por el desplazamiento necesario para "llegar" desde la dirección en la que estamos parados, hasta la de la etiqueta de destino.

Desplazamiento = Dirección Etiqueta - Dirección instrucción (post fetch)

- = Dirección instrucción + incremento
- = Contenido del PC



```
main: MOV R1,[[et1]]
0 \times 0000
0 \times 0002
                   ADD [R1],0x6000
0 \times 0004
                   JMP
                         main
0 \times 0006
           et1: DW
                        0 \times 0007
0 \times 0007
                   DW 0x0004
8000x0
           loop: CMP R1,0\times0001
                   JL fin
0 \times 000A
0 \times 000B
                   JGE main
0 \times 000C
            fin: RET
```

**Dirección Etiqueta fin = 0x000C** 

Dirección instrucción = 0x000B

**Desplazamiento: 1 (decimal)** 

El desplazamiento es **positivo** porque avanza en el programa



$0 \times 0000$	main:	MOV R1,[[et1]]
0×0002		ADD [R1],0x6000
0×0004		JMP main
0×0006	et1:	DW 0×0007
0×0007		DW 0×0004
8000x0	loop:	CMP R1,0x0001
0×000A		JL fin
0x000B		JGE main
0x000C	fin:	RET

En cambio, es **negativo** cuando retrocede en el flujo del programa.

Es decir, ¡cuando se salta hacia atrás!

Importante recordar que el desplazamiento se representa en complemento a dos en la máquina de Orga1.

# Actividad 2

```
main: MOV R1, [[et1]]
      ADD [R1],0x6000
      JMP main
 et1: DW 0x0007
      DW 0x0004
loop: CMP R1,0\times0001
      JL fin
      JGE main
 fin: RET
```

1. ¿Cuál es el desplazamiento de : **JGE main** ?

#### Solución cálculo de desplazamiento JGE main



```
main: MOV R1,[[et1]]
0 \times 0000
0 \times 0002
                   ADD [R1],0x6000
0 \times 0004
                   JMP
                        main
0 \times 0006
          et1: DW
                       0 \times 0007
0 \times 0007
                   DW 0x0004
8000x0
          loop: CMP R1,0x0001
                   JL fin
0 \times 000A
0 \times 000B
                   JGE main
0 \times 000C
            fin: RET
```

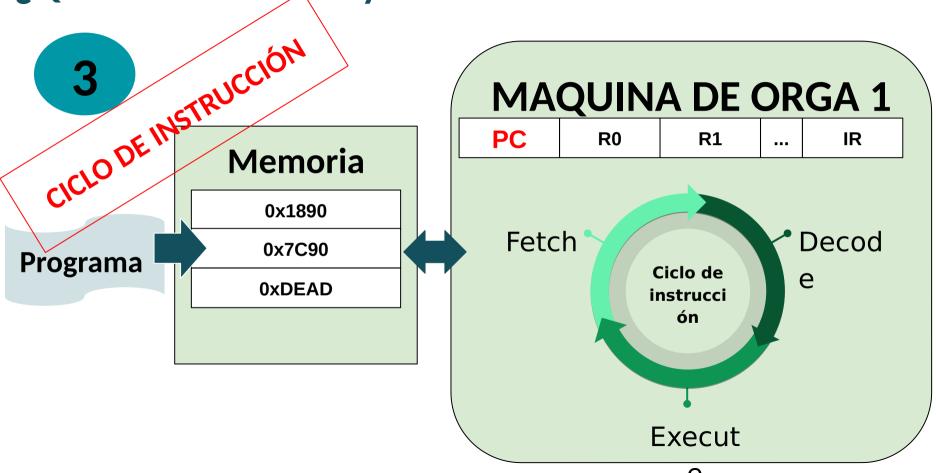
**Dirección Etiqueta main = 0x0000** 

Dirección instrucción = 0x000C

Desplazamiento: -12 (decimal) = 0xF4

# Intervalo 5 minutos

# ¿Qué vamos a ver hoy?



#### Máquina de Orga 1



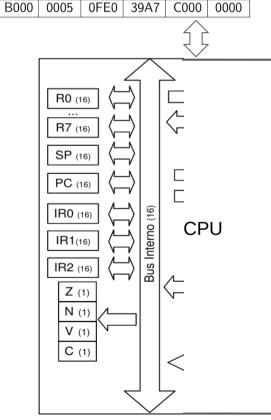
0006

0007

#### Práctica 3 - Arquitectura ORGA1

Descripción General

- ▶ Palabras de 16 bits.
- Direccionamiento a palabra.
- $\triangleright$  Espacio direccionable de 65536 palabras.
- ▷ Espacio de direcciones dedicado a entrada/salida en las direcciones 0xFFF0 0xFFFF.
- ▷ Ocho registros de propósito general de 16 bits: R0..R7.
- ▷ Program counter (PC) de 16 bits.
- ▷ Stack pointer (SP) de 16 bits inicializado en la dirección 0xFFEF.
- ► Los valores de los flags se calculan interpretando los operandos en complemento a 2.
   Flags: Z (zero), N (negative), C (carry), V (overflow).
- ▷ Todas las instrucciones alteran los flags, excepto MOV, CALL, RET, JMP y Jxx.
- $\triangleright\,$  De las que alteran los  $\mathit{flags},$ todas dejan C y V en cero, excepto ADD, ADDC, SUB, CMP y NEG.



0000

19C8

0001

0004

0002

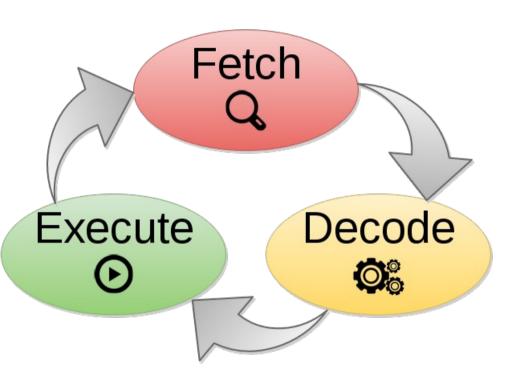
0003

0004

0005

#### Seguimiento - Simulando el ciclo de instrucción





#### → Fetch

◆ La UC (Unidad de Control) obtiene una instrucción de la posición de Memoria a la que apunta el PC y lo incrementa .

#### → Decode

◆ La UC decodifica la instrucción (si es necesario: busca más palabras de la instrucción usando el PC e incrementándolo cada vez).

#### → Execute

◆ La UC ejecuta la instrucción.

#### Seguimiento - Simulando el ciclo de instrucción



#### ¿Qué herramientas de análisis tenemos?

• Lapiz y papel: una planilla de seguimiento





				Planilla de Seguimiento    SP   R0   R1   R2   R3   R4   R5   R6   R7   2																
V	alores i	niciales:	PC	SP		RC	F	R1	R2	R3	R4	R5	R6	R7	-	Z	С	V	N	4
			+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D		+E	+]	
М	noria:		- 10	11	12	13	1-9	13	10	.,	10	13	· A	15	10	1.0			',	
Mei	noria.																			
		T			SP+1] IR Instrucción - 1 <sup>er</sup> palabra PC 2 <sup>da</sup> PC 3 <sup>era</sup> PC										In	stru	cció	n		
	PC		SP	[SP+1]	Pt] IR (en bits) PC palabra PC palabra											de	codi	ica	la	
1																				
	E	jecució	in														Flags <sup>1</sup>	Z	С	V N
2																				
	E	jecució	in														Flags <sup>1</sup>	Z	С	V N
3																1				
	E	jecució	n														Flags <sup>1</sup>	Z	С	V N
4																				
	E	jecució	n														Flags <sup>1</sup>	Z	С	V N
5																				
	E,	jecució	n														Flags <sup>1</sup>	Z	С	V N
6																				
	E	jecució	n	,								,			,		Flags <sup>1</sup>	Z	С	V N
7																				
	E	jecució	n							•							Flags <sup>1</sup>	Z	С	V N
8																				
	Е	jecució	n														Flags <sup>1</sup>	Z	С	V N

<sup>&</sup>lt;sup>1</sup>Sólo deben completarse luego de la ejecución de una instrucción que los altere.



		Planilla de Seguimiento    PC   SP   R0   R1   R2   R3   R4   R5   R6   R7     2																		_	
	V	alores ini	ciales:	PC	SP		RC	) ]	R1	R2	R3	R4	R5	R6	R7		Z	C	V	N	
	٧,	alores IIII	Ciales.																		
				+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D		+E	+F	7
J.	Men	noria:																			
*																					
Estado de los		PC	Т	SP	500.43	IR	Ins	trucció	n - 1	r palab	ra	PC	$2^{da}$	PC	3 <sup>era</sup>	PC		Ir	struc	cción	
		PC		SP	[SP+1]	IK		(	n bits	)		PC	palabra	PC	palabra	PC		de	codif	icada	1
registros	1																				
•		Ej	ecució	in	'		'								•		F	lags <sup>1</sup>	Z	C V	N
	2																				
,		Ej	ecució	ón										F	lags <sup>1</sup>	Z	C V	N			
	3																				
,				in													F	lags <sup>1</sup>	Z	C V	N
	4																				
,		Ej	ecució	in											1		F	lags <sup>1</sup>	Z	C V	N
	5																				
,		Ej	ecució	in	·						'						F	lags <sup>1</sup>	Z	C V	N
	6																				
,		Ej	ecució	in	'												F	lags <sup>1</sup>	Z	C V	N
	7																				
•		Ej	ecució	in	1		•				,				•	•	F	lags <sup>1</sup>	Z	C V	N
	8																				
•		Ej	ecució	in										•	•	•	F	lags <sup>1</sup>	Z	C V	N

<sup>&</sup>lt;sup>1</sup>Sólo deben completarse luego de la ejecución de una instrucción que los altere.



		Planilla de Seguimiento  PC   SP   R0   R1   R2   R3   R4   R5   R6   R7   Z																			
	٧	'alores inicial		C SP		RO	R1		R2	R3	R4	R5	R6	R7	-	Z	С	V	N		
			+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E		+F		
	Mer	noria:																			
<b>↓</b>		PC	SP	[SP+1]	IR	Inst		- 1 <sup>er</sup>	palabra	a P	C p	2 <sup>da</sup> alabra	PC	3 <sup>era</sup> palabra	PC		Instrucción decodificada				
Memoria	1																				
		Ejecu	ción													F	Lags <sup>1</sup>	Z C	V N		
	2																				
		Ejecu	ción													F	Lags <sup>1</sup>	Z C	V N		
	3																				
		Ejecu	ción												F.	Lags <sup>1</sup>	Z C	V N			
	4																				
		Ejecu	ción													F	Lags <sup>1</sup>	z c	V N		
	5																				
		Ejecu	ción													F	Lags <sup>1</sup>	z c	V N		
	6																				
		Ejecu	ción									'			'	F	Lags <sup>1</sup>	z c	V N		
	7																•	İ			
		Ejecu	ción									•				F	Lags <sup>1</sup>	Z C	V N		
	8																•				
		Ejecu	ción							•	'					F	Lags <sup>1</sup>	Z C	V N		

<sup>&</sup>lt;sup>1</sup>Sólo deben completarse luego de la ejecución de una instrucción que los altere.



	Planilla de Seguimiento																				
	\	/alores ini	ciales:	PC	SP		RC		R1	R2	R3	R4	R5	R6	R7	-	Z	С	V	N	
				+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E		+F	
	Mo	moria:															- 2				-
	Wic	moria.																			-
		PC	S	SP	[SP+1]	IR	Ins	trucció	n - 1 <sup>er</sup>	palabi	ra	PC	2 <sup>da</sup> palabra	PC	3 <sup>era</sup> palabra	PC				ción .cada	
	1																				
		Ej€	ecució	n	'												F	lags <sup>1</sup>	Z C	V	N
J	2																				
<b>V</b>		Eje	ecució	n													F	lags <sup>1</sup>	Z	v	N
Ciclo de	3																			•	
Instrucción		Eje	ecució	n													F	lags <sup>1</sup>	Z	v	N
	4																				
		Eje	cució	n													F	$lags^1$	Z	v	N
	5																				
		Eje	cució	n													F	lags <sup>1</sup>	Z C	v	N
	6																				
		Eje	cució	n	'						'			'		'	F	lags <sup>1</sup>	Z C	V	N
	7																			•	
		Eje	ecució	n	'									•	•		F	$lags^1$	Z	V	N
	8																				
		Ej€	cució	n										•	•		F	lags <sup>1</sup>	z	v	N

<sup>&</sup>lt;sup>1</sup>Sólo deben completarse luego de la ejecución de una instrucción que los altere.

#### **Ejercicio**



main: MOV R1,[[et1]]

ADD [R1],0x6000

0001

0006

8000

0000

JMP main

et1: DW 0x0007

DW 0x0004

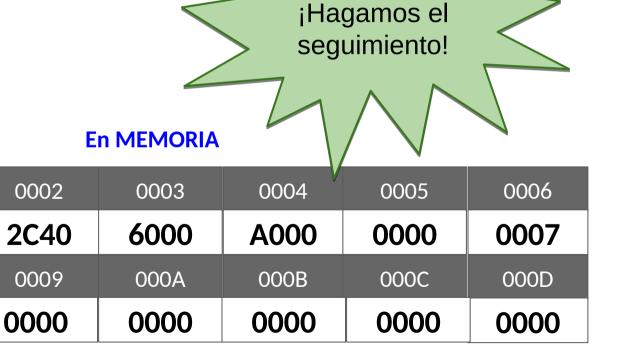
fin: RET

0000

1858

0007

0004





R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1											
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										



R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000										
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0	000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0	010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000								
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	1858							
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags



R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000						
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags



R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001					
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags



R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra PC 2da palabra PC 3ra palabra PC		Instrucción decodificada			
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002		
	Ejecución									Flags
2										
	Ejecución									Flags
3										
	Ejecución									Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags



R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
			. ,				palabra		palabra		decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0x0006]]=[0x0007]=4									Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
C	0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
C	010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
		٠.	[0. , _]				palabra		palabra		decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0×0006]]
	Ejecución	R1=[[0	×0006]]=[	0×0007]	=4						Flags
2	0002										
	Ejecución										Flags
3											
	Ejecución									·	Flags



Valores PC SP iniciales 0000 FFEF

R0 R1 R2 R3 R4 R5 R6 R7 0000 0000 0000 0000 0000 0000 0000 0000

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada		
							palabra		parabra		decodificada		
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]		
	Ejecución	R1=[[0	)×0006]]=[	0×0007]	=4						Flags		
2	0002			2C40									
	Ejecución										Flags		
3													
	Ejecución									Flags			



Valores PC SP iniciales 0000 FFEF

							R7
0000	0000	0000	0000	0000	0000	0000	0000

Z		: V	N	
0	0	0	0	

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
							Palabia		ратарта		
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	)×0006]]=	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000						
	Ejecución										Flags
3											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
			[ , -]				palabra		palabra		decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003					
	Ejecución										Flags
3											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
		٥,	[51   1]		mistracción - Ici palabra		palabra	' `	palabra	' `	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0×0006]]
	Ejecución	R1=[[0	×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			
	Ejecución										Flags
3											
	Ejecución									·	Flags



Valores	PC	SP
iniciales	0000	FFEF

					R5		
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra		2da	PC	3ra	PC	Instrucción
	1	31	[31 +1]	111	mstrucción - Ter palabra	PC	palabra	1	palabra		decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0x0006]]=[0x0007]=4							Flags		
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución										Flags
3											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

					R5		
0000	0000	0000	0000	0000	0000	0000	0000

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra		2da	PC	3ra	PC	Instrucción
		31	[31-1]		mstrucción - Ter palabra	PC	palabra	1	palabra		decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×0004	4]=0×A00	$0+0\times60$	00=0×0000						Flags
3											
	Ejecución				Flags						



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
	FC	31	[31-1]	IIX	ilistruccion - Ter palabra	FC	palabra	PC	palabra	1	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	R1=[[0x0006]]=[0x0007]=4								
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×0004	4]=0×A00	0+0×60	00=0×0000						Flags 1 1 0 0
3											
	Ejecución				Flags						



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción		
		31	[3[+1]	IIX	mstrucción - Ter palabra	FC	palabra	1	palabra	rc	decodificada		
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]		
	Ejecución	R1=[[0x0006]]=[0x0007]=4									Flags		
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000		
	Ejecución	[0×0004	4]=0×A00	0+0x6000=0x0000 Flags 1 1							Flags 1 1 0 0		
3	0004												
	Ejecución										Flags		



Valores PC SP iniciales 0000 FFEF

				R4			
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
	10	31	[31 +1]	IIX	mstrucción - Ter palabra	1.0	palabra	1	palabra	10	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	)×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×0004								Flags 1 1 0 0	
3	0004			0000							
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

					R5		
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
	1	31	[31 +1]	IIX	mstruccion - Ter palabra	1.0	palabra	1	palabra	10	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	×0006]]=	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0x0004]=0xA000+0x6000=0x0000						Flags 1 1 0 0			
3	0004			0000	0000 0000 0000 0000						
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
		31	[31-1]	IIX	mstrucción - Ter palabra	FC	palabra	FC	palabra	1	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0x0004	4]=0×A00	0+0×60	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000						
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

				R4			
0000	0000	0000	0000	0000	0000	0000	0000

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
	1	31	[31 +1]	IIX	mstrucción - Ter palabra	1.0	palabra	1	palabra	10	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	)×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0x0004]=0xA000+0x6000=0x0000					Flags 1 1 0 0				
3	0004			0000	0000 0000 0000 0000	0005					
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
		31	[31-1]	IIX.	Instrucción - Ter palabra	FC	palabra	1	palabra	1	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×000	4]=0×A00	0+0×60	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000	0005					Instrucción Inválida
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0000

_		V	IN
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
C	0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
C	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
	FC	31	[31-1]	IIX	mstrucción - Ter palabra	FC	palabra	FC	palabra	rc	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	×0006]]=[	0×0007]	=4						Flags
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×0004	4]=0×A00	0+0×60	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000	0005					Instrucción Inválida
	Ejecución	FIN DE	E LA EJEC	CUCIÓN							Flags



							. 0000 F	lanil	la de S	Segui	mient	to								
V	alores i	niciales:	PC		<b>–</b> (	RC	R	1	R2	R3	R4	R5	R6	R7		Z	C	V	N	
٠	alores ii	inclates.	000		\	000				0000	0000	0000	0000	0000		0	0	0	0	_
	-		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D		+E	+F	-
Mer	noria:	0000	1858	0006	2C40	6000	-A000	0000	0007	0004	0000	0000	0000	0000	0000	000	0 00	000	0000	1
		0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0 0	000	0000	١
	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$																			
1	0000	) F	FEF	0000	1858	000	1 1000	0101	1 1000	00	01	0006	0002				MOV	R1,	[[0x00	06]]
	E	jecuci	ón	R1 = [[0>	(0006]] =	[0x00	07] = 4										Flags <sup>1</sup>	Z	C V	N
2	0002	2			2C40	00	10 110	0100	0000	00	03	6000	0004				ADD	[R1]	,0x600	00
	E	jecuci	ón	[0x0004	] = 0xA0	00 + 0>	6000 =	0x0000	0								Flags <sup>1</sup>	z 1	0 v	N O
3	0004	1			0000	000	0000	0000	0000	00	05						Instru	ıcció	n inváli	ida
	E	jecuci	ón	FIN DE L	A EJECUC	IÓN											Flags <sup>1</sup>	Z	C V	N

#### Seguimiento - Simulando el ciclo de instrucción



#### ¿Qué herramientas de análisis tenemos?

• Lapiz y papel: una planilla de seguimiento

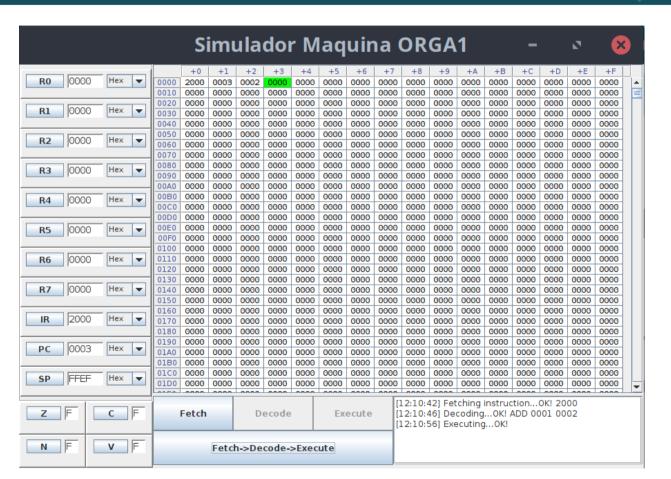


• El simulador



#### El simulador





# **Ejercicio**



Realizar el seguimiento de la ejecución del siguiente programa cargado en memoria a partir de la dirección 0x0000.

0000	0001	0002	0003	0004	0005	0006	0007
19C8	0004	B000	0005	0FE0	39A7	C000	0000



#### Ya se ejecutó la primer instrucción de 2 palabras

Valor	es	PC	S	SP		R0	R1	.	R	2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les	0000	F	FEF		000	0 00	00	00	000	0000	0000	0000	0000	0FE	<b>O</b>	0	0	0	0
	-	) +	1	+2		+3	+4	+.	5	+6	+7	+8	+9	+A	+B	+C	+	·D	+E	+F
0000	19C	8 000	)4	B000	0	005	0FE0	39/	47	C000	0000	0000	0000	0000	0000	0000	00	000	0000	0000
0010	000	000	00	0000	0	000	0000	000	00	0000	0000	0000	0000	0000	0000	0000	00	000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	×0004]=0×	k0FE0							Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags



Valor	es P	C	SP	F	30	R1	R	2	R3	R4	R5	R6	R7		Z	C	V	N	
inicia	les 0	000	FFEF	C	0000	00	00 00	000	0000	0000	0000	0000	0FE	0	0	0	0	0	
																			_
	+0	+1	+2	+3	3   -	+4	+5	+6	+7	+8	+9	+A	+B	+C	+[	)	+E	+F	
0000	19C8	0004	B000	000	5 01	FE0	39A7	C000	0000	0000	0000	0000	0000	0000	000	00	0000	0000	1
0010	0000	0000	0000	000	0 00	000	0000	0000	0000	0000	0000	0000	0000	0000	000	00	0000	0000	1

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	×0004]=0×	OFE0							Flags
2	0002		0x0004]=0x0FE0								
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución	ón								Flags	



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	0000	0FE0

Z	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0	0000	19C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0	010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001 0004 0002				MOV R7, [0x0004]	
	Ejecución	R7=[0:	×0004]=0×	OFE0							Flags
2	0002			B000							
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	0000	0FE0

Ζ	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	×0004]=0×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000						
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	0000	0FE0

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	R7=[0×0004]=0×0FE0								Flags
2	0002			B000	1011 0000 0000 0000	0003					
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0FE0

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0x0004]=0x0FE0									Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags



Valores	PC	SP		
iniciales	0000	FFEF		

R0							
0000	0000	0000	0000	0000	0000	0000	0FE0

Ζ	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	7=[0x0004]=0x0FE0							Flags	
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0FE0

2	Ζ	C	V	N	
	0	0	0	0	

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

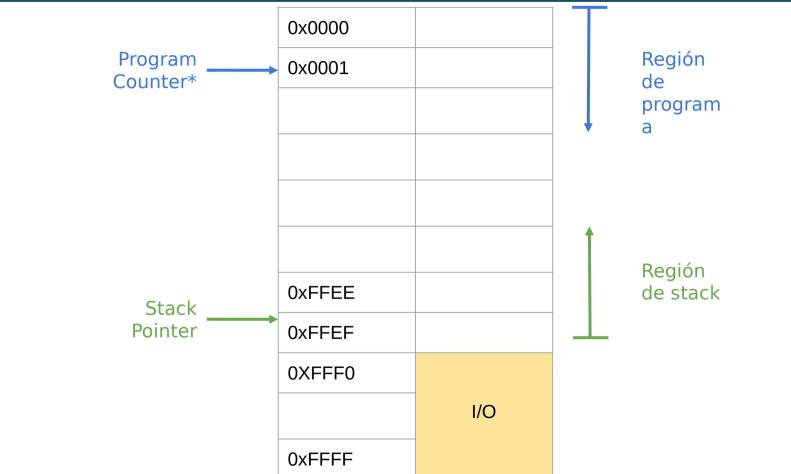
	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	×0004]=0×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags





Repasemos la organización de la memoria...







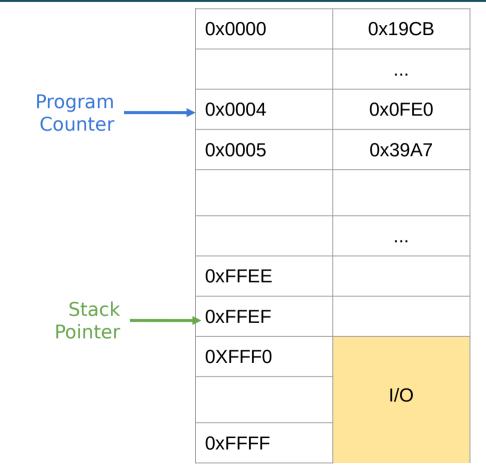
Volviendo a nuestro ejercicio:

#### **CALL 0x0005**

operación	cod. op.	efecto
$JMP\ f$	1010	$PC \leftarrow f$
CALL f	1011	$[SP] \leftarrow PC,  SP \leftarrow SP - 1,  PC \leftarrow f$

El formato del operando fuente responde a la tabla de formatos de operando mostrada más arriba.





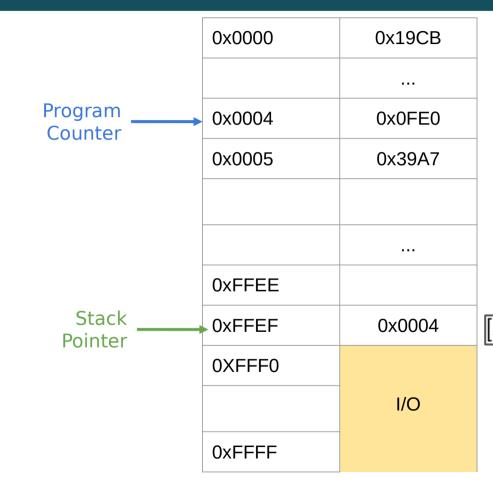
#### En nuestro caso:

$CALL\ f$	1011	$[SP] \leftarrow PC,  SP \leftarrow SP - 1,  PC \leftarrow f$

• 
$$f = 0 \times 0005$$

• 
$$PC = 0x0004$$





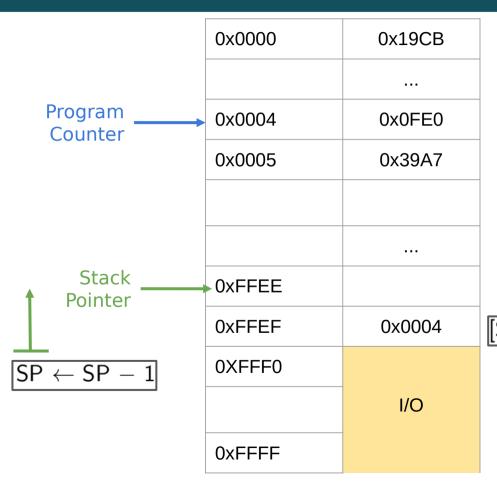
#### En nuestro caso:

$CALL\ f$	1011	$[SP] \leftarrow PC, SP \leftarrow SP - 1,$	$PC \leftarrow f$

• 
$$f = 0 \times 0005$$

• 
$$PC = 0x0004$$



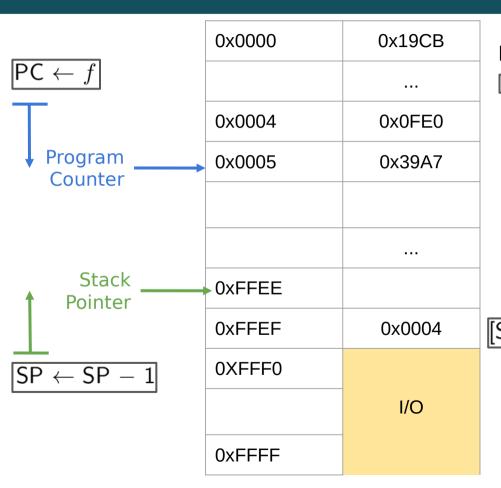


#### En nuestro caso:

CALL <i>f</i> 101	1 [SP] $\leftarrow$ PC, S	$SP \leftarrow SP - 1,  PC \leftarrow f$
-------------------	---------------------------	--

- f = 0x0005
- SP 1 = 0xFFEE
- PC = 0x0004





#### En nuestro caso:

$CALL\ f$	1011	$ $ [SP] $\leftarrow$ PC, SP $\leftarrow$ SP $-$ 1, PC $\leftarrow$ $f$

- $f = 0 \times 0005$
- SP 1 = 0xFFEE
- PC = 0x0004



#### Volviendo a la planilla:

Valores				1						
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0FE

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción		
		٥.	[5, 12]		motracción Ter parabra		palabra		palabra		decodificada		
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]		
	Ejecución	R7=[0:	×0004]=0×	Flags									
2	0002			B000	1011 0000 0000 0000	CALL 0×0005							
	Ejecución	[0×FFE	F]=0×000		Flags								
3													
	Ejecución										Flags		
4													
	Ejecución										Flags		
5													
	Ejecución				Flags								



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	0000	0FE0

Z	С	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]	
	Ejecución	R7=[0:	R7=[0x0004]=0x0FE0									
2	0002			B000   1011 0000 0000 0000   0003   0005   0004   C						CALL 0x0005		
	Ejecución	[0×FFE	F]=0×000	Flags								
3	0005											
	Ejecución										Flags	
4												
	Ejecución										Flags	
5												
	Ejecución			Flags								



### La 3era instrucción es un SUB - Esto ya lo sabemos hacer

Valor	es P	C	SP		R0	R1		R2	R3	R4	R5	R6	R7		Z	C	V	N
inicia	les 0	000	FFEF		0000	00	00 (	0000	0000	0000	0000	F020	0FE	<b>0</b>	0	0	0	0
	+0	+1	+2	-	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	19CB	0004	B000	0	005	0FE0	39A7	C00	0000	0000	0000	0000	0000	0000	00	00	0000	0000
0010	0000	0000	0000	0	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	00	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción
					,		palabra		palabra		decodificada
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	×0004]=0×	0FE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	F]=0×000	4 SP=0	×FFEE PC=0×0005						Flags
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1
4											
	Ejecución										Flags
5											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Z	С	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
(	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción	
							palabra		palabra		decodificada	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]	
	Ejecución	R7=[0:	×0004]=0×	0FE0							Flags	
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×0005	
	Ejecución	[0×FFE	F]=0×000	4 SP=0	xFFEE PC=0x0005						Flags	
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7	
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1	1
4	0006											
	Ejecución										Flags	
5												
	Ejecución										Flags	



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	РС	Instrucción decodificada
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0000	ратарга		MOV R7, [0x0004]
1					0001 1001 1100 1000	0001	0004	0002			
	Ejecución	R7=[0:	×0004]=0×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×0005
	Ejecución	[0×FFE	F]=0×000	4 SP=0	×FFEE PC=0×0005						Flags
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1
4	0006			C000							
	Ejecución										Flags
5											
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Z	С	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
C	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
C	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instr			
		٠.	[0. 12]		To parable		palabra		palabra		decod	lifica	da	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	[0x0	004]	
	Ejecución	R7=[0:	=[0x0004]=0x0FE0							Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0	005		
	Ejecución	[0×FFE	0xFFEF]=0x0004 SP=0xFFEE PC=0x0005						Flags					
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6,	R7		
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0	1	0	1
4	0006			C000	1100 0000 0000 0000									
	Ejecución										Flags			
5														
	Ejecución										Flags			



Valores	PC	SP		
iniciales	0000	FFEF		

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	F020	0FE0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra		2da palabra	PC	3ra palabra	PC	Instru decod			
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	[0×0]	004]	
	Ejecución	R7=[0:	=[0x0004]=0x0FE0							Flags				
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0	005		
	Ejecución	[0×FFE	xFFEF]=0x0004 SP=0xFFEE PC=0x0005							Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7			
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007								
	Ejecución										Flags			
5														
	Ejecución										Flags			



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Z	С	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
(	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción	
		٠.	[0. 12]		modification and parameter		palabra		palabra		decodificada	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]	٦
	Ejecución	R7=[0x0004]=0x0FE0							Flags	٦		
2	0002	B000 1011 0000 0000 0000 0003 0005 0004							CALL 0x0005			
	Ejecución	[0×FFE	xFFEF]=0x0004 SP=0xFFEE PC=0x0005							Flags		
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7	
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1	
4	0006			C000	1100 0000 0000 0000	0007					RET	٦
	Ejecución										Flags	٦
5												
	Ejecución										Flags	



Otra instrucción nueva:

# **RET**

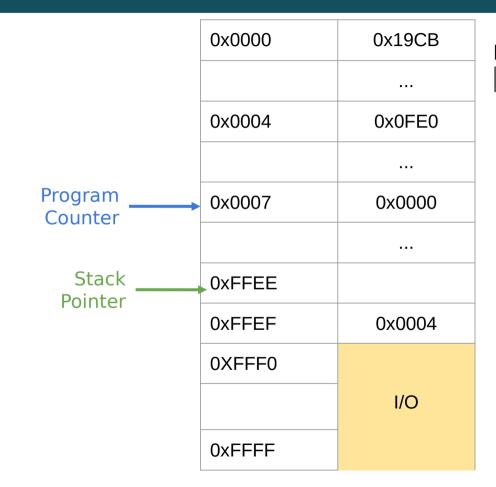
### Tipo 3: Instrucciones sin operandos

4 bits	$6 \ bits$	$6 \ bits$
cod. op.	000000	000000

operación	cod. op.	efecto
RET	1100	$PC \leftarrow [SP+1], SP \leftarrow SP + 1$

## Organización de la memoria





#### En nuestro caso:

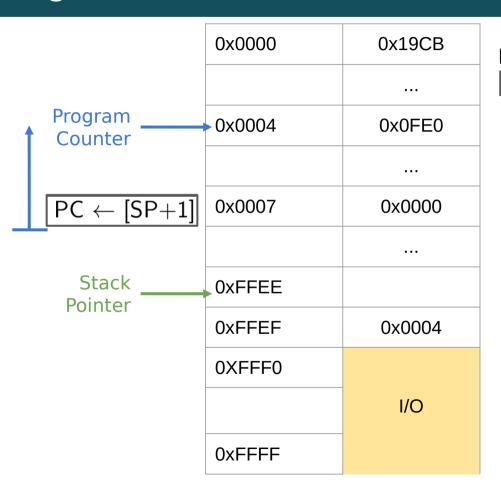
RET	1100	$PC \leftarrow [SP+1],  SP \leftarrow SP + 1$
-----	------	---

#### **RET:**

- SP + 1 = 0xFFEF
- PC = 0x0007

# Organización de la memoria





#### En nuestro caso:

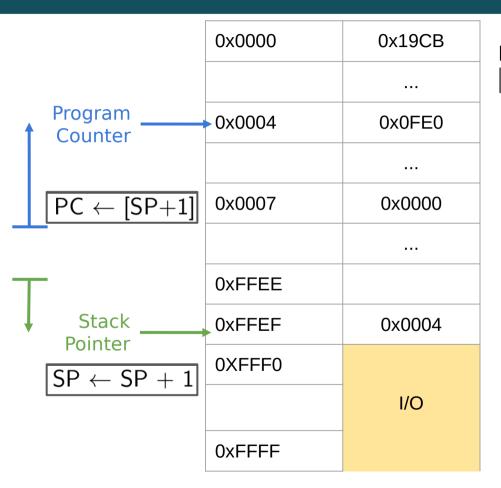
RET	1100	$\mid$ PC $\leftarrow$ [SP+1], SP $\leftarrow$ SP +	1

#### **RET:**

- $\mathbf{SP} + \mathbf{1} = \mathbf{0} \mathbf{x} \mathbf{F} \mathbf{F} \mathbf{F}$
- PC = 0x0007

## Organización de la memoria





#### En nuestro caso:

RET	1100	$\mid$ PC $\leftarrow$ [SP+1], SP $\leftarrow$ SP $+$ 3

#### **RET:**

- SP + 1 = 0xFFEF
- PC = 0x0007



#### Volviendo a la planilla:

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	F020	0FE0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción 1er nalabra	PC	2da	PC	3ra	PC	Instr	ıcció	n	
	1	31	[31 +1]	111	0011 1001 1010 0111 020 1100 0000 0000 0	1.0	palabra	10	palabra	1.0	decodificada			
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0		004]	
	Ejecución	R7=[0:	7=[0x0004]=0x0FE0							Flags				
2	0002									CALL 0×0005				
	Ejecución	[0×FFE	xFFEF]=0x0004 SP=0xFFEE PC=0x0005							Flags				
3	0005	FFEE 0004 39A7 0011 1001 1010 0111 0006						SUB R6,	R7					
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET			
	Ejecución	PC=[0	PC=[0xFFEF]=0x0004 SP=0xFFEF						Flags					
5														
	Ejecución										Flags			



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Ζ	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
(	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	РС	Instruc		
							palabra		palabra		decodif		
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [	0×000	4]
	Ejecución	R7=[0:	×0004]=0×	0FE0							Flags		
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×00	05	
	Ejecución	[0×FFE	F]=0×000	4 SP=0	FFEE PC=0x0005						Flags		
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R	7	
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0	1 0	1
4	0006			C000	1100 0000 0000 0000	0007					RET		
	Ejecución	PC=[0	×FFEF]=0	×0004 S	P=0xFFEF						Flags		
5	0004												
	Ejecución										Flags		



Valores	PC	SP	R0	
iniciales	0000	FFEF	0000	

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	F020	0FE0

Z	C	V	N
0	0	0	0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instri	ıcció	n	
	10	31	[31 +1]	IIX	mstruccion - Ter parabra	10	palabra	10	palabra	10	decod	ificad	da	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	[0×0]	004]	
	Ejecución	R7=[0:	<0004]=0×	0FE0							Flags			
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0	005		
	Ejecución	[0×FFE	F]=0×000	4 SP=0:	xFFEE PC=0x0005					Flags				
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6,	R7		
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET			
	Ejecución	PC=[0	xFFEF]=0	×0004 S	P=0xFFEF						Flags			
5	0004	FFEF	0000											
	Ejecución										Flags			



Valores	PC	SP
iniciales	0000	FFEF

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	F020	0FE0

Ζ	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
C	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
C	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción	
		٥.	[5, 12]		motracción Ter parabra		palabra		palabra		decodificada	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]	
	Ejecución	R7=[0:	×0004]=0×	0FE0							Flags	
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005	
	Ejecución	[0×FFE	0xFFEF]=0x0004 SP=0xFFEE PC=0x0005 Flags									
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7	
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1	
4	0006			C000	1100 0000 0000 0000	0007					RET	
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									Flags	
5	0004	FFEF	0000	0FE0								
	Ejecución									Flags		



Valores	PC	SP	
iniciales	0000	FFEF	

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Ζ	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
(	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0	<0004]=0×	0FE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×0005
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005 Flags									
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1
4	0006			C000	1100 0000 0000 0000	0007					RET
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF Flags									Flags
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000						
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Ζ	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
(	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada		
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]	٦	
	Ejecución	R7=[0	<0004]=0×	0FE0							Flags	٦	
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×0005		
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005 Flags										٦	
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7		
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1		
4	0006			C000	1100 0000 0000 0000	0007					RET		
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF Flags									Flags		
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000	0005							
	Ejecución		Flags							Flags			



Valores	PC	SP	RO
iniciales	0000	FFEF	00

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	F020	0FE0

Z	С	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
(	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	РС	Instrucción			
							palabra		palabra		decodificada			
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]			
	Ejecución	R7=[0		Flags										
2	0002			B000	1011 0000 0000 0000	0003	0005	0005 0004			CALL 0x0005			
	Ejecución	[0×FFE	F]=0×000		Flags									
3	0005	FFEE		SUB R6, R7										
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1			
4	0006			C000	1100 0000 0000 0000	0007					RET			
	Ejecución	PC=[0:	xFFEF]=0	×0004 S	P=0xFFEF						Flags			
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000	0005					Instrucción Inválida			
	Ejecución										Flags			



Valores	PC	SP	R
iniciales	0000	FFEF	0

R0	R1	R2	R3	R4	R5	R6	R7
0000	0000	0000	0000	0000	0000	F020	0FE0

Ζ	C	V	N
0	0	0	0

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
(	0000	19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
(	0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucción			
		٠.	[0. 12]		modification and parameter		palabra		palabra		decodificada			
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]			
	Ejecución	R7=[0:	×0004]=0×		Flags									
2	0002			B000	1011 0000 0000 0000	0003	0003 0005 0004				CALL 0x0005			
	Ejecución	[0×FFE	F]=0×000		Flags									
3	0005	FFEE	0004		SUB R6, R7									
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0 1 0 1			
4	0006			C000	1100 0000 0000 0000	0007					RET			
	Ejecución	PC=[0	xFFEF]=0	×0004 S	P=0xFFEF						Flags			
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000	0005					Instrucción Inválida			
	Ejecución	FIN DE	E LA EJEC		Flags									



							F	lani	lla de	Se	egui	mient	0	F020	0FE0	)						
			PC	SP SP		RO		11	R2		13	R4	R5	R6	R7		Z	C	V	N		
V	alores i	niciales:	000	0 FFEF	:	000	00 00	000	0000 00		000	0000	0000	0000	0000		0	0	0	0	)	
	[		+0	+1	+2	+3	+4	+5	+6	$\perp$	+7	+8	+9	+A	+B	+C	+D		+E	+	F	1
Mer	moria:	0000	19C	8 0004	B000	0005	0FE0	39A7	C00	0 0	0000	0000	0000	0000	0000	0000	000	0 0	000	00	00	
		0001	0000	0000	0000	0000	0000	0000	0000	0	0000	0000	0000	0000	0000	0000	000	0 0	000	00	00	
	PC	PC SP [SP+1] IR Instrucción - 1 <sup>cr</sup> palabra (en bits)		P	PC 2 <sup>da</sup> palabra		PC	3 <sup>era</sup> palabra	P(			nstru codi										
1	0000	) F	FEF	0000 19C8 0001 1001 1100 1000 0001 0004 0002									MOV	R7,	[0x0	004	4]					
	E	jecuci	ón	R7 = [0x	R7 = [0x0004] = 0x0FE0									Flags <sup>1</sup>	z	С	٧	N				
2	0002	2			B000 1011 0000 0000 0000 0003 0005 0004									CALL	0x0	005						
	E	jecuci	ón	[0xFFEF]	= 0x000	4 9	SP = 0xF	FEE	PC:	= 0x	0005							Flags <sup>1</sup>	Z	С	V	N
3	0005	5 F	FEE	0004	39A7	001	1 100	1 101	0 0111	l	00	06						SUB	R6, R	7		
	E	jecuci	ón	R6 = 0x0	0000 - 0x	0FE0 =	0xF020	)										Flags <sup>1</sup>	z 0	1	v O	1 N
4	0006	006 C000 1100 0000 0000 0007									RET											
	E	jecuci	ón	PC = [0x	FFEF] = (	] = 0x0004 SP = 0xFFEF									Flags <sup>1</sup>	Z	С	٧	N			
5	0004	4 F	FEF	0000	0FE0	000	0 111	1 111	0 000	)	00	05						Instru	ıcció	n In	válio	la
	E	jecuci	ón	FIN DE LA	A EJECUC	ΙÓΝ.												Flags <sup>1</sup>	Z	С	٧	N

## ¿Qué conocimientos tenemos que tener en cuenta?



Cómo codificar y decodificar instrucciones usando la cartilla de Orga 1



Cómo calcular las palabras que ocupa cada instrucción en un programa



Cómo calcular las etiquetas de un programa



Cómo calcular el desplazamiento en los saltos



Ciclo de instrucción y cómo hacer un seguimiento de programas



Cómo realizar el seguimiento de programas con CALL