ORGA1: Arquitectura y Seguimiento

Organización del Computador I

Verónica Coy

Departamento de Computación - FCEyN UBA

Primer Cuatrimestre 2019 - Turno Tarde

 Programación
 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierro

 ●0
 0000000
 0000
 00000
 00000

Ciclo de Vida de un Programa

 Programación
 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre

 ●0
 0000000
 0000
 00000
 00000

Ciclo de Vida de un Programa

Ciclo de Vida de un Programa

Programación: Escribir un algoritmo en lenguaje ensamblador (o assembly). Esto es el código fuente.
 Programación
 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre

 ●0
 0000000
 0000
 00000
 00000

Ciclo de Vida de un Programa

- Programación: Escribir un algoritmo en lenguaje ensamblador (o assembly). Esto es el código fuente.
- Ensamblado: Un programa llamado Ensamblador toma el código fuente y lo traduce a un código máquina.

Ciclo de Vida de un Programa

Programación

- Programación: Escribir un algoritmo en lenguaje ensamblador (o assembly). Esto es el código fuente.
- Ensamblado: Un programa llamado Ensamblador toma el código fuente y lo traduce a un código máquina.
 - Resuelve las directivas dirigidas al **Ensamblador**.
 - Calcula el tamaño de cada instrucciones, y resuelve las etiquetas.
 - Traduce las instrucciones a 0s y 1s.

Ciclo de Vida de un Programa

Programación

- Programación: Escribir un algoritmo en lenguaje ensamblador (o assembly). Esto es el código fuente.
- Ensamblado: Un programa llamado Ensamblador toma el código fuente y lo traduce a un código máquina.
 - Resuelve las directivas dirigidas al Ensamblador.
 - 2 Calcula el tamaño de cada instrucciones, y resuelve las etiquetas.
 - 3 Traduce las instrucciones a 0s y 1s.
- Carga: Se le indica al Ensamblador una dirección inicial, y éste copia el código máquina en la Memoria desde esa posición en adelante. Luego, carga esa misma dirección en el PC (Program Counter).

Programación

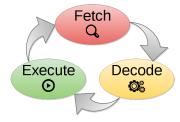
- Programación: Escribir un algoritmo en lenguaje ensamblador (o assembly). Esto es el código fuente.
- Ensamblado: Un programa llamado Ensamblador toma el código fuente y lo traduce a un código máquina.
 - Resuelve las directivas dirigidas al **Ensamblador**.
 - Qualcula el tamaño de cada instrucciones, y resuelve las etiquetas.
 - 3 Traduce las instrucciones a 0s y 1s.
- Carga: Se le indica al Ensamblador una dirección inicial, y éste copia el código máquina en la Memoria desde esa posición en adelante. Luego, carga esa misma dirección en el PC (Program Counter).
- Ejecución: El CPU da inicio a su ciclo de ejecución comenzando por la posición indicada en el PC.

 Programación
 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre ooo

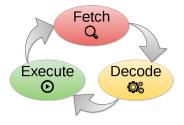
 o ●
 0000000
 0000
 0000
 0000

Ciclo de Ejecución

Ciclo de Ejecución



Ciclo de Ejecución

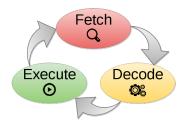


Fetch La UC (Unidad de Control) obtiene una instrucción de la posición de Memoria a la que apunta el PC y lo incrementa (Si es necesario:

busca más palabras de la instrucción usando el **PC** e incrementándolo cada vez.)

vez.)

Ciclo de Ejecución

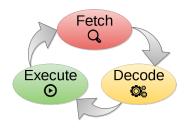


Fetch La UC (Unidad de Control) obtiene una instrucción de la posición de Memoria a la que apunta el PC y lo incrementa (Si es necesario: busca más palabras de la instrucción usando el PC e incrementándolo cada

Decode La **UC** decodifica la instrucción.

Ciclo de Ejecución

Ciclo de Ejecución



Fetch La UC (Unidad de Control)

obtiene una instrucción de la posición de **Memoria** a la que apunta el **PC**

y lo incrementa (Si es necesario: busca más palabras de la instrucción usando el PC e incrementándolo cada vez.)

Decode La UC decodifica la instrucción.

Execute La **UC** ejecuta la instrucción.

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

Tenemos que:

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

Tenemos que:

 ver cuántas palabras necesita cada instrucción

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

Tenemos que:

- ver cuántas palabras necesita cada instrucción
- calcular los valores de las etiquetas

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[et1]] main:

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0x0004 et2:

Tenemos que:

- ver cuántas palabras necesita cada instrucción
- calcular los valores de las etiquetas

DW (Define Word): directiva al ensamblador que provoca que en la posición de memoria que le corresponde, aparezca el valor indicado.

> dirección ocupa

MOV R1,[[et1]] main:

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0x0004 et2:

dirección ocupa

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección ocupa

0×0000

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección ocupa

0x0000 dos palabras

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección ocupa

 0×0000 dos palabras

 0×0002

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección ocupa

0x0000 dos palabras

0x0002 dos palabras

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección ocupa

0x0000 dos palabras

0x0002 dos palabras

0x0004

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

 0×0004 dos palabras

 0×0006

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección ocupa

0x0000 dos palabras

0x0002 dos palabras

0x0004 dos palabras

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra

main:	MOV	R1.[[et1]]
mann.	IVIOV	1 \ _ ,	10411

$$0x0000$$
 dos palabras

dos palabras

0×0002

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

main:	MOV	R1.ſ	[et1]]
mum.	1 V 1 O V	., +, [الحبباا

ADD [R1],0x6000

JMP main

et1: DW 0x0007

et2: DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

dirección

0×0006

 0×0007

ocupa

una palabra

una palabra

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

main:	MOV R1,[[et1]]	0×0000	dos palabras
	ADD [R1],0x6000	0×0002	dos palabras
	JMP main	0×0004	dos palabras
	Į.		

La etiqueta main corresponde a la dirección de memoria 0x0000. La etiqueta et1 corresponde a la dirección de memoria 0x0006. La etiqueta et2 corresponde a la dirección de memoria 0x0007.

Reemplazando las etiquetas por valores de las direcciones estamos listos para codificar.

et1:

et2:

DW 0x0007

DW 0x0004

MOV R1,[[0×0006]]
ADD [R1],0×6000
JMP 0x0000
DW 0×0007
DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0x0006]]
ADD [R1],0×6000
JMP 0x0000
DW 0×0007
DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0×0006]]
ADD [R1],0×6000
JMP 0×0000
DW 0x0007
DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0×0006]]
ADD [R1],0×6000
JMP 0x0000
DW 0×0007
DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006						



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0×0006]]
ADD [R1],0x6000
JMP 0x0000
DW 0×0007
DW 0×0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006						



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0×0006]]
ADD [R1],0×6000
JMP 0x0000
DW 0×0007
DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000				



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0x0006]]
ADD [R1],0x6000
JMP 0x0000
DW 0×0007
DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000				



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0×0006]]
ADD [R1],0x6000
JMP 0×0000
DW 0×0007
DW 0×0004
DW 0×0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000		



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0x0006]]
ADD [R1],0×6000
JMP 0x0000
DW 0×0007
DW 0x0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000		



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

ADD [R1],0x6000 JMP 0x0000 DW 0x0007	MOV	R1,[[0×0006]]
	ADD	[R1],0×6000
DW 0×0007	JMP	0×0000
	DW 0	×0007
DW 0×0004	DW 0	×0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

MOV R1,[[0×0006]]
ADD [R1],0×6000
JMP 0×0000
DW 0×0007
DW 0×0004

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	



	dirección	ocupa
MOV R1,[[0x0006]]	0×0000	dos palabras
ADD [R1],0x6000	0×0002	dos palabras
JMP 0×0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

00	00	0001	0002	0003	0004	0005	0006	0007
18	58	0006	2C40	6000	A000	0000	0007	0004

Y también cargamos la dirección inicial en el PC:



Codifiquemos el siguiente programa y carguémoslo desde la posición de memoria 0x0000:

	dirección	ocupa
MOV R1,[[0×000	6]] 0×0000	dos palabras
ADD [R1],0x6000	0×0002	dos palabras
JMP 0×0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000							
1858	0006	2C40	6000	A000	0000	0007	0004



Ejercicio 1

0000000

Empezamos a hacer el ciclo **Fetch-Decode-Execute** con el **PC** = 0x0000.

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004

Valor	es F	PC	SP	R0	R1	L F	R2	R3	R4	R5	R6	R7		Z	С	V	Ν
inicia	les 0	0000	FFEF	00	00 00	000 0	000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1 0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00	\rightarrow	+E 0000	+F

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

Valor	es	PC	SP		R0	R1	L	R2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les [0000	FFEF		000	00 00	00	0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2		+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D)	+E	+F
0000	1858	3 0006	2C4) (5000	A000	0000	0007	0004	0000	0000	0000	0000	0000	000	0	0000	0000
0010	0000	0000	0000) (0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0	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

Valor	es	PC	SP		R0	R1	L	R2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les [0000	FFEF		000	00 00	00	0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2		+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D)	+E	+F
0000	1858	3 0006	2C4) (5000	A000	0000	0007	0004	0000	0000	0000	0000	0000	000	0	0000	0000
0010	0000	0000	0000) (0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0	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

Valor	es	PC	SP		R0	R1	L	R2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les [0000	FFEF		000	00 00	00	0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2		+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D)	+E	+F
0000	1858	3 0006	2C4) (5000	A000	0000	0007	0004	0000	0000	0000	0000	0000	000	0	0000	0000
0010	0000	0000	0000) (0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0	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

Valor	es F	PC	SP	R0	R1	L F	2	R3	R4	R5	R6	R7		Z	C	V	N
inicia	les C	0000	FFEF	00	00 00	00 0	000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	±1	+2	⊥ 3	+4	+5	⊥ 6	۱7	+8	+9	+A	+B	10		n	1 E	1 -
			1 -		77	73		T1	T-0	T9	\top	\top D	T-C	+1		TL	T1
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	000	00	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC		struc codif		
1	0000	FFEF	0000	1858	0001 1000 0101 1000									
	Ejecución										Flags			\Box
2													•	
	Ejecución										Flags			\Box
3													•	
	Ejecución										Flags			

Valor	es F	PC	SP	R) R:	1 F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les C	0000	FFEF	00	00 00	000 0	0000	0000	0000	0000	0000	000	0	0	0	0	0
				_													
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	1858	0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00	\rightarrow	+E	+F

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

Valor	es F	PC	SP	RO	R	L F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 0	0000	FFEF	00	00 00	000 0	000	0000	0000	0000	0000	000	0	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	00	\rightarrow	+E 0000	+F

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

Valor	es F	PC	SP	R0	R:	L F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 0	0000	FFEF	00	00 00	000 0	000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1 0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8	+9 0000	+A 0000	+B 0000	+C	+ 00	D 00	+E 0000	+F

	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

Valor	es F	PC	SP	RO	R	L F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 0	0000	FFEF	00	00 00	000 0	000	0000	0000	0000	0000	000	0	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	00	\rightarrow	+E 0000	+F

	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											
	Ejecución										Flags
3											
	Ejecución										Flags

Valor	es F	PC	SP	R0	R	L F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 0	0000	FFEF	00	00 00	000 0	000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1 0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8	+9 0000	+A 0000	+B 0000	+C	+ 00	D 00	+E 0000	+F

	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										
	Ejecución										Flags
3											
	Ejecución										Flags

Valor	es F	PC	SP	R) R	1	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 🛚 (0000	FFEF	00	00 0	000	0000	0000	0000	0000	0000	000	00	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	00	00	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	nn	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

Valor	es F	PC	SP	R0	R1	L F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 🛚 (0000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	0	0	0	0	0
	+0	1 1	1.2	1.3													
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	1858	0006	2C40	6000	+4 A000	0000	+6 0007	+7 0004	0000	+9 0000	+A 0000	+B 0000	+C	00	_	+E	+F 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	0010 1100 0100 0000						
	Ejecución										Flags
3											
	Ejecución										Flags

Valor	es F	PC	SP	R0	R	L I	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les (0000	FFEF	00	00 00	000	0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1 0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00	D 00	+E 0000	+F

	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	0010 1100 0100 0000	0003					
	Ejecución										Flags
3											
	Ejecución										Flags

Valor	es P	C	SP	R0	R1	L F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 0	000	FFEF	00	00 00	00 0	000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1 0006	+2 2C40	+3 6000	+4 A000	+5 0000	+6 0007	+7 0004	+8	+9 0000	+A 0000	+B 0000	+C	+ 00	D 00	+E 0000	+F

	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	0010 1100 0100 0000	0003	6000	0004			
	Ejecución										Flags
3											
	Ejecución										Flags

Valor	es F	PC	SP	RO) R:	L F	2	R3	R4	R5	R6	R7		Ζ	C	V	N
inicia	les 🛚 (0000	FFEF	00	00 00	000 0	000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	±F	⊢F
								1	10	1 -	1 , ,	, ,	1 0				
0000	1858	0006	2C40	6000	A000	0000	0007	0004	0000	0000	0000	0000	0000	00	\rightarrow	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	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución										Flags
3											
	Ejecución										Flags

Valor		C	SP	RO			R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 0	0000	FFEF	00	00 00	000 0	0000	0000	0000	0000	0000	000	0	0	0	0	0
	+0	+ 1	1.2	1.2			_							_			
	70	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	1858	0006	2C40	6000	+4 A000	+5 0000	+6	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	00	\rightarrow	+E 0000	+F

	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	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×000	4]=0×A00	$0+0\times600$	00=0×0000						Flags
3											
	Ejecución										Flags

Valor	es l	PC	SP	R0	R:	1 F	R2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les (0000	FFEF	00	00 00	000 0	0000	0000	0000	0000	0000	000	0	0	0	0	0
	. 0																
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1	+2 2C40	+3	+4 0000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00	D 00	+E 0000	+F

	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	F		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											
	Ejecución										Flags

Valore	es F	PC	SP	R0	R:	1 F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicial	les C	0000	FFEF	00	00 00	000 0	0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1	+2 2C40	+3 6000	+4 0000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00		+E 0000	+F

	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	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										
	Ejecución										Flags

Valore	es F	PC	SP	R0	R:	1 F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicial	les C	0000	FFEF	00	00 00	000 0	0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1	+2 2C40	+3 6000	+4 0000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00		+E 0000	+F

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instrucción decodificada
							palabra		palabra		
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	$0+0\times600$	00=0×0000						Flags 1 1 0 0
3	0004			0000							
	Ejecución										Flags

Valor	es F	PC	SP	RO	R	1 F	R2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les (0000	FFEF	00	00 00	000 0	000	0000	0000	0000	0000	000	0	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	000	_	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	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0x0004	4]=0×A00	0+0×600	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000						
	Ejecución										Flags

Valor	es F	PC	SP	R0	R:	1 F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les C	0000	FFEF	00	00 00	000 0	0000	0000	0000	0000	0000	000	0	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1	+2 2C40	+3 6000	+4 0000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+	D 00	+E 0000	+F

	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	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×000	4]=0×A00	$0+0\times600$	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000						
	Ejecución										Flags

Valor	es F	PC	SP	R0	R:	1 I	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les (0000	FFEF	000	00 00	000	0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1	+2 2C40	+3 6000	+4 0000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00	D 00	+E 0000	+F 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, [[0×0006]]
	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	$0+0\times600$	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000	0005					
	Ejecución										Flags

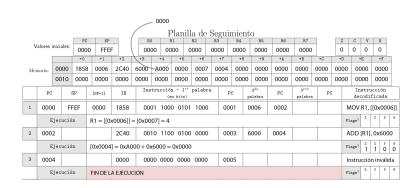
Valor	es	PC	SP	R0	R:	1 I	R2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les (0000	FFEF	000	00 00	000 (0000	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	1858	0006	2C40	6000	0000	0000	0007	0004	0000	0000	0000	0000	0000	000	00	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	nn	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	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×000	4]=0×A00	0+0×600	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000	0005					Instrucción Inválida
	Ejecución										Flags

Valor	es F	PC	SP	R0	R:	1 F	R2	R3	R4	R5	R6	R7		Z	С	V	N
inicia	les 0	0000	FFEF	000	00 00	000	0000	0000	0000	0000	0000	000	0	0	0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	+0 1858	+1	+2 2C40	+3 6000	+4 0000	+5 0000	+6 0007	+7 0004	+8 0000	+9 0000	+A 0000	+B 0000	+C	+ 00	_	+E	+F

	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	0010 1100 0100 0000	0003	6000	0004			ADD [R1],0x6000
	Ejecución	[0×000	4]=0×A00	$0+0\times600$	00=0×0000						Flags 1 1 0 0
3	0004			0000	0000 0000 0000 0000	0005					Instrucción Inválida
	Ejecución	FIN DE	LA EJE	CUCIÓN							Flags

Ejercicio 1 - Solución



ramación **Ejercicio 1** Ejercicio 2 Ejercicio 3 Cierro 000**0000** 0000 00000 0000

Ejercicio 1 - Seguimiento

Ejercicio 1 - Observaciones

¿Qué pasó con el programa anterior?

amación **Ejercicio 1** Ejercicio 2 Ejercicio 3 Cierro 000**0000 0**000 0000 0000

Ejercicio 1 - Seguimiento

Ejercicio 1 - Observaciones

¿Qué pasó con el programa anterior?

En la ejecución del programa cambió una posición de memoria donde estaba cargado... **el mismo programa!**

ación **Ejercicio 1** Ejercicio 2 Ejercicio 3 Cierre
000**000●** 0000 00000 0000

Ejercicio 1 - Seguimiento

Ejercicio 1 - Observaciones

¿Qué pasó con el programa anterior?

En la ejecución del programa cambió una posición de memoria donde estaba cargado... **el mismo programa!**

Eso hizo que nuestro programa se modificara, haciendo que la máquina llegase a una instrucción inválida.

 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre

 ○○○○○○
 ○○○○○
 ○○○○○
 ○○○○○

Ejercicio 1 - Seguimiento

Ejercicio 1 - Observaciones

¿Qué pasó con el programa anterior?

En la ejecución del programa cambió una posición de memoria donde estaba cargado... **el mismo programa!**

Eso hizo que nuestro programa se modificara, haciendo que la máquina llegase a una instrucción inválida.

Moraleja:



Ejercicio 2

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

Ejercicio 2 - Seguimiento

Valor	es F	PC	SP		R0	R1	F	22	R3	R4	R5	R6	R7		Ζ	C	V	N
inicia	les (0000	FFEF		0000	000	0 00	000	0000	0000	0000	0000	000	0	0	0	0	0
	+0	+1	+2	+	3 -	+4	+5	+6	+7	+8	+9	+A	+B	+C	+[)	+E	+F
0000	19CB	0004	B000	000	05 OF	FE0	39A7	C000	0000	0000	0000	0000	0000	0000	000	00	0000	0000
0010	0000	0000	0000	000	00 00	000	0000	0000	0000	0000	0000	0000	0000	0000	000	00	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
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es	PC	S	P		R0	R1		R2	F	83	R4	R5	R6	R7		Ζ	C	V	N
inicia	les [0000	F	FEF	[000	0 00	00	0000	C	0000	0000	0000	0000	000	00	0	0	0	0
	+0	+1	.	+2		+3	+4	+5	+	6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
0000	19CF	3 000	4	B000	0	005	0FE0	39A	7 C0	00	0000	0000	0000	0000	0000	0000	00	00	0000	0000
0010	0000	000	0	0000	0	0000	0000	0000	00	00	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										
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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								
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z	V	N
inicia	les 00	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	0 0	0	0
	. 0	. 1					1		. 0			. D				
	+0	+1	+2	+3	+4	+5	+0	+1	+8	+9	+A	+6	+c	+0	+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							
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	res F	C	SP		R0	R1	F	22	R3	R4	R5	R6	R7		Z (2 V	N	
inicia	les (0000	FFEF		000	0 00	00 0	000	0000	0000	0000	0000	000	00	0 (0	0	
	+0	+1	+2		+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+1	-	+F
0000	19CB	0004	B000	C	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	000	0 00	000
0010	0000	0000	0000	C	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0 00	000

	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						
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 00	000	0000	0000	0000	0000	000	0	0 0	0	0
	+0	±1	+2	±3	+4	+5	+6	⊥ 7	+8	+9	+Δ	±R	+0	+D	⊥F	_+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					
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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			
	Ejecución										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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										Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C S	SP	R0	R1	R	2 I	3	R4	R5	R6	R7		Z (. V	N
inicia	les 00	000	FEF	000	00 00	00 0	000	0000	0000	0000	0000	000	0	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×	oFE0							Flags
2											
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z	V	N
inicia	les 00	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	0 0	0	0
	. 0	. 1					1		. 0			. D				
	+0	+1	+2	+3	+4	+5	+0	+1	+8	+9	+A	+6	+c	+0	+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>	OFE0							Flags
2	0002										
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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							
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z	V	N
inicia	les 00	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	0 0	0	0
	. 0	. 1					1		. 0			. D				
	+0	+1	+2	+3	+4	+5	+0	+1	+8	+9	+A	+6	+c	+0	+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×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000						
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z	V	N
inicia	les 00	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	0 0	0	0
	. 0	. 1					1		. 0			. D				
	+0	+1	+2	+3	+4	+5	+0	+1	+8	+9	+A	+6	+c	+0	+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>	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003					
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 00	000	0000	0000	0000	0000	000	0	0 0	0	0
	+0	±1	+2	±3	+4	+5	+6	⊥ 7	+8	+9	+Δ	±R	+0	+D	⊥F	_+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>	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			
	Ejecución										Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 00	000	0000	0000	0000	0000	000	0	0 0	0	0
	+0	±1	+2	±3	+4	+5	+6	⊥ 7	+8	+9	+Δ	±R	+0	+D	⊥F	_+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×	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

Ejercicio 2 - Seguimiento

Valor	es P	C S	SP	R0	R1	. R	2 I	3	R4	R5	R6	R7		Z C	V	N
inicia	les 00	1 00C	FEF	000	00 00	00 00	000	0000	0000	0000	0000	000	0	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>	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	$F] = 0 \times 000$	4 SP=0:	×FFEE PC=0×0005						Flags
3											
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C S	SP	R0	R1	. R	2 I	3	R4	R5	R6	R7		Z C	V	N
inicia	les 00	1 00C	FEF	000	00 00	00 00	000	0000	0000	0000	0000	000	0	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
	1.0	51	[31 +1]	"\	mstruccion - 1er parabra	' C	palabra	' C	palabra	' C	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]=0x000	4 SP=0:	FFEE PC=0x0005						Flags
3	0005										
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C S	SP	R0	R1	. R	2 I	3	R4	R5	R6	R7		Z C	V	N
inicia	les 00	1 00C	FEF	000	00 00	00 00	000	0000	0000	0000	0000	000	0	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>	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	×FFEE PC=0×0005						Flags
3	0005	FFEE									
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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 0x0005
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	×FFEE PC=0×0005						Flags
3	0005	FFEE	0004	39A7							
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	×FFEE PC=0×0005						Flags
3	0005	FFEE	0004	39A7	0011 1001 1010 0111						
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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>	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×0005
	Ejecución	[0xFFE	F]=0×000	4 SP=0	xFFEE PC=0x0005						Flags
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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>	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	×FFEE PC=0×0005						Flags
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, R7
	Ejecución										Flags
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	000	00 00	00 0	000	0000	0000	0000	0000	000	00	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	4 SP=0:	FFEE 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
4											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	F	22	R3	R4	R5	R6	R7		Z C	V	N
inicia	les C	0000	FFEF		000	0 00	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	T	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	4 B000	1	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×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	FFEE 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											
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	F	22	R3	R4	R5	R6	R7		Z C	V	N
inicia	les C	0000	FFEF		000	0 00	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	T	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	4 B000	1	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 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										
	Ejecución										Flags
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF		0000	000	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	1 +	-3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	00	005 (0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	00	000	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.	[51 1]		mistracción ici palabra	' ~	palabra		palabra	' -	decodi	ficad	a
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	0×00	04]
	Ejecución	R7=[0:	×0004]=0×	0FE0						Flags CALL 0x0005			
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x00	005	
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	FFEE PC=0x0005						Flags		\Box
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		\Box
5													
	Ejecución										Flags		

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF		0000	000	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	1 +	-3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	00	005 (0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	00	000	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×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×0005
	Ejecución	[0xFFE	F]=0×000	4 SP=0:	FFEE 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

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	F	22	R3	R4	R5	R6	R7		Z C	V	N
inicia	les C	0000	FFEF		000	0 00	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	T	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	4 B000	1	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×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	FFEE 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

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF		0000	000	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	1 +	-3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	00	005 (0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	00	000	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	n	
	1.0	31	[31 +1]	"``	mstruccion - 1er parabra	' C	palabra	1 0	palabra	1.0	decod	ificac	la
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	[0x0	004]
	Ejecución	R7=[0:	×0004]=0>	OFE0							Flags		
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0	005	•
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	FFEE PC=0x0005						Flags		
3	0005	FFEE	0004	39A7	0011 1001 1010 0111	0006					SUB R6, I	₹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										Flags		
5													
	Ejecución										Flags		

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	F	22	R3	R4	R5	R6	R7		Z C	V	N
inicia	les C	0000	FFEF		000	0 00	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	T	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	4 B000	1	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×	OFE0							Flags
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0005
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	FFEE PC=0x0005	Flags					
3	0005	FFEE	0004	39A7	7 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	F]=0x0004 SP=0xFFEF						Flags	
5											
	Ejecución										Flags

Ejercicio 2 - Seguimiento

Valor	es P	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z (. V	N
inicia	les 0	000	FFEF		0000	000	00 00	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	+	3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	000	05 0	FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	000	00 0	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		stru codi			7
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV I	R7,	0x0	004]	╗
	Ejecución	R7=[0	×0004]=0×	0FE0							Flags				П
2	0002			B000						CALL	0x00	05			
	Ejecución	[0×FFE	F]=0×000	4 SP=0:						Flags				٦	
3	0005	FFEE	0004	39A7						SUB R	6, R	7		٦	
	Ejecución	R6=0x	0000-0×0F	E0=0×F)20						Flags	0	1	0	1
4	0006			C000	1100 0000 0000 0000	0007					RET				
	Ejecución	PC=[0:	xFFEF]=0	×0004 S	4 SP=0xFFEF					Flags				٦	
5	0004														\neg
	Ejecución										Flags				٦
	Ljecacion										065			_	_

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF		0000	000	0 00	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	+	-3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	00	05 (0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	00	00 (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 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		Ť
2	0002			B000	1011 0000 0000 0000							005	
	Ejecución	[0×FFE	F]=0x000	4 SP=0:	xFFEE PC=0x0005								\Box
3	0005	FFEE	0004	39A7	0011 1001 1010 0111 0006 S						SUB R6, F	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	xFFEF]=0	×0004 S	P=0xFFEF					Flags		\Box	
5	0004	FFEF	0000										
	Ejecución				•						Flags		\Box

Ejercicio 2 - Seguimiento

Valor	es P	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z (. V	N
inicia	les 0	000	FFEF		0000	000	00 00	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	+	3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	000	05 0	FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	000	00 0	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	In	stru	cció	n	
	1.0	J 1	[31 +1]	111	Ilistruccion - Ter parabra	' C	palabra	1 0	palabra	1 C	de	codi	ficac	la	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV	R7, [0x0	004]	٦
	Ejecución	R7=[0	<0004]=0×	0FE0							Flags		П	\Box	٦
2	0002			B000							CALL	0×00	05		٦
	Ejecución	[0×FFE	F]=0x000	4 SP=0						Flags			\Box	٦	
3	0005	FFEE	0004	39A7	0011 1001 1010 0111 0006 St						SUB R	6, R	7		٦
	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	PC=[0:	xFFEF]=0	×0004 S	0004 SP=0xFFEF						Flags		П	\Box	٦
5	0004	FFEF	0000	0FE0											٦
	Ejecución										Flags				٦
															_

Ejercicio 2 - Seguimiento

Valor	es P	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z (. V	N
inicia	les 0	000	FFEF		0000	000	00 00	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	+	3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	000	05 0	FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	000	00 0	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	[0xFFE	F]=0x000	4 SP=0:							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	SP=0xFFEF					Flags	
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000						
	Ejecución										Flags

 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre

 0000000
 0●00
 00000
 0000

Ejercicio 2 - Seguimiento

Ejercicio 2 - Seguimiento

Valor	es P	C	SP	RO	R	L F	R2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	00	00 00	000 (0000	0000	0000	0000	F020	000	00	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.	In	stru	cció	n	٦
	FC	3F	[21-1]	IIX	mstruccion - 1er palabra	I C	palabra	FC	palabra	r C	decodificada				
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004		004]	٦	
	Ejecución	R7=[0:	R7=[0x0004]=0x0FE0								Flags		П	П	٦
2	0002					CALL	0x00	05		\neg					
	Ejecución	[0×FFE	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005								Flags			П	٦
3	0005	FFEE 0004 39A7 0011 1001 1010 0111 0006					SUB R	6, R	7		\neg				
	Ejecución	R6=0x	0000-0×0F	E0=0×F	020						Flags	0	1	0	ī
4	0006			C000	1100 0000 0000 0000	0007					RET				\neg
	Ejecución	PC=[0xFFEF]=0x0004 SP=0xFFEF						Flags				٦			
5	0004	FFEF	0000	0FE0	0000 1111 1110 0000	0005									\neg
	Ejecución										Flags			\top	٦

 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre

 0000000
 0●00
 00000
 0000

Ejercicio 2 - Seguimiento

Ejercicio 2 - Seguimiento

Valor	es F	C	SP		R0	R1	R	2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF		0000	000	00 0	000	0000	0000	0000	F020	000	00	0 0	0	0
	+0	+1	+2	1 +	-3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	19CB	0004	B000	00	005 (0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

	PC	SP	[SP+1] IR I		Instrucción - 1er palabra	PC.	2da	PC.	3ra	PC.	Instrucción
		51	[31 +1]	IIX	mstruccion - 1er parabra	' `	palabra	' C	palabra	1 0	decodificada
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0x0004]
	Ejecución	R7=[0:	R7=[0x0004]=0x0FE0								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=[0:	xFFEF]=0	×0004 S	P=0×FFEF						Flags
5	0004	FFEF 0000 0FE0 0000 1111 1110 0000 0005 Instrucción Inválida						Instrucción Inválida			
	Ejecución										Flags

Ejercicio 2 - Seguimiento

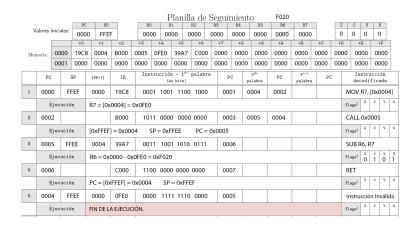
Ejercicio 2 - Seguimiento

Valor	es P	C	SP	RO	R	L F	R2	R3	R4	R5	R6	R7		Z C	V	N
inicia	les 0	000	FFEF	00	00 00	000 (0000	0000	0000	0000	F020	000	00	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	Instr	ucció difica	
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7,	[0x0	004]
	Ejecución	R7=[0	R7=[0x0004]=0x0FE0 Flags								Flags		
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0x0	0005	•
	Ejecución	[0×FFE	[0xFFEF]=0x0004 SP=0xFFEE PC=0x0005								Flags		
3	0005	FFEE 0004 39A7 0011 1001 1010 0111 0006 S							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 Flag							Flags				
5	0004	FFEF 0000 0FE0 0000 1111 1110 0000 0005 Instru						Instrucció	n Inv	álida			
	Ejecución	FIN DE	FIN DE LA EJECUCIÓN								Flags		

Eiercicio 2 - Seguimiento

Ejercicio 2 - Solución



Ejercicio 2 - Seguimiento

Ejercicio 2 - Observaciones

¿Qué pasó con el programa anterior?

Ejercicio 2 - Observaciones

¿Qué pasó con el programa anterior?

- Utilizamos la instrucción CALL.
- El valor del PC de retorno se almacena en la pila durante de la ejecución de la instrucción.
- Recuerden que la pila crece en el sentido de las direcciones de memoria menores.
- Al regresar con RET, se "saca" la dirección de retorno de allí.

Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

> main: MOV R1, 0x0000

loop: CMP R1, 0×0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: **RET**

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RET

dirección ocupa

0×0000

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

0x0000 dos palabras

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

0x0002

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

 $\sf JL\ sumoUno$

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

 0×0002 dos palabras

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

0×0004

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

 0×0002 dos palabras

0x0004 una palabra

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

 0×0002 dos palabras

0x0004 una palabra

0×0005

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

 0×0004 una palabra

 0×0005 una palabra

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

0x0004 una palabra

0x0005 una palabra

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

JGE fin

sumoUno: ADD R1, 0x0001

JNE loop

fin: RFT

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

0x0004 una palabra

0x0005 una palabra

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JGE fin

JL sumoUno

ADD R1, 0x0001 JNE loop

fin: RFT

dirección ocupa

0x0000 dos palabras

0x0002 dos palabras

0x0004 una palabra

0x0005 una palabra

0x0006 dos palabras

8000x0

sumoUno:

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main: MOV R1, 0x0000

loop: CMP R1, 0x0001

JL sumoUno

sumoUno: ADD R1, 0x0001

JGE fin

JNE loop

fin: RET

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	una palabra
0×0005	una palabra
0×0006	dos palabras
0×0008	una palabra

Codifiquemos el siguiente código y carguémoslo desde la posición de

main:	MOV R1, 0x0000
loop:	CMP R1, 0x0001
	JL sumoUno
	JGE fin
sumoUno:	ADD R1, 0x0001
	JNE loop
fin:	RET

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	una palabra
0×0005	una palabra
0×0006	dos palabras
0×0008	una palabra
0×0009	

Ejercicio 3

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

main:	MOV R1, 0×0000
loop:	CMP R1, 0x0001
	JL sumoUno
	JGE fin
sumoUno:	ADD R1, 0x0001
	JNE loop
fin:	RET

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	una palabra
0×0005	una palabra
0×0006	dos palabras
8000×0	una palabra
0×0009	una palabra

Codifiquemos el siguiente código y carguémoslo desde la posición de memoria 0x0000:

		dirección	ocupa
main:	MOV R1, 0×0000	0×0000	dos palabras
loop:	CMP R1, 0x0001	0×0002	dos palabras
	JL sumoUno	0×0004	una palabra
	JGE fin	0×0005	una palabra
sumoUno:	ADD R1, 0x0001	0×0006	dos palabras
	JNE loop	0×0008	una palabra
fin:	RET	0×0009	una palabra

Buenísimo, tenemos las direcciones de las etiquetas, pero... *los saltos condicionales son* **relativos**.

Buenísimo, tenemos las direcciones de las etiquetas, pero... *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. Buenísimo, tenemos las direcciones de las etiquetas, pero... *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.
- Vamos a tener que calcular estos desplazamientos.

0×0000: MOV R1, 0×0000

0×0002: CMP R1, 0×0001

0x0004: JL sumoUno

0x0005: JGE fin

0×0006: ADD R1, 0×0001

0x0008: JNE loop

0x0009: RET

 $\mathsf{Despl} = \mathsf{Dir} \mathsf{_Etiqueta} - Dir \mathsf{_Instr}_{postfetch}$

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL sumoUno

0x0005: JGE fin

0×0006: ADD R1, 0×0001

0x0008: JNE loop

0x0009: RET

 $\mathsf{Despl} = \mathsf{Dir} _ \mathsf{Etiqueta} - Dir _ Instr_{postfetch}$

desplazamiento sumoUno:

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL sumoUno

0x0005: JGE fin

0×0006: ADD R1, 0×0001

0x0008: JNE loop

0x0009: RET

 $\mathsf{Despl} = \mathsf{Dir} \ldotp \mathsf{Etiqueta} \cdot Dir \ldotp \mathit{Instr}_{postfetch}$

desplazamiento sumoUno:

0×0000: MOV R1, 0×0000

0×0002: CMP R1, 0×0001

0x0004: JL sumoUno

0x0005: JGE fin

0×0006: ADD R1, 0×0001

0x0008: JNE loop

0x0009: RET

 $\mathsf{Despl} = \mathsf{Dir_Etiqueta} - Dir_Instr_{postfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005$

0×0000: MOV R1, 0×0000

0×0002: CMP R1, 0×0001

0x0004: JL sumoUno

0x0005: JGE fin

0×0006: ADD R1, 0×0001

0x0008: JNE loop

0x0009: RET

 $\mathsf{Despl} = \mathsf{Dir_Etiqueta} - Dir_Instr_{postfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1$

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL sumoUno

0x0005: JGE fin

0×0006: ADD R1, 0×0001

0x0008: JNE loop

0x0009: RET

 $\mathsf{Despl} = \mathsf{Dir} \text{_Etiqueta} \text{-} Dir \text{_} Instr_{postfetch}$

desplazamiento sumoUno: $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

0×0000: MOV R1, 0×0000

0x0002: CMP R1, 0x0001

0x0004: JL sumoUno

0x0005: JGE fin

0×0006: ADD R1, 0×0001

0x0008: JNE loop

0x0009: RET

 $\mathsf{Despl} = \mathsf{Dir_Etiqueta} - Dir_Instr_{postfetch}$

desplazamiento sumoUno: $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0x0001

0×0008: JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno: $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0x0001

0×0008: JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006$

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0x0001

0×0008: JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin: $0 \times 0009 - 0 \times 0006 = 3$

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0x0001

0×0008: JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006 = 3 = 0 \times 03$

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0×0001

 0×0008 : JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006 = 3 = 0 \times 03$

desplazamiento loop:

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0×0001

 0×0008 : JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006 = 3 = 0 \times 03$

desplazamiento loop:

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0×0001

 0×0008 : JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006 = 3 = 0 \times 03$

desplazamiento loop:

 $0 \times 0002 - 0 \times 0009$

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0×0001

 0×0008 : JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006 = 3 = 0 \times 03$

desplazamiento loop: $0 \times 0002 - 0 \times 0009 = -7$

 0×0000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0×0001

 0×0008 : JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006 = 3 = 0 \times 03$

desplazamiento loop:

 $0 \times 0002 - 0 \times 0009 = -7 = 0 \times F9$

 0×00000 : MOV R1, 0×0000

 0×0002 : CMP R1, 0×0001

 0×0004 : JL sumoUno

 0×0005 : JGE fin

 0×0006 : ADD R1. 0x0001

0×0008: JNE loop

 0×0009 : RET $Despl = Dir_Etiqueta - Dir_Instr_{nostfetch}$

desplazamiento sumoUno:

 $0 \times 0006 - 0 \times 0005 = 1 = 0 \times 01$

desplazamiento fin:

 $0 \times 0009 - 0 \times 0006 = 3 = 0 \times 03$

desplazamiento loop:

 $0 \times 0002 - 0 \times 0009 = -7 = 0 \times F9$

0x0000: MOV R1, 0x0000

0x0002: CMP R1, 0x0001

0x0004: JL 0x01

0x0005: JGE 0x03

0x0006: ADD R1, 0x0001

0x0008: JNE 0xF9

0x0009: RET

Ahora sí: nuestro programa esta listo para ser ensamblado!

Tarea: ensamblado y segui-

miento!

Ejercicio 3 - Ensamblado

Ejercicio 3 - Observaciones

Ejercicio 3 - Ensamblado

Ejercicio 3 - Observaciones

Moraleja del ejercicio anterior:

 Los desplazamientos de los saltos condicionales siempre son relativos.
 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre

 0000000
 0000
 0000
 0000

Ejercicio 3 - Observaciones

- Los desplazamientos de los saltos condicionales siempre son relativos.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.

Ejercicio 3 - Observaciones

- Los desplazamientos de los saltos condicionales siempre son relativos.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.
- Para calcular los desplazamientos, tengan en cuenta que el salto se ejecuta después de incrementar el PC.

Ejercicio 3 - Observaciones

- Los desplazamientos de los saltos condicionales siempre son relativos.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.
- Para calcular los desplazamientos, tengan en cuenta que el salto se ejecuta después de incrementar el PC.
- Los desplazamientos se representan en complemento a dos.

Ejercicio 3 - Observaciones

- Los desplazamientos de los saltos condicionales siempre son relativos.
- Siempre van a ser traducidas las etiquetas primero, y luego calculados los desplazamientos.
- Para calcular los desplazamientos, tengan en cuenta que el salto se ejecuta después de incrementar el PC.
- Los desplazamientos se representan en complemento a dos.
- Siempre extender los desplazamientos con signo a la hora de aplicarlos.

Resumen

Hoy vimos:

- Máquina Orga 1 y su arquitectura
 - ISA: formato de instrucción, modos de direccionamiento
 - ciclo de ejecución
- Ciclo de vida de un programa
 - uso de etiquetas
 - directivas
- Ensamblamos programas
- Seguimos programas dentro de la máquina Orga1

¿Preguntas?

¿Preguntas?



 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre 000000

 0000000
 00000
 00000
 00000

Avisos

Martes 8/5: Taller de Ciclo de Instrucción en los labos. IMPORTANTE: Traer Cartilla Orga 1 y planillas de Seguimiento

 Programación
 Ejercicio 1
 Ejercicio 2
 Ejercicio 3
 Cierre 000

 00
 000000
 0000
 0000
 0000

¡Eso es todo amigos!

