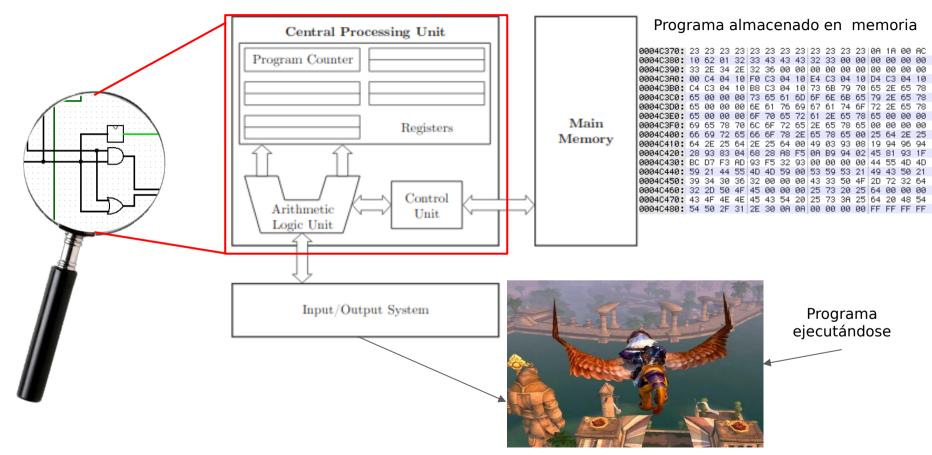
Organización del Computador I

Ciclo de Instrucción Codificación de instrucciones memoria

1c2022

Universidad de Buenos Aires - FCEyN - Departamento de Computación

¿Qué buscamos comprender?



¿Qué vamos a ver hoy?

1

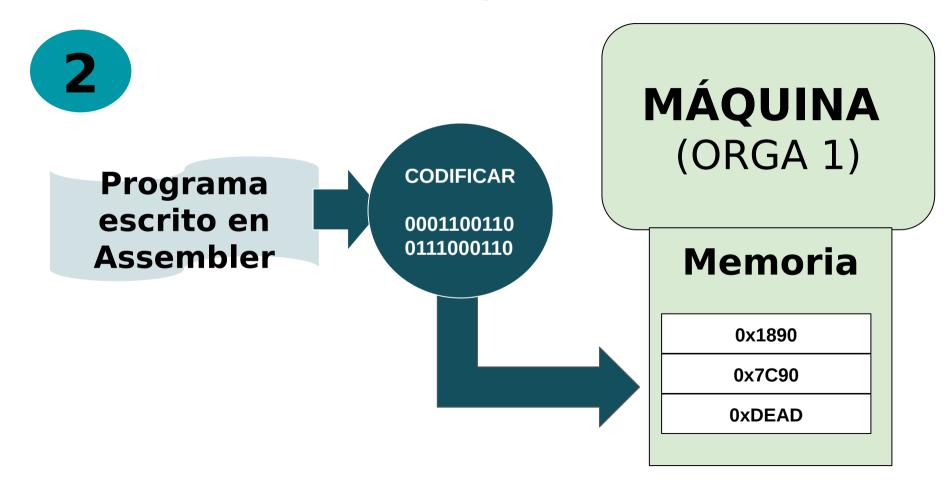
Usaremos la cartilla o "manual" de ORGA 1

Programa escrito en Assembler CODIFICAR
0001100110
0111000110

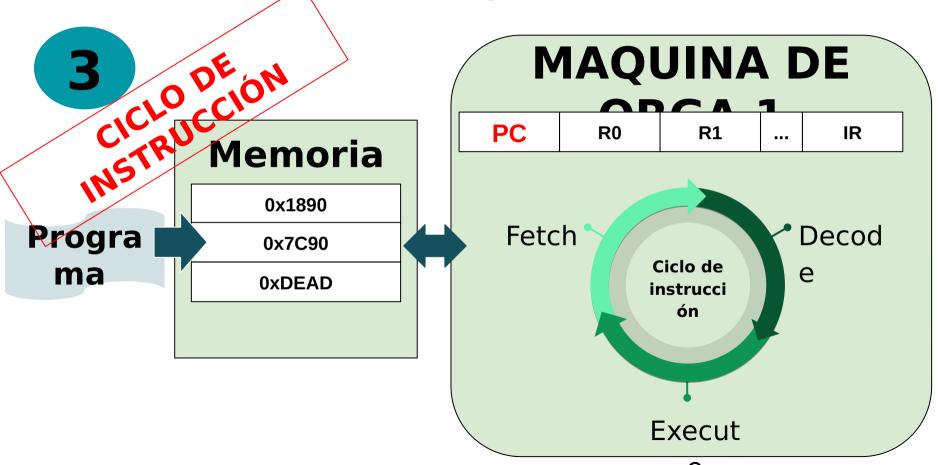
MOV R1, 0x02

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

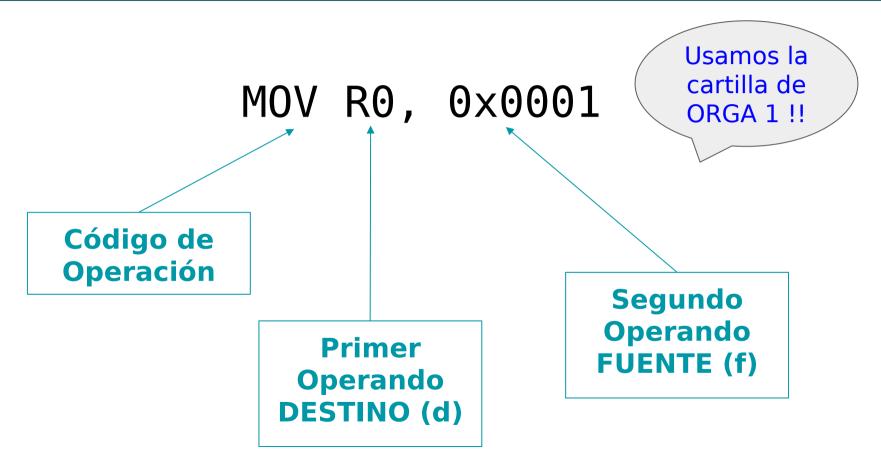
inc: DW 0x0002



Veamos la primer instrucción del programa:

MOV R0, 0×0001







MOV RO, 0×0001

Tipo de instrucción

Formato de instrucción

Tipo 1: Instrucciones de dos operandos

4 bits	b bits	b bits	1b $bits$	16 bits
cod. op.	destino	fuente	constante destino (opcional)	constante fuente (opcional)



MOV R0, 0×0001

Format

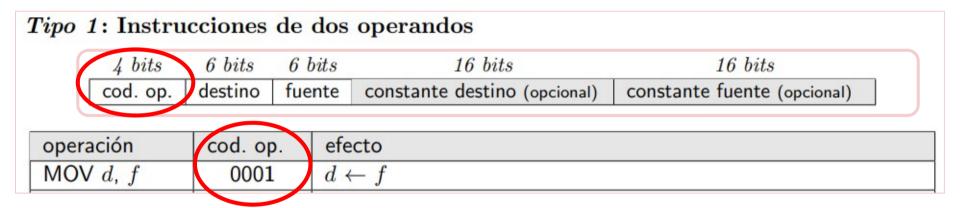
Tipo 1: Instrucciones de dos operandos

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	$d \leftarrow f$

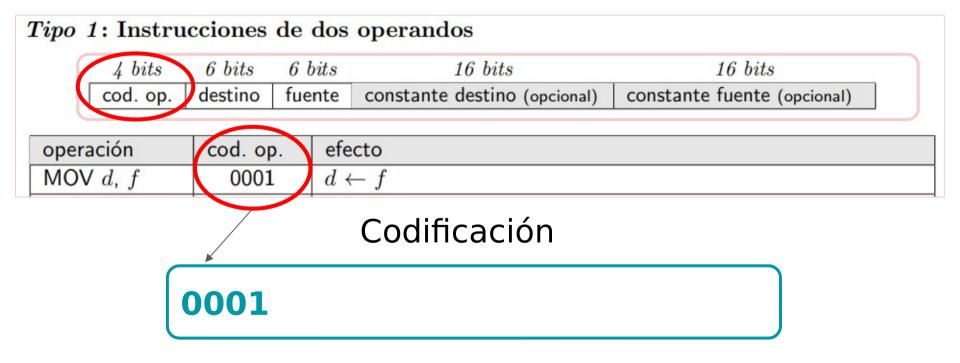


MOV RO, 0×0001



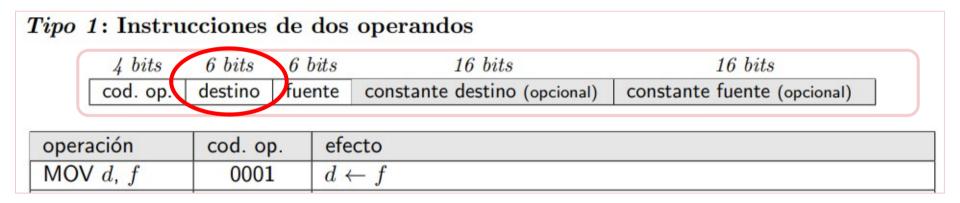


MOV RO, 0×0001





MOV RO, 0×0001



Codificación

0001



MOV RO, 0×0001

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]

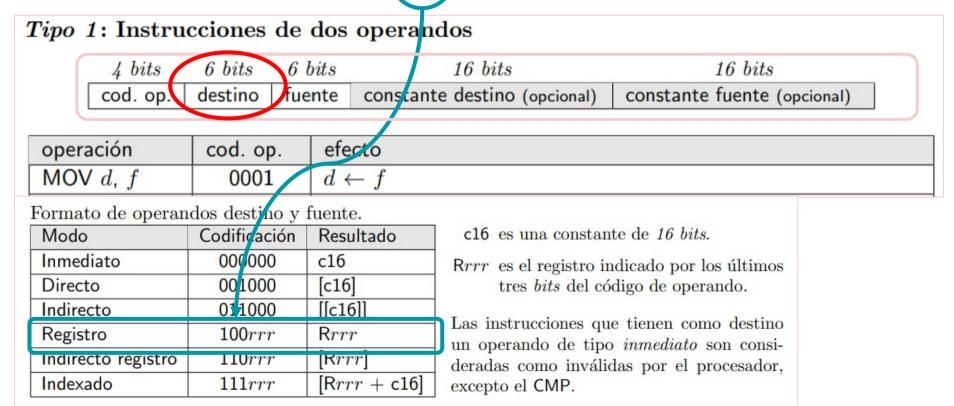
c16 es una constante de 16 bits.

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.

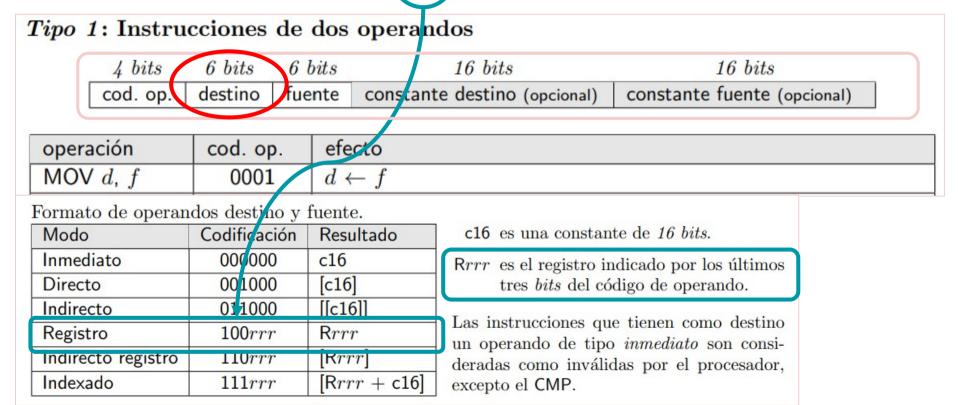


MOV (RO), 0x0001





MOV (RO), 0x0001







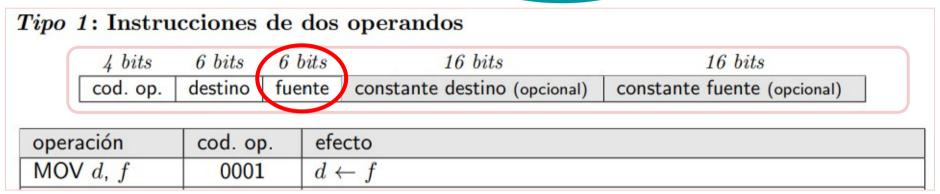


Codificación

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

	$6\ bits$			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.

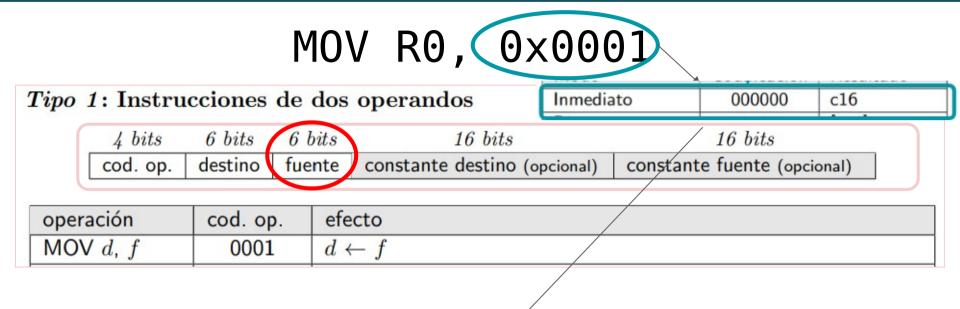
Modo	Cod ficación	Resultado
Inmediato	000000	c16
Directo	001000	[C10]
Indirecto	011000	[[c16]]
Registro	100rrr	Rrrr
Indirecto registro	110rrr	[Rrrr]
Indexado	111rrr	[Rrrr + c16]

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.

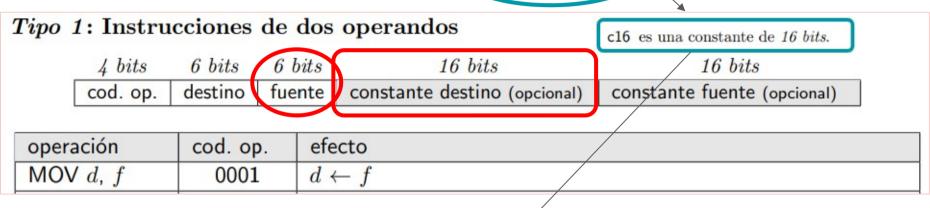




Codificación







Codificación



MOV RO, 0×0001

Codificación



MOV RO, 0×0001

Codificación

0001 1000 0000 0000 0000 0000 0000 0001

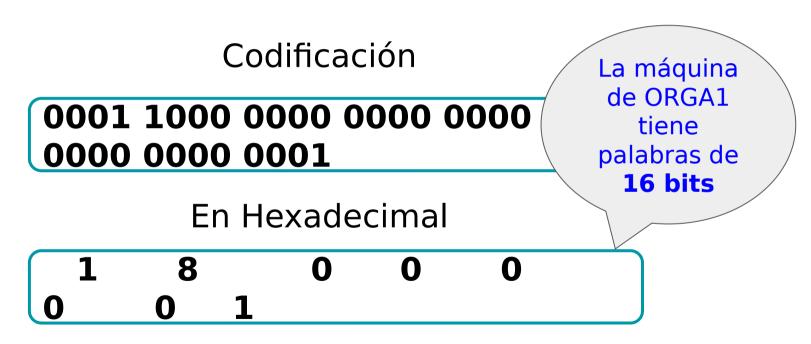
En Hexadecimal

1 8 0 0 0 0 0 1

¿Cómo lo guarda en memoria?

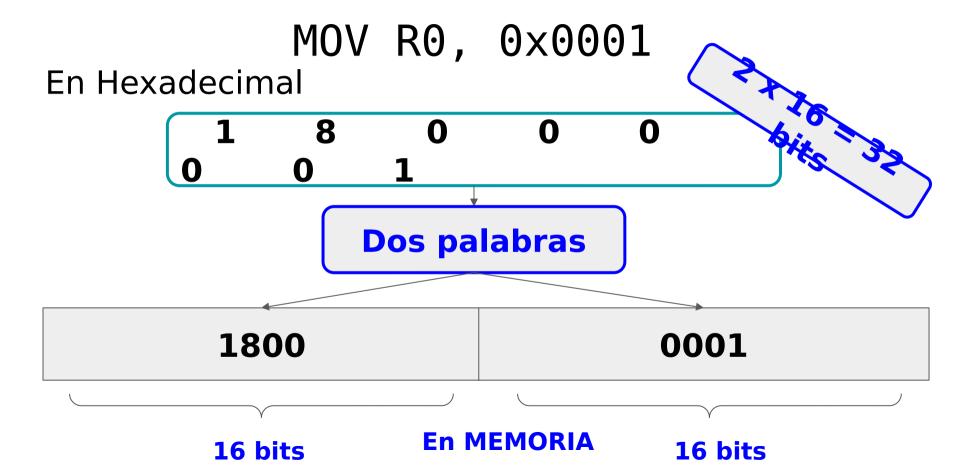


MOV RO, 0×0001



¿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	A000	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					
000					0000	0005
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)
CIID 1 f	0011	1 . 1 f (

Código de operació

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.

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.

Segundo Operando



Veamos la siguiente instrucción de un programa:

ADD R0, [inc]

Formato de operandos destino y fuente. Modo Codificación Resultado Inmediato c16 000000 Directo 001000 [c16]Indirecto 011000 [[CTD]] Registro 100rrrRrrr[Rrrr]Indirecto registro 110rrrIndexado [Rrrr + c16]111rrr

c16 es una constante de 16 bits.

tres

Las instru

un operan

deradas co

excepto el

 $\mathsf{K}rrr$ es el registro indicado por los ultimos

main: MOV R0, 0x0001

ADD R0, [inc]

JMP main

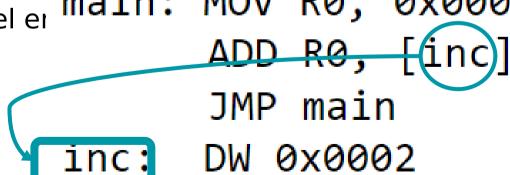
inc: DW 0x0002

Segundo Operando

Etiquetas



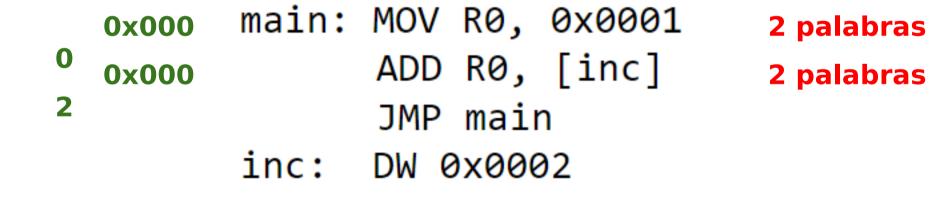
- Las etiquetas son renombres de direcciones de memoria.
- Al momento de programar en Assembler, es imposible saber en qué dirección de memoria se va a guardar la información del programa.
- Las etiquetas nos ayudan a que podamos referirnos a posiciones de n el el main: MOV RO, 0x0001 ama y son resueltas por el el el main: MOV RO, 0x0001



Cálculo de etiquetas



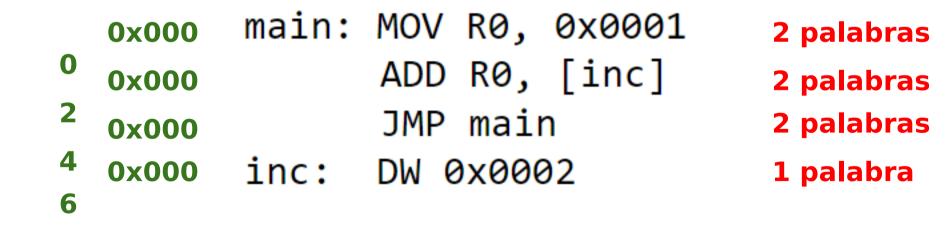
¿En qué posición de memoria va a quedar cada etiqueta?
 Calculemos cuántas palabras (16 bits en la máquina de Orga1) ocupa cada instrucción.



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.



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:

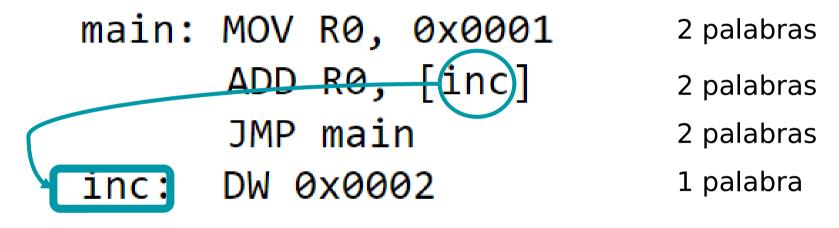
main 0x0000 inc 0x0006

	0x000	main:	MOV R0, 0x0001	2 palabras
	0x000		ADD R0, [inc]	2 palabras
2	0x000		JMP main	2 palabras
4	0x000	inc:	DW 0x0002	1 palabra
6				

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,

[0x0006]

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 001

<u>"inc" 0x0006 er</u>



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

1010 000000 000000 00000000000000000



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
ENSAMBLA
DOR

Directiva	Efecto
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 0x0007
0 \times 0007
                  DW 0x0004
8000x0
          loop: CMP R1,0x0001
                  JL fin
0 \times 000A
0 \times 000B
                  JGE main
0 \times 000C
           fin: RFT
```

Dirección Etiqueta fin = 0x000C

Dirección instrucción = 0x000B

Desplazamiento: 1 El desplazamiento es positivo (decimal) porque avanza en el programa



```
0x0000 main: MOV R1,[[et1]]
                  ADD [R1],0x6000
0 \times 0002
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
```

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 ?
```

Solucion calculo de desplazamiento JGE main



```
0x0000 main: MOV R1,[[et1]]
                 ADD [R1],0x6000
0 \times 0002
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 000 A
0×000B
                 JGE main
0 \times 000C fin: RET
```

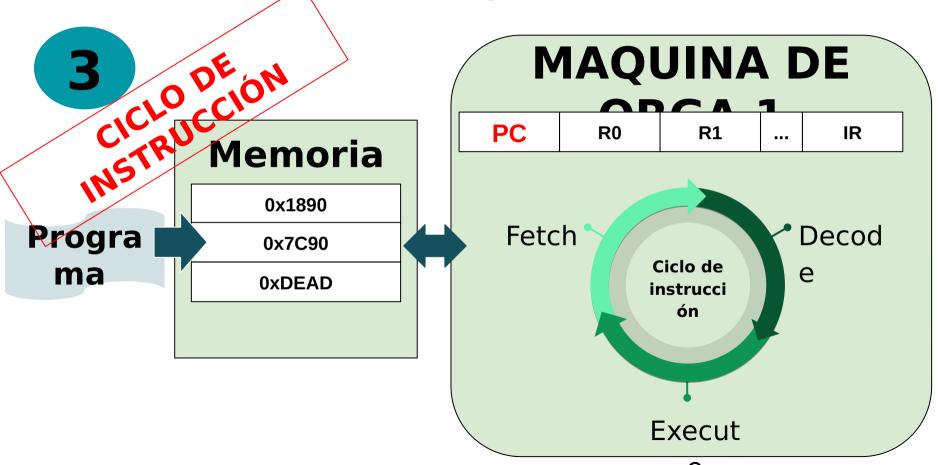
Dirección Etiqueta main = 0x0000

Dirección instrucción = 0x000C

Desplazamiento: -12 (decimal) = 0xF4

Intervalo 5 minutos

¿Qué vamos a ver hoy?



Máquina de Orga 1



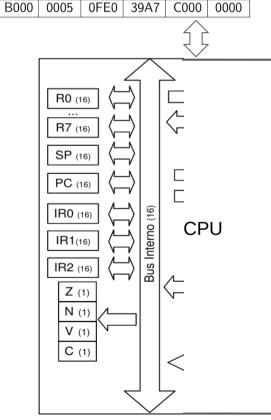
0006

0007

Práctica 3 - Arquitectura ORGA1

Descripción General

- Palabras de 16 bits.
- Direccionamiento a palabra.
- Espacio direccionable de 65536 palabras.
- Espacio de direcciones dedicado a entrada/salida en las direcciones 0xFFF0 0xFFFF.
- Do 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 calcular 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.
- ▷ De las que alteran los flags, todas dejan C y V en cero, excepto ADD, ADDC, SUB, CMP y NEG.



0000

19C8

0001

0004

0002

0003

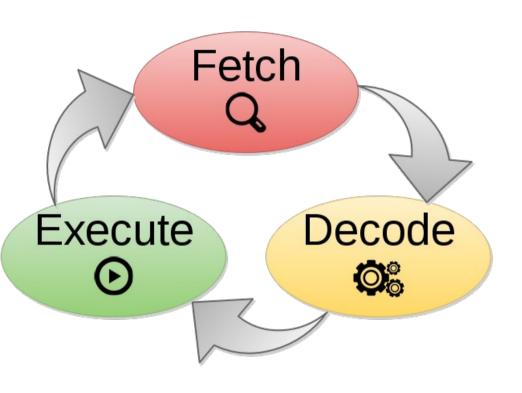
0004

0005

39A7

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																	
V	alores i	niciales:	PC	SP		RO	F	11	R2	R3	R4	R5	R6	R7		Z	С	V	N		
	Г		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D		E	+	F	1
Mor	noria:			,,			, ,			- ''	10	1.5	, A	.,,			1				
Wici	noria.																				
	PC	-	SP	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$												stru					
1		Ť								Ť	Ť								_		乛
	E;	jecució	n													F	lags ¹	Z	С	٧	N
2																					
	E,	jecució	n	1		•							•		'	F	lags ¹	Z	С	V	N
3																					
	E;	jecució	n													F	lags ¹	Z	С	V	N
4																					
	E,	jecució	n													F	lags ¹	Z	С	V	N
5																					
	E,	jecució	n													F	lags ¹	Z	С	V	N
6																					
	E	jecució	n													F	lags ¹	Z	С	V	N
7																					
	E	jecució	n													F	lags ¹	Z	С	٧	N
8																					
	E	jecució	n													F	lags ¹	Z	С	V	N

¹Sólo deben completarse luego de la ejecución de una instrucción que los altere.



							F	Planil	la de	Segu	imier	nto								
	V	alores inicia	eles:	SP		RC	I	R1	R2	R3	R4	R5	R6	R7		Z	C I	/ 1	N	
			+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	-	+F	
•	Men	noria:							+				-							
Catada da las																				
Estado de los		PC	SP	[SP+1]	IR	Ins	trucció (e	n - 1 ^{er} n bits)	palab	ra	PC	2 ^{da} palabra	PC	3 ^{era} palabra	PC		Instrucción decodificada			
registros	1					İ				T	İ				Ì					
		Ejec	ución			<u> </u>								1		F	lags ¹	С	V N	
	2																			
		Ejec	ución							'						F	lags ¹ Z	С	V N	
	3																			
		Ejec	cución										F	lags ¹	С	V N				
	4																			
		Ejec	ución											1		F	lags ¹ Z	С	V N	
	5																•			
		Ejec	ución													F	lags ¹ Z	С	V N	
	6																			
		Ejec	ución													F	lags ¹	С	V N	
	7																			
		Ejec	ución													F	lags ¹ Z	С	V N	
	8																			
		Ejec	ución													F	lags ¹ Z	С	V N	

¹Sólo 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 inicia		PC	SP		RO	R1		R2	R3	R4		R5	R6	R7	-	Z	С	V	N	1		
			+) +	. +2	+	-3	+4	+5	+6	+7	+8	3	+9	+A	+B	+C	+D	+	Е	+F			
	Mer	moria:																						
ţ		PC	SP	[SP+:	ı II	l	Instru	ıcción · (en t		palabr	a	PC	2 ^{da} palab		PC	3 ^{era} palabra	PC		Instrucción decodificada					
Memoria	1																							
		Ejec	ución										F	lags ¹	Z	C V	N							
	2																							
		Ejec	ución												FI	$lags^1$	Z	C V	N					
	3																							
		Ejec	ución											F	lags ¹	Z	C V	N						
	4																							
		Ejec	ución															FI	$lags^1$	Z	C V	N		
	5																							
		Ejec	ución															FI	lags ¹	Z	C V	N		
	6																							
		Ejec	ución		·						,							FI	$lags^1$	Z	C V	N		
	7																		•					
		Ejec	ución		•	•								·				FI	$lags^1$	Z	C V	N		
	8																				•			
		Ejec	ución															F	lags ¹	Z	C V	N		

¹Sólo deben completarse luego de la ejecución de una instrucción que los altere.



								I	Plan	illa de	e Segu R3	imie	nto									
	V	alores ini	iciales:	PC	SP		RO	1	R1	R2	R3	R4	R.	R6	R7		Z	С	V	N		
												Ц					L		<u> </u>	<u></u>		
				+0	+1	+2	+3	+4	+5	+6	+7	+	8 +	9 +A	+B	+C	+[)	+E	+F	7	
	Mer	noria:												-								
		L					1 -			er					1			Щ.				
		PC	:	SP	[SP+1]	IR	Ins		n - 1 en bits	^{er} palab)	ra	PC	2 ^{da} palabra	PC	3 ^{era} palabr	a PC	:		nstru ecodi			
	1																					
		Ej	ecució	n	l								1					Flags	2 Z	С	٧	N
	2																			•		
V		Ej	ecució	n										·				Flags	1 Z	С	V	N
Ciclo de	3																					
Instrucción		3 Ejecución		n										-				Flags	1 Z	С	V	N
	4																İ					
	-	Ej	ecució	n														Flags	1 Z	С	V	N
	5																					
		Ej	ecució	n														Flags	2 Z	С	V	N
	6																					
		Ej	ecució	n														Flags	2 Z	С	V	N
	7																					
		Ejecución		n														Flags	2 Z	С	V	N
	8																					
	Ejecución																	Flags	1 Z	С	v	N

¹Sólo deben completarse luego de la ejecución de una instrucción que los altere.

Ejercicio



main: MOV R1,[[et1]]

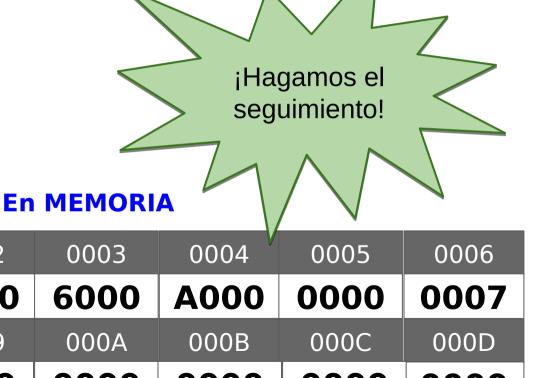
ADD [R1],0x6000

JMP main

DW 0x0007 et1:

DW 0x0004

fin: **RET**



2C40



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											
	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

UC	00	0	0	0	0	
	+C	+	D	+E	+F	
$\overline{}$	0000	~~	00	0000	0000	٦.

	+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	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								
	Ejecución										Flags
2											
	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	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	1858							
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	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	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



Valores PC SP iniciales 0000 FFEF

				R4			
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	ecución									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	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	ón									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	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	parabra		MOV R1, [[0×0006]]
1		111	0000	1030	0001 1000 0101 1000	0001	0000	0002			
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución	1									Flags



Valores	PC	SP
iniciales	0000	FFEF

					R5		
0000	0000	0000	0000	0000	0000	0000	0000

2	v	1.71
0 0	0	0

7 C V N

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	0000	0000	0000
001	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	R1=[[0x0006]]=[0x0007]=4									Flags		
2													
	Ejecución												
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	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002	Pereere		MOV R1, [[0x0006]]
	Ejecución	R1=[[0)×0006]]=	0×0007]							Flags
2	0002										
	Ejecución										Flags
3											
╗	Ejecución	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	R1=[[0	×0006]]=[0×0007]	=4						Flags
2	0002			2C40							
	Ejecución										Flags
3											
	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	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

					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	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	0010 1100 0100 0000	0003					
	Ejecución										Flags
3											
	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	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
									parabra		
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			
	Ejecución										Flags
3											
	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	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	PC 3ra		Instrucción
		٥,	[51 1]	""	mistracción - Ici palabra		palabra	' `	palabra	PC	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										Flags
3											
	Ejecución	ıción								Flags	



Valores	PC	SP
iniciales	0000	FFEF

				R4			
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,	[5, 12]		motracción 20 parabra		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	6000	0004			ADD [R1],0×6000
	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
	1	31	[31 +1]		mstrucción - Ter palabra	1	palabra	1	palabra	10	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			ADD [R1],0x6000
	Ejecución	[0×000	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

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	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	×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										
	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]	ID	R Instrucción - 1er palabra		2da	PC	3ra	PC	Instrucción
		31	[31-1]	IIX	mstrucción - Ter palabra	PC	palabra	1	palabra	1	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	1=[[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							
	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	mstruccion - ter palabra	PC	palabra	FC	palabra	1	decodificada
1	0000	FFEF	0000	1858	0001 1000 0101 1000	0001	0006	0002			MOV R1, [[0x0006]]
	Ejecución	R1=[[0	×0006]]=[Flags						
2	0002			2C40	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×0004	[0x0004]=0xA000+0x6000=0x0000						Flags 1 1 0 0		
3	0004			0000	0000 0000 0000 0000						
	Ejecución	jecución									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	[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

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
	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	[0×000	4]=0×A00	0+0×60	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000	0005					
	Ejecución	n									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		2da	PC	3ra	PC	Instrucción
	10	31	[51 +1]	111	mstruccion - Ter palabra	PC	palabra	1.0	palabra	1.0	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 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

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		2da	PC	3ra	PC	Instrucción		
	1.0	31	[51 +1]	"	mstruccion - Ter palabra	PC	palabra	' '	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	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 LA EJECUCIÓN									Flags		



							. 0000	Dlanil	la de S	Zomi	mion	to								
			PC	SP	_ (RO			R2	R3	R4	R5	R6	R7		Z	C	V	N	
V	alores i	niciales:	000	0 FFEF	: \	000	00 00	00	0000	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	3 0006	2C40	6000	-A000	0000	0007	0004	0000	0000	0000	0000	0000	000	0 0	000	000	00
		0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0 0	000	000	Ю
	PC		SP	[SP+1]	IR	Ins		n - 1 ^{er} n bits)	palabra	F	PC .	2 ^{da} palabra	PC	3 ^{era} palabra	P	c			cció ficad	
1	0000) F	FEF	0000	1858	000	1 1000	010	1 1000	00	01	0006	0002				MOV	/R1,	[[0x0	006]]
	E	jecuci	ón	R1 = [[0x	(0006]] =	= [0x00	07] = 4										Flags ¹	Z	С	V N
2	0002	2			2C40	001	0 110	010	0000	00	03	6000	0004				ADD	[R1]	, 0x6	000
	Ejecución [0x0004] = 0xA000 + 0x6000 = 0x0000									Flags ¹	z 1		v N 0 0							
3	0004 0000 0000 0000 0000 0005								Instr	ucció	n inv	álida								
	Ejecución FIN DE LA EJECUCIÓN											Flags ¹	Z	С	V N					

Seguimiento - Simulando el ciclo de instrucción



¿Qué herramientas de análisis tenemos?

• Lapiz y papel: una planilla de seguimiento

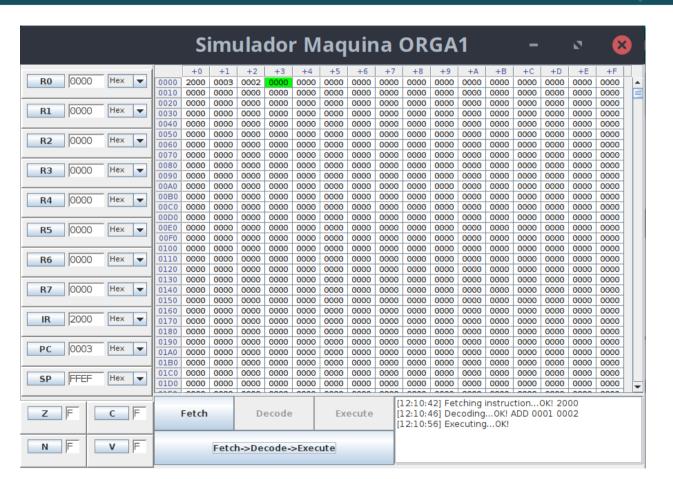


• El simulador



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

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7	Z	С	V	N
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	0FE0	0	0	0	0

	-	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	0 19	C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001	0 00	000	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											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución						Flags				



Valor	es	PC	5	SP		R0	R1	.	R2	R	3	R4	R5	R6	R7		Z	C	V	N
inicia	les	0000	F	FEF		000	0 00	00 (0000	0	000	0000	0000	0000	0FE	0	0	0	0	0
	+0	+	-1	+2	-	+3	+4	+5	+6	ŝ	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	19C8	00	04	B000	0	005	0FE0	39A7	C00	00	0000	0000	0000	0000	0000	0000	00	00	0000	0000
0010	0000	00	00	0000	0	000	0000	0000	000	0	0000	0000	0000	0000	0000	0000	00	00	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										
	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=[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:	R7=[0x0004]=0x0FE0								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	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
00	000	19C8	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
00)10	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					
	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		2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000		0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	7=[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	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		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



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

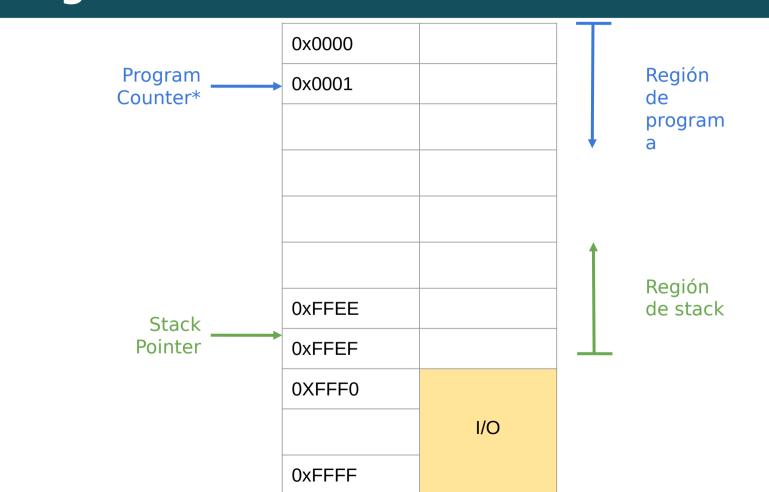
	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
1	0000	FFEF	0000	19C8	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	×0004]=0×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución						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





Repasemos la organización de la memoria...







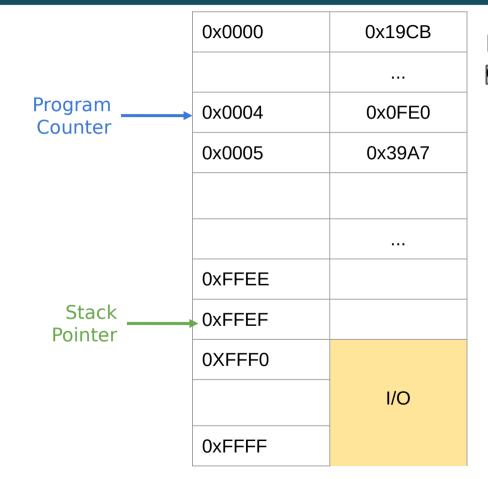
Volviendo a nuestro ejercicio:

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.





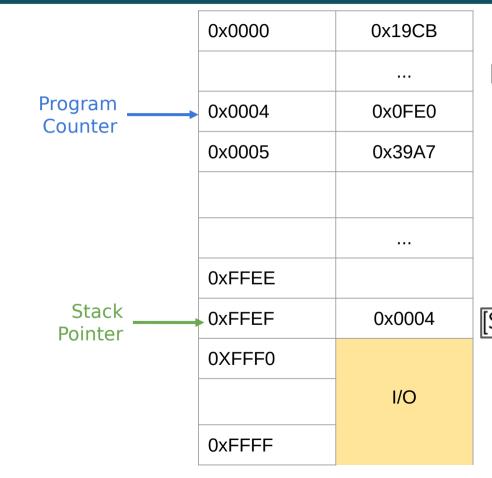
En nuestro

caso:	1011	[CD]	, DC	CD /	CD	1 DC	, f
	1011		← r c,	<u> </u>	<u> </u>	1, FC	\leftarrow J

•
$$f = 0x0005$$

$$PC = 0x0004$$





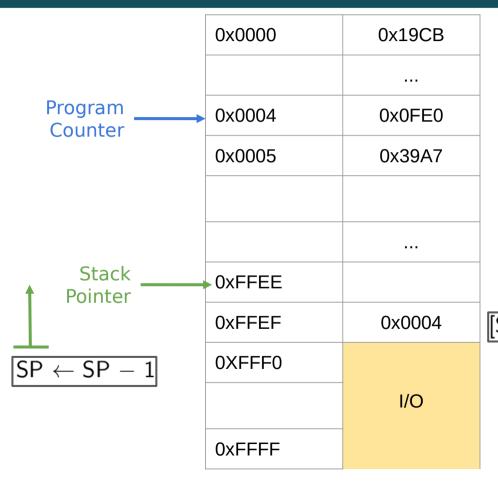
En nuestro

caso:	1011	\Box [SP] \leftarrow PC SP \leftarrow SP $-$ 1 PC \leftarrow f
CMB 9 :	1011	

•
$$f = 0 \times 0005$$

•
$$PC = 0x0004$$





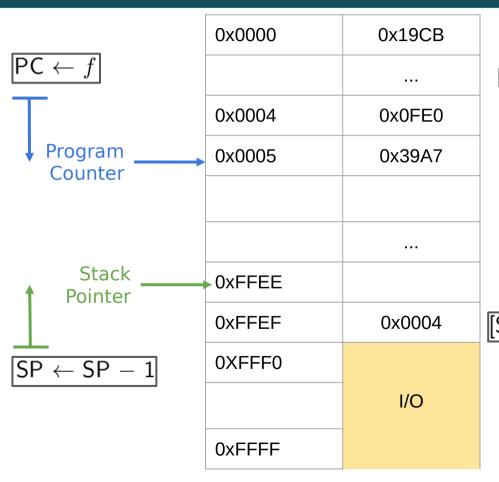
En nuestro

Caso: 1011 $ SP \leftarrow PC, SP \leftarrow SP - 1, PC \leftarrow j$

•
$$f = 0 \times 0005$$

•
$$PC = 0x0004$$





En nuestro

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

•
$$f = 0 \times 0005$$

•
$$PC = 0x0004$$



Volviendo a la planilla:

Valores	PC	SP	R0	R1	R2	R3	R4	R5	R6	R7
iniciales	0000	FFEF	0000	0000	0000	0000	0000	0000	0000	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	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 0×0005
	Ejecución	[0×FFE	F]=0×000	4 SP=0	xFFEE PC=0x0005						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

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	C 2da		PC 3ra		Instrucción decodificada		
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002	parabra		MOV R7, [0x0004]		
	Ejecución	R7=[0x0004]=0x0FE0									Flags		
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×0005		
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005									Flags		
3	0005												
	Ejecución										Flags		
4													
	Ejecución										Flags		
5													
	Ejecución									Flags			



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

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 palabra	PC	2da	PC	3ra	PC	Instrucción
			[0., -]		поставот поставот		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	×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

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	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

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 decodif		
			-				palabra		palabra				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0	0×0004]
	Ejecución	R7=[0:	×0004]=0×	0FE0							Flags		
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x000	05	
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE 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									
	Ejecución										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
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:	×0004]=0×	0FE0							Flags			
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0	005		
	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									
	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	РС	Instru		
							palabra		palabra		decodi		
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0×0004	-]
	Ejecución	R7=[0:	×0004]=0×	0FE0							Flags		
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×00	05	
	Ejecución	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005								Flags			
3	0005	FFEE	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							
	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 palabra	PC	3ra palabra	PC	Instruc decodifi		
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 0x000)5	
	Ejecución	[0×FFE	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005										
3	0005	FFEE	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									Flags			
5													
	Ejecución									Flags			



Otra instrucción nueva:

RET

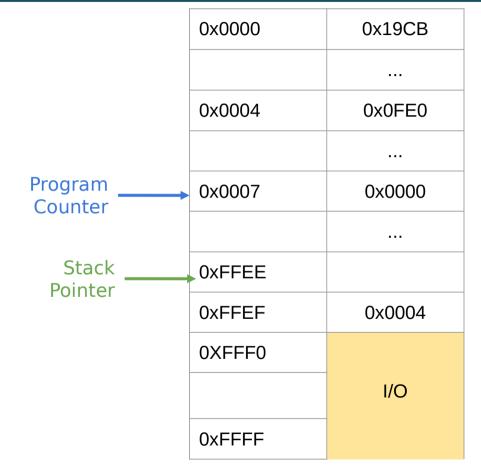
Tipo 3: Instrucciones sin operandos

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

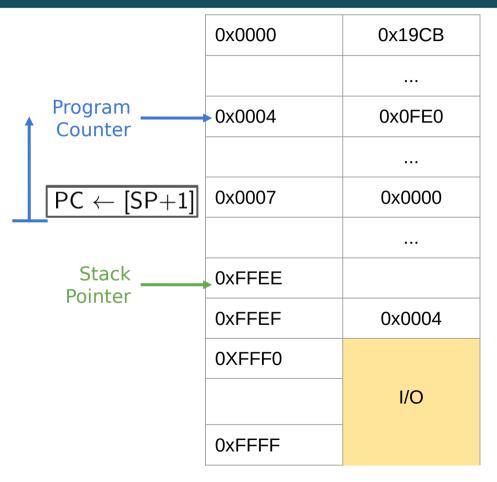
RET 1100 PC \leftarrow [SP+1], SP \leftarrow SP + 1

RET:

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

Organización de la memoria





En nuestro

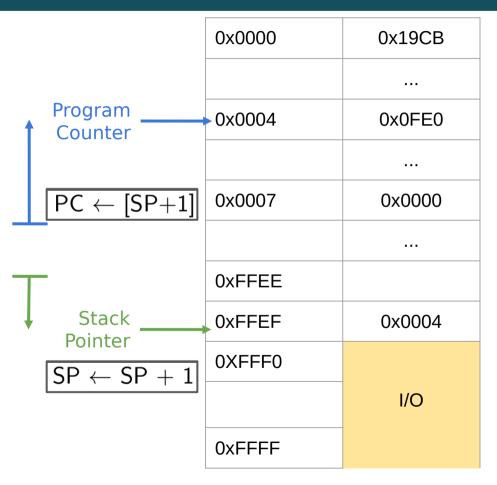
							4
RET	1100	PC ←	[SP+1]	, SP	$\leftarrow SP$	+	

RET:

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

Organización de la memoria





En nuestro

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 palabra	PC	2da	PC	3ra	PC	Instru					
		٥.	[5, 12]		motracción Ter parabra		palabra		palabra		decodificada		a			
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	[0x00	04]			
	Ejecución	R7=[0:	×0004]=0×	:0x0FE0												
2	0002										CALL 0x0005					
	Ejecución	[0×FFE	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005										Flags			
3	0005	FFEE	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			\neg		
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF									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	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	[0×FFE	F]=0×000		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	0004												
	Ejecución				Flags								



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

Z	C	V	Ν	
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	[0×FFE	F]=0×000		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	0004	FFEF	0000								
	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	PC	3ra	PC	Instrucción
	10	31	[31-1]	IIX	mstruccion - Ter parabra	10	palabra	1.0	palabra	10	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:	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	×FFEF]=0	×0004 S	P=0xFFEF						Flags
5	0004	FFEF	0000	0FE0							
	Ejecución										Flags



Valores	PC	SP
iniciales	0000	FFEF

R0							
0000	0000	0000	0000	0000	0000	F020	0FE0

E	0	0	U	0	U	
	+C	+0)	+E	+	·F
n	0000	იიი	n	იიიი	00	nn

		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0	0000	19CB	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	PC	3ra	РС	Instru			
							palabra		palabra		decodi			
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	[0×00	04]	
	Ejecución	R7=[0:	×0004]=0×	0FE0							Flags			
2	0002			B000	1011 0000 0000 0000	0003 0005 0004				CALL 0×00	005			
	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, F	27		
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags 0	1	0 1	1
4	0006			C000	1100 0000 0000 0000	0007					RET			
	Ejecución	PC=[0	×FFEF]=0	×0004 S	P=0xFFEF						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

Z	C	V	Ν	
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
_							· .		ратарта		
1	0000	FFEF	0000	19CB	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 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	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					
	Ejecución										Flags



Valores	PC	SP	
iniciales	0000	FFEF	

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	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 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	1100 0000 0000 0000	0007					RET
	Ejecución	PC=[0:	xFFEF]=0	×0004 S	P=0×FFEF						Flags
5	0004	FFEF	0000	0FE0	0000 1111 1110 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	F020	0FE0

Z	C	V	Ν
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×	0FE0							Flags
2	0002			B000	1011 0000 0000 0000	0003 0005 0004					CALL 0×0005
	Ejecución	[0×FFE	F]=0×000		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	0FE0	0000 1111 1110 0000	0005					Instrucción Inválida
	Ejecución	FIN DE	E LA EJEC	CUCIÓN							Flags



							F	Plani	lla de	Segui	mient	0	F020	0FE0							
			PC	SP		RO		11	R2	R3	R4	R5	R6	R7		Z	C	V	N		
٧	/alores ini	ciales:	0000	FFEF	=	000	00 00	000	0000	0000	0000	0000	0000	0000	7	0	0	0	0		
			+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	1	+E	+	F	
Mer	moria:	0000	19C8	0004	B000	0005	0FE0	39A	7 C000	0000	0000	0000	0000	0000	0000	0000	00	000	00	00	
	(0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	000	00	00	
	$\begin{array}{ c c c c c c c c c c c c c c c c c c c$												Instrucción decodificada								
1	0000 FFEF 0000 19C8 0001 1001 1100 1000 0001 0004 0002												- 1	MOV	R7,	[0x0	004]			
	Ejecución R7 = [0x0004] = 0x0FE0												F	lags ¹	Z	С	٧	N			
2	0002				B000	101	1 000	0 000	0000	00	03 (0005	0004			(CALL	0x0	005		
	Eje	cució	n	[0xFFEF]] = 0x00	04 5	SP = 0xF	FEE	PC=	0x0005	·					F	lags ¹	Z	С	٧	N
3	0005	FF	EE	0004	39A7	001	1 100	1 101	0 0111	00	06						SUBF	R6, F	7		
	Eje	cució	n	R6 = 0x0	0000 - 0	k0FE0 =	0xF020)								F	lags1	z 0	с 1	v 0	1 N
4	0006 C000 1100 0000 0000 0007										1	RET									
	Eje	cució	n	PC = [0x	FFEF] =	0x0004	SP	= 0xF	FEF	_						F	lags ¹	Z	С	٧	N
5	0004	FI	FEF	0000	0FE0	000	0 111	1 111	0000	00	05						Instru	ıcció	n Inv	/álic	a
	Eje	cució	n	FIN DE LA	A EJECUC	IÓN.				-	•				•	F	lags1	Z	С	٧	N

¿Que 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