ORGA1: Ciclo de Instrucciones

Organización del Computador I

Facundo Pessacg

Departamento de Computación - FCEyN UBA

Segundo Cuatrimestre 2018 - Turno Tarde

Tamaño de memoria

- Tamaño de memoria
- ② Dirección de memoria

- Tamaño de memoria
- 2 Dirección de memoria
- Unidad direccionable

- Tamaño de memoria
- 2 Dirección de memoria
- Unidad direccionable
- Instrucción

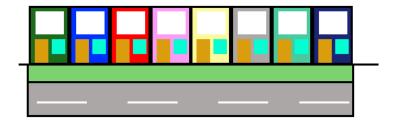
- Tamaño de memoria
- ② Dirección de memoria
- Unidad direccionable
- Instrucción
- Odificación de una instrucción

Bienvenido al Mundo Línea. Acá las personas ocupan 1m de ancho, y nada tiene altura ni profundidad.

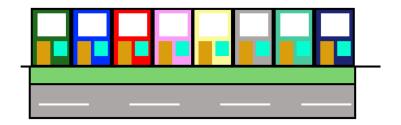
Bienvenido al Mundo Línea. Acá las personas ocupan 1m de ancho, y nada tiene altura ni profundidad. Supongamos una cuadra de unos 32m en este mundo.

Agreguemos casas en los 32 metros de la cuadra sabiendo que cada casa tiene 4m de ancho.

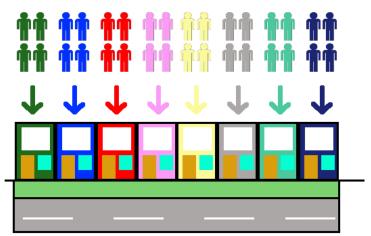
Agreguemos casas en los 32 metros de la cuadra sabiendo que cada casa tiene 4m de ancho.



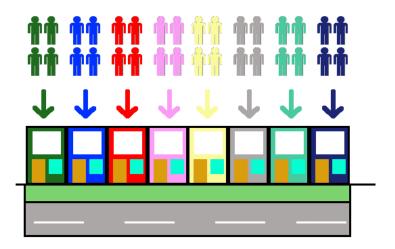
Como una persona ocupa 1m de ancho, y estamos en Mundo Línea donde nada tiene altura ni profundidad, la capacidad de cada casa es de 4 personas. ¿Cuántas personas entran en total en la cuadra?



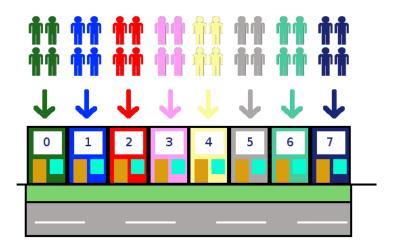
Como una persona ocupa 1m de ancho, y estamos en Mundo Línea donde nada tiene altura ni profundidad, la capacidad de cada casa es de 4 personas. ¿Cuántas personas entran en total en la cuadra?



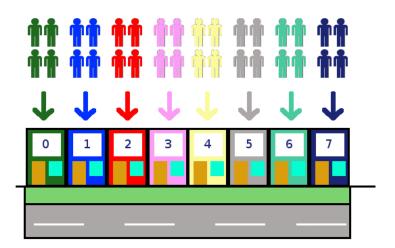
Numeremos las casas.



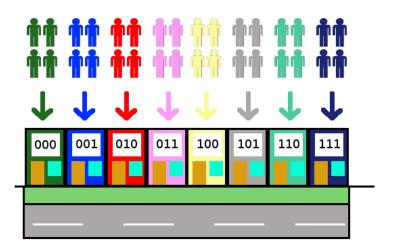
Numeremos las casas.



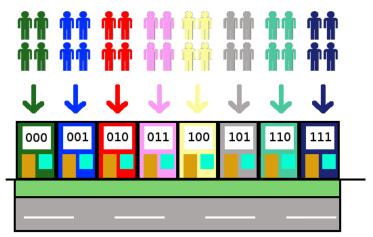
¿Cuántos bits necesitamos para identificar las casas?



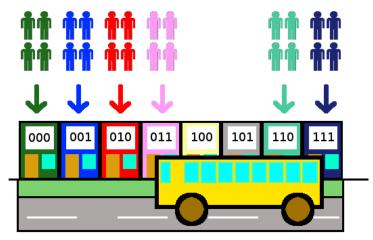
¿Cuántos bits necesitamos para identificar las casas?



Para ir a trabajar hay un colectivo con lugar para 8 personas que solamente arranca cuando esta lleno. Lleva siempre dos casas contiguas completas.

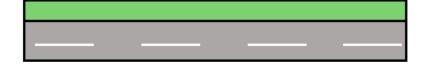


Para ir a trabajar hay un colectivo con lugar para 8 personas que solamente arranca cuando esta lleno. Lleva siempre dos casas contiguas completas.



Supongamos que tenemos otra cuadra de 40m.

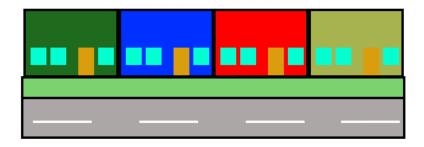
Supongamos que tenemos otra cuadra de 40m.



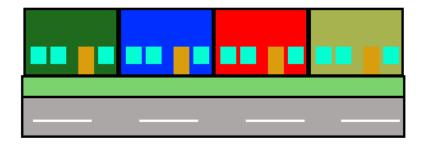
Ahora agreguemos casas a la cuadra de 40m, esta vez de las casas son de 10m de frente.



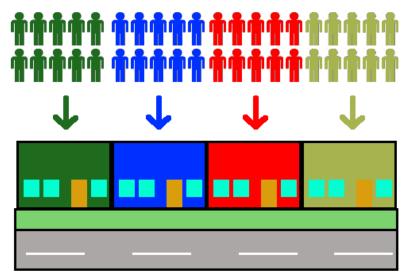
Ahora agreguemos casas a la cuadra de 40m, esta vez de las casas son de 10m de frente.



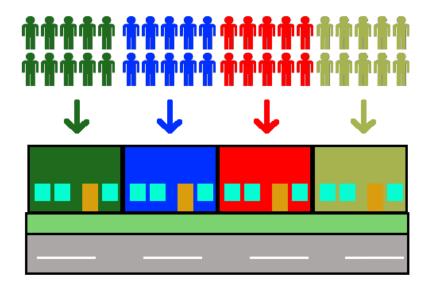
La capacidad de cada casa será de 10 personas. ¿Cuántas personas entran en la cuadra?



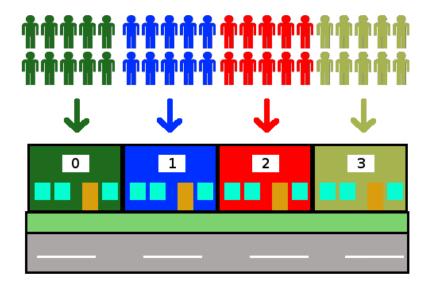
La capacidad de cada casa será de 10 personas. ¿Cuántas personas entran en la cuadra?



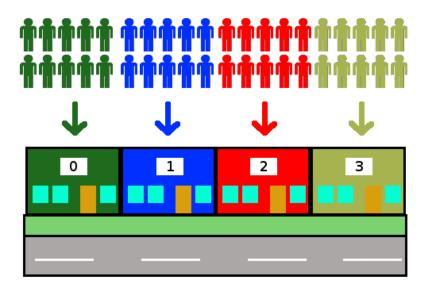
Numeremos las casas.



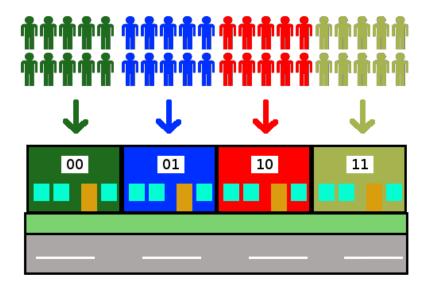
Numeremos las casas.



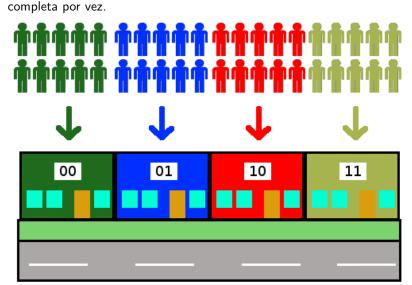
¿Cuántos bits necesitamos para identificar las casas?



¿Cuántos bits necesitamos para identificar las casas?

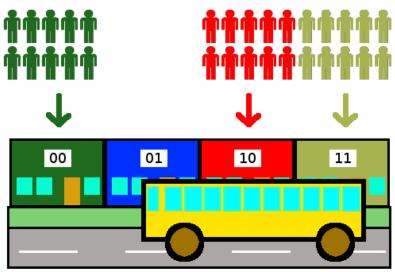


Para ir a trabajar hay un colectivo con lugar para 10 personas que solamente arranca cuando esta lleno. Lleva únicamente a una casa complete para una



¿De qué estamos hablando? (toma 2)

Para ir a trabajar hay un colectivo con lugar para 10 personas que solamente arranca cuando esta lleno. Lleva únicamente a una casa completa por vez.





Analogía entre la cuadra y la memoria.

 La cantidad de personas que entra en la cuadra es el tamaño de la memoria

- La cantidad de personas que entra en la cuadra es el tamaño de la memoria
- La cantidad de personas que entra en una casa es el tamaño de la unidad direccionable

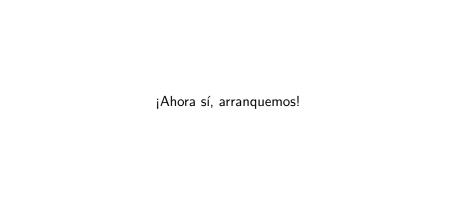
- La cantidad de personas que entra en la cuadra es el tamaño de la memoria
- La cantidad de personas que entra en una casa es el tamaño de la unidad direccionable
- La cantidad de casas de la cuadra es la cantidad de direcciones de la memoria

- La cantidad de personas que entra en la cuadra es el tamaño de la memoria
- La cantidad de personas que entra en una casa es el tamaño de la unidad direccionable
- La cantidad de casas de la cuadra es la cantidad de direcciones de la memoria
- La cantidad de bits necesarios para numerar las casas es la cantidad de bits necesarios para distinguir las direcciones de memoria

¿Qué tenía esto que ver?

Analogía entre la cuadra y la memoria.

- La cantidad de personas que entra en la cuadra es el tamaño de la memoria
- La cantidad de personas que entra en una casa es el tamaño de la unidad direccionable
- La cantidad de casas de la cuadra es la cantidad de direcciones de la memoria
- La cantidad de bits necesarios para numerar las casas es la cantidad de bits necesarios para distinguir las direcciones de memoria
- La capacidad del colectivo (o bus) es el tamaño de la palabra



• ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?

- ¿Cuántos bits son necesarios para poder direccionar la
 - memoria en los siguientes casos?

Memoria de 2 GB con direccionamiento a byte

- ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?
- Memoria de 2 GB con direccionamiento a byte
 - #directiones = $\frac{2 \text{ GB}}{1 \text{ B}/dir} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B}/dir} = 2^1 \times 2^{30} \ dir$

- ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?
- Memoria de 2 GB con direccionamiento a byte
 - #directiones = $\frac{2 \text{ GB}}{1 \text{ B}/dir} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B}/dir} = 2^1 \times 2^{30} \text{ dir}$
 - #bits de direccionamiento = $\log_2(2^1 \times 2^{30}) = 31$ bits

- ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?
 - 1 Memoria de 2 GB con direccionamiento a byte
 - - #directiones = $\frac{2 \text{ GB}}{1 \text{ B}/dir} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B}/dir} = 2^1 \times 2^{30} \ dir$
 - #bits de direccionamiento = $\log_2(2^1 \times 2^{30}) = 31 \ bits$
 - Memoria de 512 MB con direccionamiento a 16 bits

- ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?
 - Memoria de 2 GB con direccionamiento a byte
 - #directiones = $\frac{2 \text{ GB}}{1 \text{ B}/dir} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B}/dir} = 2^1 \times 2^{30} \ dir$
 - #bits de direccionamiento = $\log_2(2^1 \times 2^{30}) = 31 \ bits$
 - Memoria de 512 MB con direccionamiento a 16 *bits* #direcciones = $\frac{512 \text{ MB}}{2 \text{ B}/dir} = \frac{2^9 \times 2^{20} \text{B}}{2 \text{ B}/dir} = 2^{28} dir$

- ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?
 - Memoria de 2 GB con direccionamiento a byte
 - #directiones = $\frac{2 \text{ GB}}{1 \text{ B}/dir} = \frac{2 \times 2^{30} \text{B}}{1 \text{ B}/dir} = 2^{1} \times 2^{30} \ dir$
 - #bits de direccionamiento = $\log_2(2^1 \times 2^{30}) = 31$ bits
 - Memoria de 512 MB con direccionamiento a 16 bits

 S12 MB 29 × 220 B 228 V
 - #directiones = $\frac{512 \text{ MB}}{2 \text{ B/dir}} = \frac{2^9 \times 2^{20} \text{B}}{2 \text{ B/dir}} = 2^{28} \text{ dir}$
 - #bits de direccionamiento = $\log_2(2^{28}) = 28 \ bits$

- ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?
- Memoria de 2 GB con direccionamiento a byte
 - #directiones = $\frac{2 \text{ GB}}{1 \text{ B}/dir} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B}/dir} = 2^1 \times 2^{30} \ dir$ • #bits de direccionamiento = $\log_2(2^1 \times 2^{30}) = 31$ bits
 - Memoria de 512 MB con direccionamiento a 16 bits
 - #directiones = $\frac{512 \text{ MB}}{2 \text{ B/dir}} = \frac{2^9 \times 2^{20} \text{B}}{2 \text{ B/dir}} = 2^{28} dir$
 - #bits de direccionamiento = $\log_2(2^{28}) = 28 \ bits$
- ¡Qué tamaño tiene la memoria si se usan 25 bits para
- direccionarla y tenemos direccionamiento a nibble (4 bits)? \blacksquare #directiones = 2^{25}

- ¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?
- Memoria de 2 GB con direccionamiento a *byte* #direcciones = $\frac{2 \text{ GB}}{1 \text{ R}/dir} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ R}/dir} = 2^1 \times 2^{30} \text{ dir}$
 - #bits de direccionamiento = $\log_2(2^1 \times 2^{30}) = 31 \ bits$
 - 2 Memoria de 512 MB con direccionamiento a 16 bits $^{-1}$
 - #direcciones = $\frac{512 \text{ MB}}{2 \text{ B}/dir} = \frac{2^9 \times 2^{20} \text{B}}{2 \text{ B}/dir} = 2^{28} \ dir$ • #bits de direccionamiento = $\log_2(2^{28}) = 28 \ bits$
- ¿Qué tamaño tiene la memoria si se usan 25 *bits* para direccionarla y tenemos direccionamiento a *nibble* (4 *bits*)?
- #direcciones = 2^{25} Tamaño de la Memoria = $2^{25}dir2^2\frac{bits}{dir} = 2^3\frac{B}{bits}2^42^{20}bits$ = 16 MB



¿Qué hicimos?

Para saber la cantidad de direcciones que teníamos que direccionar en cada caso, usamos la siguiente fórmula:

$${\rm cantidad\ de\ direcciones} = \frac{{\rm tama\~no\ de\ la\ memoria}}{{\rm unidad\ de\ direccionamiento}}$$

¿Qué hicimos?

Para saber la cantidad de direcciones que teníamos que direccionar en cada caso, usamos la siguiente fórmula:

$${\rm cantidad\ de\ direcciones} = \frac{{\rm tama\~no\ de\ la\ memoria}}{{\rm unidad\ de\ direccionamiento}}$$

Una vez encontrada la cantidad de direcciones de memoria, utilizamos el logaritmo en base 2 para calcular la cantidad de bits necesarios para poder hacer referencia a todas las direcciones de memoria.

Arquitectura

¿Qué es la Arquitectura de una Computadora?

Es el conjunto de reglas e instrucciones que describen su funcionalidad, así como su organización e implementación.

¿Qué es la Arquitectura de una Computadora?

Es el conjunto de reglas e instrucciones que describen su funcionalidad, así como su organización e implementación.

A nivel general, podemos afirmar que la máquina **ORGA1** tiene arquitectura de *von Neumann*. Esto implica que:

¿Qué es la Arquitectura de una Computadora?

Es el conjunto de reglas e instrucciones que describen su funcionalidad, así como su organización e implementación.

A nivel general, podemos afirmar que la máquina **ORGA1** tiene arquitectura de *von Neumann*. Esto implica que:

• Tiene un **CPU** que cuenta con una ALU y registros internos.

¿Qué es la Arquitectura de una Computadora?

Es el conjunto de reglas e instrucciones que describen su funcionalidad, así como su organización e implementación.

A nivel general, podemos afirmar que la máquina **ORGA1** tiene arquitectura de *von Neumann*. Esto implica que:

- Tiene un **CPU** que cuenta con una ALU y registros internos.
- Tiene una Memoria que guarda tanto datos como instrucciones.

¿Qué es la Arquitectura de una Computadora?

Es el conjunto de reglas e instrucciones que describen su funcionalidad, así como su organización e implementación.

A nivel general, podemos afirmar que la máquina **ORGA1** tiene arquitectura de *von Neumann*. Esto implica que:

- Tiene un CPU que cuenta con una ALU y registros internos.
- Tiene una Memoria que guarda tanto datos como instrucciones.
- Tiene dispositivos de Entrada/Salida para interactuar con el 'afuera' de la máquina.

Registros

Registros de Propósito General

- 8 registros de **propósito general** de 16 bits.
 - R0
 - ...
 - R7
- Se utilizan para poder traer datos desde Memoria hacia el CPU y poder operar con ellos.
- Están disponibles para que el programador los use a su antojo.

Registros de Propósito Específico

- 3 registros de **propósito específico** de 16 *bits*.
 - PC (Program Counter): Apunta a la dirección que contiene la próxima instrucción a ejecutar.
 - **SP** (Stack Pointer¹): Apunta a la próxima posición libre de la pila.
 - IR (Instruction Register): Guarda la instrucción a ser inmediatamente ejecutada por el CPU.
- No son directamente accesibles por el programador. Sólo pueden ser modificados por la ejecución de instrucciones.

¹El **Stack** es una estructura de datos en **Memoria** que veremos hacia el final de la clase.

FLAGS

- Los **FLAGS** también son registros de 1 bit.
- Sus valores cambian en función de la ejecución de algunas operaciones. No todas las operaciones alteran los FLAGS.
- Son los mismos que vimos en los Talleres de Lógica Digital.
 - **Z** (zero)
 - N (negative)
 - C (carry)
 - V (overflow)

Memoria

Memoria

- La **Memoria** tiene un tamaño fijo con 65520 posiciones $(2^{16} 16 \text{ posiciones}).$
- Cada posición de Memoria tiene asociada una dirección de 16 bits. Entonces:
 - Las posiciones de Memoria se numeran en el rango de 0x0000 a 0xFFFF.
 - Quedan 16 direcciones libres (0xFFF0 a 0xFFFF) que no apuntan a posiciones de Memoria. En cambio, usan para mapear registros de E/S.²
- La Memoria tiene direccionamiento a palabra. Eso significa que:
 - Cada posición de Memoria puede guardar una palabra.
 - Las palabras en la ORGA1 son datos de 16 bits.

²Más de esto más adelante en la materia.

Set de Instrucciones

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 on	efecto		

operación	cod. op.	efecto	modifica flags
MOV d , f	0001	$d \leftarrow f$	no
ADD d , f	0010	$d \leftarrow d + f$ (suma binaria)	sí
SUB d , f	0011	$d \leftarrow d - f$ (resta binaria)	sí
AND d , f	0100	$d \leftarrow d$ and f	SÍ (*)
OR d , f	0101	$d \leftarrow d \text{ or } f$	SÍ (*)
CMP d , f	0110	Modifica los flags según el resultado de $d-f$.	sí
ADDC d , f	1101	$d \leftarrow d + f + \textit{carry}$ (suma binaria)	sí

 $^{(\}star)$ dejan el flag de carry (C) y el de $\mathit{overflow}$ (V) en cero.

Tipo 2: Instrucciones de un operando

Tipo 2a: Instrucciones de un operando destino.

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

operacin	cod. op.	efecto	modifica flags
NEG d	1000	$d \leftarrow$ el inverso aditivo de d	s
NOT d	1001	$d \leftarrow not\ d\ (bit\ a\ bit)$	S (*)

 $^{(\}star)$ deja el flag de carry (C) y el de overflow (V) en cero.

Tipo 2b: Instrucciones de un operando fuente.

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

operacin	cod. op.	efecto	modifica flags
$JMP\ f$	1010	$PC \leftarrow f$	no
CALL f	1011	$[SP] \leftarrow PC, SP \leftarrow SP - 1, PC \leftarrow f$	no

Tipo 3: Instrucciones sin operandos

		4 bits	6 bits	6 bits
		cod. op.	000000	000000
eración	cod. op.	efecto		

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

• No modifica flags.

Tipo 4: Saltos relativos condicionales

- el salto se produce si se cumple la condición de salto correspondiente.
- \bullet PC \leftarrow PC + desplazamiento
- el desplazamiento se representa en complemento a 2 de 8 bits.
- No modifican flags.

8 bits	8 bits
cod. op.	desplazamiento

Codop	Operación	Descripción	Condición de Salto
1111 0001	JE	Igual / Cero	Z
1111 1001	JNE	Distinto	not Z
1111 0010	JLE	Menor o igual	Z or (N xor V)
1111 1010	JG	Mayor	not(Z or(N xor V))
1111 0011	JL	Menor	N xor V
1111 1011	JGE	Mayor o igual	not (N xor V)
1111 0100	JLEU	Menor o igual sin signo	C or Z
1111 1100	JGU	Mayor sin signo	not (C or Z)
1111 0101	JCS	Carry / Menor sin signo	С
1111 0110	JNEG	Negativo	N
1111 0111	JVS	Overflow	V

Modos de Direccionamiento

Modos de Direccionamiento

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.

Ciclo de Vida de un Programa

Programación: Escribir un algoritmo en lenguaje ensamblador (o assembly). Esto es el código fuente.

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

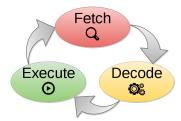
- Programación: Escribir un algoritmo en lenguaje ensamblador (o assembly). Esto es el código fuente.
- Ensamblado: Un programa llamado Ensamblador toma el código fuente y lo traduce a un código máquina.
 - Resuelve las directivas dirigidas al Ensamblador.
 - Que Calcula el tamaño de cada instrucciones,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

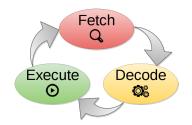
Ciclo de Vida de un Programa

- 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.
 - Resuelve las directivas dirigidas al **Ensamblador**.
 - 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).

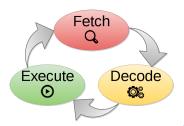
Ciclo de Vida de un Programa

- 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.
 - 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.



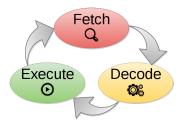


Fetch La UC (Unidad de Control)
obtiene una instrucción de la
posición de Memoria a la que
apunta el PC
y lo incrementa (Si es necesario:
busca más palabras de la instrucción
usando el PC e incrementándolo cada
vez.)



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.



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],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

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

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0×0001

JL fin

JGE main

fin: RET

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

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

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

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

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:

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

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

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0×0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

 0×0000

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

0x0000 dos palabras

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

 0×0000 dos palabras

0×0002

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

0x0004

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

 0×0000 dos palabras

 0×0002 dos palabras

0x0004 dos palabras

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

 0×0004 dos palabras

0x0006

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0×0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

0x0000 dos palabras

0x0002 dos palabras

0x0006 una palabra

dos palabras

0×0004

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

 0×0000 dos palabras

0x0002 dos palabras

0x0004 dos palabras

0x0006 una palabra

0×0007

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa0x0000 dos palabras0x0002 dos palabras0x0004 dos palabras0x0006 una palabra

una palabra

0x0007

main: MOV R1,[[et1]]

ADD [R1],0×6000

JMP main

et1: DW 0x0007

DW 0x0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

0x0000 dos palabras

 $0x0002 \quad {\rm dos\ palabras}$

 0×0004 dos palabras

0x0006 una palabra

0x0007 una palabra

8000x0

main: MOV R1,[[et1]]

ADD [R1],0x6000

JMP main

et1: DW 0x0007

DW 0×0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa0x0000 dos palabras0x0002 dos palabras0x0004 dos palabras0x0006 una palabra

0x0007 una palabra

0x0008 dos palabras

main: MOV R1,[[et1]]

ADD [R1],0x6000

JIVIF IIIaili

et1: DW 0x0007

DW 0×0004

loop: CMP R1,0x0001

JL fin

JGE main

fin: RET

dirección ocupa

0x0000 dos palabras

 0×0002 dos palabras

 0×0004 dos palabras

 0×0006 una palabra

0x0007 una palabra

0x0008 dos palabras

0x000A

main:	MOV R1,[[et1]]
	ADD [R1],0x6000
	JMP main
et1:	DW 0×0007
	DW 0×0004
loop:	CMP R1,0x0001
	JL fin
	JGE main

fin:

RET

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra
0×0008	dos palabras
0×000A	una palabra

main:	MOV R1,[[et1]]
	ADD [R1],0x6000
	JMP main
et1:	DW 0×0007
	DW 0x0004
loop:	CMP R1,0x0001
	JL fin
	JGE main
fin:	RET

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra
0×0008	dos palabras
0×000A	una palabra
0x000B	

main:	MOV R1,[[et1]]
	ADD [R1],0x6000
	JMP main
et1:	DW 0x0007
	DW 0×0004
loop:	CMP R1,0x0001
	JL fin
	JGE main
fin:	RET

dirección	ocupa
0×0000	dos palabras
0×0002	dos palabras
0×0004	dos palabras
0×0006	una palabra
0×0007	una palabra
0×0008	dos palabras
0×000A	una palabra
0x000B	una palabra

dirección

		direction	ocupa
main:	MOV R1,[[et1]]	0×0000	dos palabras
	ADD [R1],0x6000	0×0002	dos palabras
	JMP main	0×0004	dos palabras
et1:	DW 0×0007	0×0006	una palabra
	DW 0×0004	0×0007	una palabra
loop:	CMP R1,0x0001	0×0008	dos palabras
	JL fin	0×000A	una palabra
	JGE main	0x000B	una palabra
fin:	RET	0×000C	

dirección

		direction	ocupa
main:	MOV R1,[[et1]]	0×0000	dos palabras
	ADD [R1],0x6000	0×0002	dos palabras
	JMP main	0×0004	dos palabras
et1:	DW 0×0007	0×0006	una palabra
	DW 0x0004	0×0007	una palabra
loop:	CMP R1,0x0001	0×0008	dos palabras
	JL fin	0×000A	una palabra
	JGE main	0x000B	una palabra
fin:	RET	0×000C	una palabra

dirección

		direction	ocupa
main:	MOV R1,[[et1]]	0×0000	dos palabras
	ADD [R1],0x6000	0×0002	dos palabras
	JMP main	0×0004	dos palabras
et1:	DW 0×0007	0×0006	una palabra
	DW 0×0004	0×0007	una palabra
loop:	CMP R1,0x0001	0×0008	dos palabras
	JL fin	0×000A	una palabra
	JGE main	0×000B	una palabra
fin:	RET	0×000C	una palabra

		dirección	ocupa
main:	MOV R1,[[et1]]	0×0000	dos palabras
	ADD [R1],0x6000	0×0002	dos palabras
	JMP main	0×0004	dos palabras
et1:	DW 0×0007	0×0006	una palabra
	DW 0×0004	0×0007	una palabra
loop:	CMP R1,0x0001	0×0008	dos palabras
	JL fin	0×000A	una palabra
	JGE main	0×000B	una palabra
fin:	RET	0×000C	una palabra

La etiqueta main corresponde a la dirección de memoria 0x0000, et1 corresponde a 0x0006, loop corresponde a 0x0008 y fin corresponde a 0x0000C.

		dirección	ocupa
main:	MOV R1,[[0x0006]]	0×0000	dos palabras
	ADD [R1],0x6000	0×0002	dos palabras
	JMP 0x0000	0×0004	dos palabras
et1:	DW 0×0007	0×0006	una palabra
	DW 0x0004	0×0007	una palabra
loop:	CMP R1,0x0001	0×0008	dos palabras
	JL fin	0×000A	una palabra
	JGE main	0×000B	una palabra
fin:	RET	0×000C	una palabra

La etiqueta main corresponde a la dirección de memoria 0x0000, et1 corresponde a 0x0006, loop corresponde a 0x0008 y fin corresponde a 0x0000.

Reemplazando las etiqutas por valores estamos listos para codificar.

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

Reemplazando las etiqutas por valores estamos listos para codificar.

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. Reemplazando las etiqutas por valores estamos listos para codificar.

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.

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0×0006: DW 0×0007

0×0007: DW 0×0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0×000C: RET

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0×0006: DW 0×0007

0×0007: DW 0×0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0x0006: DW 0x0007

0×0007: DW 0×0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0x0006: DW 0x0007

0×0007: DW 0×0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin: $0 \times 000C - 0 \times 000B$

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0×0006: DW 0×0007

0×0007: DW 0×0004

0x0008: CMP R1,0x0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin: $0 \times 000C - 0 \times 000B = 1_{10}$

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0x0006: DW 0x0007

0×0007: DW 0×0004

0x0008: CMP R1,0x0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

 $0 \times 000 \text{C} - 0 \times 000 \text{B} = 1_{10} =$

 0×01

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0x0006: DW 0x0007

0x0007: DW 0x0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

 0×000 C $- 0 \times 000$ B $= 1_{10} =$

 0×01

desplazamiento main:

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0x0006: DW 0x0007

0x0007: DW 0x0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

 0×000 C $- 0 \times 000$ B $= 1_{10} =$

 0×01

desplazamiento main:

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0×0006: DW 0×0007

0x0007: DW 0x0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

 $0 \times 000 \text{C} - 0 \times 000 \text{B} = 1_{10} =$

0×01

desplazamiento main:

 $0 \times 00000 - 0 \times 0000C$

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0×0006: DW 0×0007

0x0007: DW 0x0004

0x0008: CMP R1,0x0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

 $0 \times 000 \text{C} - 0 \times 000 \text{B} = 1_{10} =$

 0×01

desplazamiento main:

 $0 \times 0000 - 0 \times 000C = -12_{10}$

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0×0006: DW 0×0007

0x0007: DW 0x0004

0×0008: CMP R1,0×0001

0x000A: JL fin

0x000B: JGE main

0x000C: RET

desplazamiento fin:

 $0 \times 000 \text{C} - 0 \times 000 \text{B} = 1_{10} =$

 0×01

desplazamiento main:

 $0 \times 0000 - 0 \times 000C = -12_{10} =$

0×F4

0×0002: ADD [R1],0×6000

0x0004: JMP 0x0000

0×0006: DW 0×0007

0x0007: DW 0x0004

0x0008: CMP R1,0x0001

0x000A: JL 0x01

0x000B: JGE 0xF4

0x000C: RET

desplazamiento fin:

 $0 \times 000 \text{C} - 0 \times 000 \text{B} = 1_{10} =$

0×01

desplazamiento main:

 $0 \times 0000 - 0 \times 000C = -12_{10} =$

0xF4

	dirección	ocupa
MOV R1,[[0x0006]]	0×0000	dos palabras
ADD [R1],0×6000	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
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

Traducidas las etiquetas lo codificamos y carganios en memoria.							
0000	0001	0002	0003	0004	0005	0006	0007
8000	0009	000A	000B	000C	000D	000E	000F



	dirección	ocupa
MOV R1,[[0x0006]]	0×0000	dos palabras
ADD [R1],0×6000	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
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

Traducidas las etiquetas lo codificamos y carganios en memoria.							
0000	0001	0002	0003	0004	0005	0006	0007
8000	0009	000A	000B	000C	000D	000E	000F



	dirección	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0x000B	una palabra
RET	0×000C	una palabra

Traducidas las etiquetas lo codificamos y carganios en memoria.							
0000	0001	0002	0003	0004	0005	0006	0007
8000	0009	000A	000B	000C	000D	000E	000F



	dirección	ocupa
MOV R1,[[0×0006]]	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
CMP R1,0×0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006						
8000	0009	000A	000B	000C	000D	000E	000F



	dirección	ocupa
MOV R1,[[0×0006]]	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
CMP R1,0×0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006						
8000	0009	000A	000B	000C	000D	000E	000F



	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
CMP R1,0×0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000				
8000	0009	000A	000B	000C	000D	000E	000F



	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
CMP R1,0×0001	8000×0	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

Traducidas las etiquetas lo codificamos y cargamos en memoria:

0000	0001	0002	0003	0004	0005	0006	0007	
1858	0006	2C40	6000					
8000	0009	000A	000B	000C	000D	000E	000F	



	dirección	ocupa
MOV R1,[[0x0006]]	0×0000	dos palabras
ADD [R1],0×6000	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
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000		
8000	0009	000A	000B	000C	000D	000E	000F



		dirección	ocupa
МС	OV R1,[[0×0006]]	0×0000	dos palabras
AD	D [R1],0×6000	0×0002	dos palabras
JM	P 0×0000	0×0004	dos palabras
DV	V 0×0007	0×0006	una palabra
DV	V 0×0004	0×0007	una palabra
CN	1P R1,0×0001	0×0008	dos palabra
JL	0×01	0×000A	una palabra
JG	E 0×F4	0×000B	una palabra
RE	Т	0×000C	una palabra
			1

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000		
8000	0009	000A	000B	000C	000D	000E	000F



	dirección	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	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
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

	0000	0001	0002	0003	0004	0005	0006	0007
	1858	0006	2C40	6000	A000	0000	0007	
	8000	0009	000A	000B	000C	000D	000E	000F
ı								



labras
labras
labras
labra
la la la la

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	
8000	0009	000A	000B	000C	000D	000E	000F



	dirección	ocupa
MOV R1,[[0x0006]]	0×0000	dos palabras
ADD [R1],0×6000	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
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	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	0004
8000	0009	000A	000B	000C	000D	000E	000F



	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
CMP R1,0×0001	8000×0	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0x000B	una palabra
RET	0×000C	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	0004
8000	0009	000A	000B	000C	000D	000E	000F



	dirección	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0x0007	0×0006	una palabra
DW 0x0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001						



	dirección	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001						



	dirección	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	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	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001	F301					



	dirección	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

				,	· · · · · · · · · · · · · · · ·		
0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001	F301					



	dirección	ocupa
MOV R1,[[0x0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0xF4	0×000B	una palabra
RET	0×000C	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	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001	F301	FBF4				



	dirección	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0x000B	una palabra
RET	0×000C	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	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001	F301	FBF4				



	direction	ocupa
MOV R1,[[0×0006]]	0×0000	dos palabras
ADD [R1],0×6000	0×0002	dos palabras
JMP 0x0000	0×0004	dos palabras
DW 0×0007	0×0006	una palabra
DW 0×0004	0×0007	una palabra
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0×F4	0×000B	una palabra
RET	0×000C	una palabra

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001	F301	FBF4	C000			



	dirección	ocupa
MOV R1,[[0x0006]]	0×0000	dos palabras
ADD [R1],0×6000	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
CMP R1,0x0001	0×0008	dos palabra
JL 0×01	0×000A	una palabra
JGE 0xF4	0×000B	una palabra
RET	0×000C	una palabra

0000	0001	0002	0003	0004	0005	0006	0007
1858	0006	2C40	6000	A000	0000	0007	0004
8000	0009	000A	000B	000C	000D	000E	000F
6840	0001	F301	FBF4	C000	0000	0000	0000



Caaaambioo Jueeeez!!!



Empezamos a hacer el ciclo **Fetch-Decode-Execute** con el $\mathbf{PC} = 0 \times 0000$.

ĺ	0000	0001	0002	0003	0004	0005	0006	0007
	1858	0006	2C40	6000	A000	0000	0007	0004
	8000	0009	000A	000B	000C	000D	000E	000F
	6840	0001	F301	FBF4	C000	0000	0000	0000

	alores	PC	SP	R0)	R1	R2	R3	R4	R5	R6	R7		Z C	V	N
ini	iciales [0000	FFEF	00	00	0000	0000	0000	0000	0000	000	0 000	10	0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	00 1858	3 0006	2C40	6000	A00	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
003	10 0000	0000	0000	0000	0000	0 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	PC	SP	[SP+	1] IF	₹ Iı	nstrucció	in - 1er p	alabra	PC	2da palabra	PC	3ra palabra	PC	1	strucció	
1	PC	SP	[SP+	1] IF	₹ Ii	nstrucció	n - 1er p	alabra	РС	2da palabra	PC	3ra palabra	PC	1	strucció codifica	
1	PC Ejecució		[SP+	1] IF	R II	nstrucció	in - 1er p	alabra	PC		PC		3	1		
1 2			[SP+	1] IF	R 1	nstrucció	in - 1er p	alabra	PC		PC		3	de		
		n	[SP+	1] IF	R II	nstrucció	in - 1er p	alabra	PC		PC		3	de		
	Ejecució	n	[SP+	1] IF	R I	nstrucció	in - 1er p	alabra	PC		PC		3	Flags		

Va	alores	s PO		SP	RO	1	R1	R2	R3	R4	R5	R6	R7		Z C	V	N
ini	iciales	s 00	000 I	FFEF	00	00	0000	0000	0000	0000	0000	000	000	10	0 0	0	0
		+0	+1	+2	+3	+4	4 +5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	00 1	1858	0006	2C40	6000	A00	000 000	0 0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
00	10 0	0000	0000	0000	0000	000	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	PO	'C	SP	[SP+	1] IF	1	nstruccio	ón - 1er p	alabra	PC	2da palabra	PC	3ra palabra	PC	1	strucció codifica	
		_	SP	[SP+	1] IF	1	nstruccio	ίn - 1er p	alabra	PC		PC		PC	1		
1	0000 Ejecu)	SP	[SP+	1] IF	1	nstruccio	ón - 1er p	alabra	PC		PC		PC	de		
1 2	0000)	SP	[SP+	1] IF	₹ I	nstruccio	ón - 1er p	alabra	PC		PC		PC	1		
	0000) ución	SP	[SP+	1] IF	R I	nstruccio	ón - 1er p	alabra	PC		PC		PC	de		
	0000 Ejecu) ución	SP	[SP+	1] IF	R I	Instruccio	ón - 1er p	palabra	PC		PC		PC	Flags		

Va	alores	PC	5	SP	RO)	R1	F	R2	R3	R4	R5	R6	R7		Z C	V	N
ini	iciales	000	0 F	FFEF	00	000	0000	0	0000	0000	0000	0000	000	0 000	10	0 0	0	0
	+	-0 -	+1	+2	+3	1 +4	4 -	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	00 18	58 00	006	2C40	6000	A00	00 00	000	0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
001	10 00	00 00	000	0000	0000	000	00 00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	PC	\top	SP	[SP+	1] IF	۱ ۶	Instruc	ción	- 1er pa	alabra	PC	2da	PC	3ra	PC	l l	strucció	
				, ,	1] IF	٦ ا	Instruc	ción	- 1er pa	alabra	PC	2da palabra	PC	3ra palabra	PC	l l	strucció codifica	
1	0000	F	SP FEF	[SP+]	1] IF	۱ ۶	Instruc	:ción	- 1er pa	alabra	PC		PC		3	de		
		F		, ,	1] IF	۱ ۶	Instruc	ción	- 1er pa	alabra	PC		PC		3	l l		
1 2	0000	ión F		, ,	1] IF	۱ ۶	Instruc	ción	- 1er pa	alabra	PC		PC		3	de		
	0000 Ejecuc	ión F		, ,	1] IF	R	Instruc	ción	- 1er pa	alabra	PC		PC		3	Flags		

Va	lores	-	SP	R) R	1	R2	R3	R4	R5	R6	R7		ZC	V	N
ini	ciales	0000	FFEF	00	000 0	000	0000	0000	0000	0000	000	000	0	0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	0 1858	0006	2C40	6000	A000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
01	.0 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
										2da		l 3ra		l Ir	strucció	ón Ι
	PC	SP	[SP+:	1]	R Ins	trucció	n - 1er p	alabra	PC	2da	PC	3ra	PC		strucció	
l	PC 0000	SP FFEF	<u> </u>	1]		trucciói	n - 1er p	alabra	PC	palabra	PC	3ra palabra	PC		codifica	
		FFEF	<u> </u>	,		trucció	n - 1er p	alabra	PC		PC		PC			
2	0000	FFEF	<u> </u>	,		trucció	n - 1er p	alabra	PC		PC		PC	de		
	0000 Ejecuciói	FFEF	<u> </u>	,		trucció	n - 1er p	alabra	PC		PC		PC	Flags		

Va	lores F	C S	SP	F	R0	R:	1	R2	R3	R4	R5	R6	R7		ZC	V	N
ini	ciales 0	000	FFEF		0000	00	000	0000	0000	0000	0000	000	0 000	0	0 0	0	0
	+0	+1	+2	+3	. +	-4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000		0006	2C40	600		000	0000		0004	6840	0001	F301	FBF4	C000	0000	0000	0000
001	0000	0000	0000	000	0 00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Т		Т	_	\neg	-				$\overline{}$		0.1	1		_			
	PC 0000 Fiecución	SP FFEF	[SP+	1	IR 1858			n - 1er p		PC	2da palabra	PC	3ra palabra	PC	de	ecodifica	
				1						PC		PC		PC	1		
	0000			1						PC		PC		PC	de		
2	0000 Ejecución			1						PC		PC		PC	Flags		

Va	lores F	PC !	SP	F	20	R1	R2	R3	R4	R5	R6	R7		ZC	V	N
ini	ciales 0	0000	FFEF	0	000	0000	0000	0000	0000	0000	000	000	0	0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000		0006	2C40	6000	A000	0000		0004	6840	0001	F301	FBF4	C000	0000	0000	0000
001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	PC 0000	SP FFEF	[SP+	1			n - 1er p 0 0101 10		PC 0001	2da palabra	PC	3ra palabra	PC	de	codifica	
		FFEF		1							PC		PC	1		
	0000	FFEF		1							PC		PC	de		
2	0000 Ejecución	FFEF		1							PC		PC	Flags		

Va	lores	PC :	SP	R) R	1	R2	R3	R4	R5	R6	R7		ZC	V	N
ini	ciales [0000	FFEF	00	000 00	000	0000	0000	0000	0000	000	000	0	0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000			2C40	6000	A000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
001	.0 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
\neg				$\neg \neg$						245		212		l.	ctruccia	(n
1	PC	SP	[SP+	,			1 - 1er pa			2da palabra	PC	3ra palabra	PC	1	nstruccio codifica	
	0000	FFEF	١,	1] IF			1 - 1er pa 0101 10				PC 0002		PC	de		
		FFEF	١,	,						palabra			PC	1		
	0000	FFEF	١,	,						palabra			PC	de		
1 2 3	0000 Ejecució	FFEF	١,	,						palabra			PC	Flags		

٧a	lores F	PC S	SP	RO) R	1	R2	R3	R4	R5	R6	R7		Z		/	N
ini	ciales 0	0000 I	FFEF	00	00 00	000	0000	0000	0000	0000	000	000	0 [0 () () (0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+	-E	+F
000	00 1858	0006	2C40	6000	A000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	00	000	0000
001	10 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	000	0000
П	DC.	CD	[CD	11 15) Inct	ión	1 or n	alabra	DC.	2da	DC.	3ra	DC.		Instru	ıcció	n
	PC	SP	[SP+				ı - 1er pa			palabra	PC	3ra palabra	PC	d	ecod	ificad	da
1	0000	FFEF	١.	1] IF			0101 10				PC 0002		PC	MOV	ecod R1,	ificad	
		FFEF	١.							palabra			PC	d	ecod R1,	ificad	da
L 2	0000	FFEF	١.							palabra			PC	MOV	R1,	ificad	da
	0000 Ejecución	FFEF	١.							palabra			PC	MOV Flags	R1,	ificad	da

Va	lores F	PC S	SP		R0	R:	1	R2	R3	R4	R5	R6	R7		Z	١ [/	N
ini	ciales 🛛 C	0000 I	FFEF		0000	00	000	0000	0000	0000	0000	000	000	0	0 0) () (0
	+0	+1	+2	+3	3 +	⊦4	+5	+6	+7	+8	+9	+A	+B	+C	+D	1	-E	+F
000		0006	2C40	600	0 A	000	0000		0004	6840	0001	F301	FBF4	C000	0000	00	000	0000
001	.0 0000	0000	0000	000	0 00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	000	0000
1	PC 0000	SP FFEF	[SP+3		IR 1858			n - 1er p		PC 0001	2da palabra 0006	PC 0002	3ra palabra	PC	d	ecod	icció ificad	
		FFEF	١.		1858	000					palabra			PC	d	ecod R1,	ificad	da
1	0000	FFEF R1=[[0000		1858	000					palabra			PC	d MOV	R1,	ificad	da
	0000 Ejecución	FFEF R1=[[0000		1858	000					palabra			PC	MOV Flags	R1,	ificad	da

Va	alores	PC	SP	R	R	1	R2	R3	R4	R5	R6	R7		ZC	V	N
ini	iciales [0000	FFEF	00	00 0	000	0000	0000	0000	0000	000	000	0	0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00 1858	0006	2C40	6000	A000	000	0 0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
00	10 0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	PC 0000	SP FFEF	[SP+	1] IF			ón - 1er p		PC 0001	2da palabra 0006	PC 0002	3ra palabra	PC	de	struccio codifica	da
) I I I U U	0 0101 1	000 j	0001	0000	0002			IVIOV	ivi, llox	loooojj
1														Flags		
2	Ejecució 0002		[0×0006]										T	Flags		
	Ejecució	n R1=[Flags Flags		
	Ejecució 0002	n R1=[

٧a	lores	PC	3	SP		R0	R:	1	R2	R3	R4	R5	R6	R7		ZC	V	′ [I	N
ini	ciales [000	00 F	FFEF		0000	00	000	0000	0000	0000	0000	000	000	0	0 0	0	()
	+0)	+1	+2	+3	-	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+	E	+F
00	00 185	8 (0006	2C40	600	0 A	000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	000	00	0000
00	10 000	0 (0000	0000	000	0 00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	00	0000
					$\overline{}$														
1	PC 0000		SP FFEF	[SP+		IR 1858			n - 1er p		PC 0001	2da palabra 0006	PC 0002	3ra palabra	PC	d€	stru codi R1. I	ficad	la
L	PC 0000 Ejecució		FFEF	[SP+: 0000 0x0006]		1858	000		n - 1er p		PC 0001		PC 0002		PC	l	codi	ficad	la
	0000		FFEF	0000]=[0:	1858	000					palabra			PC	de MOV	codi	ficad	la
	0000 Ejecució	ón	FFEF	0000]=[0:	L858 (0007]	000					palabra			PC	de MOV	codi	ficad	la
2	0000 Ejecució 0002	ón	FFEF	0000]=[0:	L858 (0007]	000					palabra			PC	MOV Flags	codi	ficad	la

Va	lores	PC	S	SP	R	0	R1	R2	2	R3	R4	R5	R6	R7		Z	: \	/ [V
ini	ciales [0000	0 F	FFEF	00	000	0000	00	000	0000	0000	0000	000	000	0	0 0	0) ()
	+0) -	+1	+2	+3	+4	4 +	5	+6	+7	+8	+9	+A	+B	+C	+D	1 +	Ε	+F
000			006	2C40	6000	A00	00 00	00	0007	0004	6840	0001	F301	FBF4	C000	0000	00	00	0000
001	10 000	0 00	000	0000	0000	000	00 00	00	0000	0000	0000	0000	0000	0000	0000	0000	00	00	0000
Т		\top		Ico.	., ,	П.		.,			DC	2da	DC	3ra	D.C.		nstru	cciór	1
1	PC		SP	[SP+1	,		Instrucc				PC 0001	palabra	PC	3ra palabra	PC	d	ecodi	ficad	a
1	PC 0000 Eiecució	F	FFEF	0000	18	58 (0001 10				PC 0001		PC 0002		PC	d MOV	ecodi R1,	ficad	a
	0000	F	FFEF	١,	18]=[0×0	58 C	0001 10	000 01	101 10	000		palabra			PC	d	ecodi R1,	ficad	a
1 2	0000 Ejecució	f ón R	FFEF	0000	18]=[0×0	58 C	0001 10	000 01	101 10	000		palabra			PC	d MOV	R1,	ficad	a
	0000 Ejecució 0002	f ón R	FFEF	0000	18]=[0×0	58 C	0001 10	000 01	101 10	000		palabra			PC	MOV Flags	R1,	ficad	a

Va	lores P	C S	SP	RO) R	1 F	R2	R3	R4	R5	R6	R7		ZC	. V	N	
nic	ciales 0	000 F	FFEF	00	00 00	000	0000	0000	0000	0000	000	000	0	0 0	0	0	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+	E	+F
000	00 1858	0006	2C40	6000	A000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	000	0 00	0000
001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0 00	0000
		CD	[CD	11 15	, Inci	·rucción	lor no	alabra	DC.	2da	DC.	3ra	DC.	I	nstru	cción	
	PC	SP	[SP+1	1] IF	lnst	:rucción	- 1er pa	alabra	PC	2da palabra	PC	3ra palabra	PC			cción ficada	
_	0000	FFEF	0000	185	58 000		- 1er pa		PC 0001		PC 0002		PC	MOV	R1, [ficada	
	0000 Ejecución	FFEF	١,	185]=[0×00	58 000 007]=4	1 1000	0101 10	000	0001	palabra			PC	de	R1, [ficada	
2	0000 Ejecución 0002	FFEF	0000	185	58 000 007]=4	1 1000		000		palabra			PC	MOV Flags	R1, [ficada	
2	0000 Ejecución	FFEF	0000	185]=[0×00	58 000 007]=4	1 1000	0101 10	000	0001	palabra			PC	MOV	R1, [ficada	
2	0000 Ejecución 0002	FFEF	0000	185]=[0×00	58 000 007]=4	1 1000	0101 10	000	0001	palabra			PC	MOV Flags	R1, [ficada	

Val	lores P	C S	SP	RO) R	1 F	R2	R3	R4	R5	R6	R7		ZC	V	N
nic	ciales 00	000 F	FFEF	00	00 00	000	0000	0000	0000	0000	000	000	0	0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	E +F
000	0 1858	0006	2C40	6000	A000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	000	0000
001	0 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0000
Т	PC PC	SP	[SP±	1] IE	, Inst	rucción	- 1er no	alahra	PC	2da	PC	3ra	PC		nstruc	
	PC	SP	[SP+1	1] IF	≀ Inst	rucción	- 1er pa	alabra	PC	2da palabra	PC	3ra palabra	PC		nstruc	
_	0000	FFEF	0000	185	58 000		- 1er pa		PC 0001		PC 0002		PC	de MOV	codif	
	0000 Ejecución	FFEF	١,	185]=[0×00	58 000 007]=4	1 1000	0101 10	000	0001	palabra 0006	0002		PC	d€	codif	icada
2	0000 Ejecución 0002	FFEF	0000	185	58 000 007]=4	1 1000		000	0001	palabra			PC	MOV Flags	codif	icada
2	0000 Ejecución	FFEF	0000	185]=[0×00	58 000 007]=4	1 1000	0101 10	000	0001	palabra 0006	0002		PC	de MOV	codif	icada
2	0000 Ejecución 0002	FFEF	0000	185]=[0×00	58 000 007]=4	1 1000	0101 10	000	0001	palabra 0006	0002		PC	MOV Flags	codif	icada

Va	lores	PC	SP] [R0	R	1	R2	R3	R4	R5	R6	R7		Z	: \	/ 1	1
ini	ciales [0000	FFEF] [000	0 00	000	0000	0000	0000	0000	000	000	0 [0 0	0	()
	+0	+1	+2	4	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+	Εİ	+F
000			2C40			A000	0000		0004	6840	0001	F301	FBF4	C000	0000	00		0000
001	.0 0000	0000	0000	00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	00	0000
_			_											_			.,	
	PC	SP	[SP-		IR			1 - 1er pa		PC	2da palabra	PC	3ra palabra	PC	d	nstru	ficad	a
I	0000	FFE	= 0000)	1858	000		n - 1er pa		PC 0001		PC 0002		PC	d MOV	ecodi	ficad	a
		FFE	<u>'</u>)	1858	000 7]=4	1 1000		000	0001	palabra			PC	d	R1,	ficad [[0×0	a 006]]
	0000 Ejecució	FFE on R1=	= 0000)	1858 0×000	000 7]=4	1 1000	0101 10	000	0001	palabra 0006	0002		PC	MOV Flags	R1, R1, R1],	ficad [[0×0	a 006]]
2	0000 Ejecució 0002	FFE on R1=	= 0000)	1858 0×000	000 7]=4	1 1000	0101 10	000	0001	palabra 0006	0002		PC	MOV Flags ADD	R1, R1, R1],	ficad [[0×0	a 006]]

Va	lores F	PC S	SP	F	30	R1		R2	R3	R4	R5	R6	R7		Z	۱ ا	V	N
ini	ciales 0	0000 I	FFEF		0000	000	00	0000	0000	0000	0000	000	000	0	0 0) ()	0
	+0	+1	+2	+3	1 +	-4	+5	+6	+7	+8	+9	+A	+B	+C	+D		⊦E Ì	+F
000		0006	2C40	6000			0000	0007	0004	6840	0001	F301	FBF4	C000	0000		000	0000
00	10 0000	0000	0000	0000	000	00	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	000	0000
\neg		т —	Т								0.1	1						
1	PC	SP	[SP+:					1 - 1er p		PC	2da palabra	PC	3ra palabra	PC	d		ifica	da
1	0000	FFEF	0000	1	858	0001		0101 10		PC 0001		PC 0002		PC	MOV	ecod R1,	ifica	
		FFEF	١,	1]=[0x	858 0007]=	0001	1000		000		palabra			PC	d	R1,	ifica	da 0006]]
1 2	0000 Ejecución	FFEF R1=[[0000]=[0x	858 0007]= C40	0001 =4 0010	1000	0101 10	000	0001	palabra 0006	0002		PC	MOV Flags	R1, [R1]	ifica	da 0006]]
	0000 Ejecución 0002	FFEF R1=[[0000 [0×0006]]]=[0x	858 0007]= C40	0001 =4 0010	1000	0101 10	000	0001	palabra 0006	0002		PC	MOV Flags ADD	R1, [R1]	ifica	da 0006]]

	alores iciales	P(SP FFEF		R0	R:	1000	R2 0000	R3 0000	R4	R5	R6	R7	00	Z C	V 0	N 0
		_				_				_								
	+	-	+1	+2	+3	3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00 18	58	0006	2C40	600	00	0000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
00	10 000	00	0000	0000	000	00 (0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	PC		SP	[SP+	1]	IR	Inst	rucció	n - 1er p	alabra	PC	2da palabra	PC	3ra palabra	PC	1	nstrucci	
1	0000		FFEF	0000		1858	000	1 1000	0101 10	000	0001	0006	0002			MOV	R1, [[0×	:0006]]
	Ejecuci	ón	R1=[[0×0006]	=[0;	×0007	7]=4									Flags		
2	0002					2C40	001	0 1100	0100 0	000	0003	6000	0004			ADD	[R1],0x6	000
	Ejecuci	ón	[0×000	04]=0×A	-000	+0×6	000=0	×0000								Flags	1 1	0 0
3																		
	Ejecuci	ón														Flags		

Va	lores P	C !	SP	R) R	1	R2	R3	R4	R5	R6	R7		Z C	V	N	
ini	ciales 0	000	FFEF	00	00 0	000	0000	0000	0000	0000	000	000	0	0 0	0	0	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	Ξ	+F
000	00 1858	0006	2C40	6000	0000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	000	0 (0000
00	10 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0 (0000
		1											_				$\overline{}$
	PC	SP	[SP+	1]	R Ins	trucción	- 1er p	alabra	PC	2da	PC	3ra	PC	1	istruc		
1				-1						palabra	' -	3ra palabra	PC	de	codif	icada	
1	PC 0000 Ejecución	FFEF	[SP+ 0000 0×0006]	18	58 000		0101 10		PC 0001		PC 0002		PC	1	codif	icada	
	0000	FFEF	0000	18	58 000 007]=4	1000		000		palabra	' -		PC	MOV	codif R1, [[0×00	06]]
2	0000 Ejecución	FFEF R1=[[0000	18:]=[0x00	58 000 007]=4 40 001	1000	0101 10	000	0001	palabra 0006	0002		PC	MOV Flags	codif R1, [[0×00	006]]
	0000 Ejecución 0002	FFEF R1=[[0000 0×0006]	18:]=[0x00	58 000 007]=4 40 001	1000	0101 10	000	0001	palabra 0006	0002		PC	MOV Flags	codif R1, [[0×00 0×600	006]]

	alores iciales	PC 0000	SP FFEF		R0 0000	R:		R2 0000	R3 0000	R4	R5	R6	R7	0	Z 0	2	V 0	N 0	
III	iciales [0000	FFEF	Ľ	0000	00	000	0000	0000	0000	0000	000	0 000	0	0 0	_	U	U	
	+0	+1	+2	+3	3 -	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D		+E	+	F
00	00 185	3 0006	2C40	600	00 00	000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	C	0000	000	00
00	10 000	0000	0000	000	00 00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0	0000	000	00
																			_
		_	Τ.								2da	Ι	3ra			Inst	ruccio	śn	\neg
	PC	SP	[SP+	1]	IR	Inst	rucciói	1 - 1er p	alabra	PC	palabra	PC	palabra	a PC	1		difica		
1	0000	FFE	F 0000	1	1858	000	1 1000	0101 10	000	0001	0006	0002			MOV	' R1	, [[0×	0006	
	Ejecució	n R1=	[[0×0006]]=[0>	×0007]	=4									Flags				
2	0002			2	2C40	001	0 1100	0100 0	000	0003	6000	0004			ADD	[R1],0x6	000	
	Ejecució	n [0x0	004]=0×A	0000	+0×600	00=0	×0000								Flags	1	. 1	0	0
3	0004			(0000														
	Ejecució	in	-									•	•	-	Flags				П
																			_

V	alores	PC	S	P		R0	R	1	R2	R3	R4	R5	R6	R7		Ζ	C	V	Ν	
in	iciales	0000	F	FEF		000	0 00	000	0000	0000	0000	0000	000	0 000	00	0	0	0	0]
	+() +	1	+2	+	-3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D)	+E	1	⊦F Ì
00	00 185	8 000	06	2C40	60	00	0000	0000	0007	0004	6840	0001	F301	FBF4	C000	000	0	0000	00	000
00	10 000	0 00	00	0000	00	00	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0	0000	00	000
	PC	S	iP	[SP+	1]	IR	Inst	rucció	n - 1er p	alabra	PC	2da palabra	PC	3ra palabr	PC			trucci odific		
1	0000	FF	EF	0000		1858	3 000	1 1000	0101 1	000	0001	0006	0002	ľ		MOV	V R	1, [[0:	×000	6]]
	Ejecucio	ón R1	=[[0)×0006]	<u>=[0</u>	000×0	7]=4									Flag	s	Ť		Ϊ
2	0002					2C40	001	0 1100	0100 0	000	0003	6000	0004			ADE) [R	1],0x	6000	
	Ejecucio	ón [0:	<000	4]=0×A	.000	+0×6	5000=0	×0000								Flag	s	1 1	0	0
3	0004					0000	000	0 0000	0000 0	000										
	Ejecucio	ón														Flag	s			

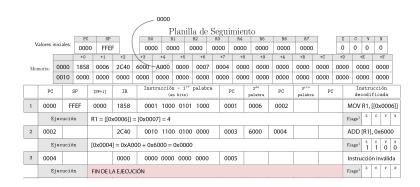
V	alores	PC	S	SP	R	09	R	1	R2	R3	R4	R5	R6	R7		Ζ	C	V	N
in	iciales	000	0 F	FEF	0	000	00	000	0000	0000	0000	0000	000	0 000	00	0	0	0	0
	+0) -	+1	+2	+3	+	-4	+5	+6	+7	+8	+9	+A	+B	+C	+D	1	+E	+F
00	00 185	8 0	006	2C40	6000	00	00	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	0 (0000	0000
00	10 000	0 0	000	0000	0000	00	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0 (0000	0000
	PC		SP	[SP+1	1]	IR	Inst	rucciói	n - 1er p	alabra	PC	2da palabra	PC	3ra palabra	PC			ruccio	
1	0000	F	FEF	0000	18	858	000	1 1000	0101 10	000	0001	0006	0002			MON	/ R1	. [[0×	0006]]
	Ejecuci	ón F	R1=[[0	0×0006]]	=[0x0	0007]=	=4									Flag	s	T	П
2	0002				20	C40	001	0 1100	0100 0	000	0003	6000	0004			ADD	(R:	1],0x6	000
	Ejecuci	ón [0×000	4]=0×A	+000	0×600	0=0	×0000								Flag	s I	1 1	0 0
3	0004				00	000	000	0 0000	0000 0	000									
	Ejecuci	ón														Flag	s		

V	alores	PC	SP		R0	R	1	R2	R3	R4	R5	R6	R7		Z 0	C	V	Ν	
in	iciales [0000	FFEF		000	00 00	000	0000	0000	0000	0000	000	0 000	00	0 ()	0	0]
	+() +1	+2		+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D		+E	1 +	-F
00	00 185	8 000	5 2C40) 6	000	0000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000) (0000	00	000
00	10 000	000	0000	0	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000) (0000	00	000
	PC	SF) [SP	+1]	IR	Inst	rucció	n - 1er p	alabra	PC	2da palabra	PC	3ra palabr	PC	1		rucci		
1	0000	FFE	F 000	0	185	8 000	1 1000	0101 10	000	0001	0006	0002			MOV	/ R1	., [[0:	(0000	6]]
	Ejecucio	ón R1:	000×0]]=	6]]=	0×000	07]=4									Flags	; [Ť		Ü
2	0002				2C4	0 001	0 1100	0100 0	000	0003	6000	0004			ADD	[R:	1],0x6	5000	
	Ejecucio	ón [0x	0004]=0	(A00	0+0×	6000=0	0000x								Flags	; []	1	0	0
3	0004				000	0 000	0 0000	0000 0	000	0005									
	Ejecucio	ón													Flags	;			

			SP	R0	R	_	R2	R3	R4	R5	R6	R7		Z C	V	N
ini	iciales 0	0000 I	FFEF	000	0 00	000	0000	0000	0000	0000	000	000	0	0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00		0006	2C40	6000	0000	0000		0004	6840	0001	F301	FBF4	C000	0000	0000	0000
00		0000	0000	0000	0000	0000		0000	0000	0000	0000	0000	0000	0000	0000	0000
_																
	PC	SP	[SP+1	l] IR	Ins	trucció	ón - 1er p	alabra	PC	2da	PC	3ra	PC		strucció	
			1,	,			· ·			palabra		palabra			codifica	
1	0000	FFEF	0000	185	8 000	100 (0 0101 1	000	0001	0006	0002			MOV	R1, [[0×	0006]]
	Ejecución	R1=[[0×0006]]	=[0x00]	07]=4									Flags		
2	0002			2C4	0 001	0 110	0 0100 0	000	0003	6000	0004			ADD	R1],0x6	000
	Ejecución	[0x000	04]=0×A	000+0×	6000=0	0000×0)							Flags	1 1	0 0
3	0004			000	0 000	000 000	0 0000 0	000	0005					Instruc	ción Inv	/álida
	Ejecución											•		Flags		

V	alores	PC	9	SP		R0	R	1	R2	R3	R4	R5	R6	R7		Z (2 V	N
in	iciales	0000	0 1	FEF		000	0 00	000	0000	0000	0000	0000	000	000	00	0 (0	0
														,				
	+0) -	+1	+2	-	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00 185	8 00	006	2C40	61	000	0000	0000	0007	0004	6840	0001	F301	FBF4	C000	0000	0000	0000
00	10 000	0 00	000	0000	0	000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	PC.		SP	ISP+	11	IR	Inct	rucció	n - 1er p	salabra	PC.	2da	PC.	3ra	PC		Instrucc	ión
	1.0		Ji	[51 +	-1	111	11131	iluccic	nı - Iei b	Jaiabia	1 C	palabra	1 0	palabra	a ' `	(lecodific	ada
1	0000	F	FEF	0000		1858	3 000	1 100	0 0101 1	000	0001	0006	0002			MOV	' R1, [[0	×0006]]
	Ejecuci	ón F	1=[[0×0006]]=[0×000	7]=4									Flags		
2	0002					2C4	001	0 110	0 0100 0	000	0003	6000	0004			ADD	[R1],0x	6000
	Ejecuci	ón [(000xC	04]=0×A	00	0+0×6	5000=C)×0000			'					Flags	1 1	0 0
3	0004					0000	000	000 0	0 0000 0	000	0005					Instr	ıcción lı	nválida
	Ejecuci	ón F	IN D	E LA E.	JEC	CUCIÓ	N			·			•			Flags		

Ejercicio 1 - Solución



¿Qué pasó con el programa anterior?

¿Qué pasó con el programa anterior?

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

¿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.

¿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:



Programen con cuidado y atención.

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

Ejecución

Ejecución

Va	lores	PC		SP	F	R0	R1		R2	R	23	R4	R5	R6	R7		Z	C	V	N
ini	iciales	00	00 F	FEF		0000	000	00	0000	0	000	0000	0000	0000	000	0	0	0	0	0
		^	. 1			2					. 7									
	+	U	+1	+2	+3	3	+4	+5) -	+6	+7	+8	+9	+A	+B	+C	+	D	+E	+F
00	00 190	В	0004	B000	000	05 0	FE0	39A	17 CI	000	0000	000	0000	0000	0000	0000	00	000	0000	0000
00	10 000	0	0000	0000	000	00 0	000	000	0 00	000	0000	000	0 0000	0000	0000	0000	00	000	0000	0000
	PC		SP	[SP+1] IR Instrucción - 1er palabra PC 2da PC 3ra palabra PC palabra PC											Instrucción decodificad					
1																				
	Ejecuci	ón															Fla	ags		
2																			•	
	Ejecuci	ón															Fla	ags		
3																				
	Ejecuci	ón															Fla	ags		

Flags

V	alores	P		SP	R0	R1		R2	R3	R	4	R5	R6	R7		Z	C	V	N	
in	iciales	00	000 F	FEF	000	00 00	00	0000	0000	00	000	0000	0000	000	0	0	0	0	0	
_																				_
		-0	+1	+2	+3	+4	+5	+6	5 +7		+8	+9	+A	+B	+C	+	-D	+E	+	-F
00	00 19	CB	0004	B000	0005	0FE0	39A	7 C00	000 000	0 (0000	0000	0000	0000	0000	00	000	0000	00	000
00	10 00	00	0000	0000	0000	0000	0000	000	0 000	0 (0000	0000	0000	0000	0000	00	000	0000	00	000
	PC		SP	[SP+1	.] IR	Inst	rucció	n - 1er	palabra	P		2da palabra	PC	3ra palabra	PC			struc		
1	0000																			\neg
	Ejecuc	ión														FI	ags			
2																				\Box
	Ejecuc	ión														FI	ags			
3																				\Box
	Ejecuc	ión														FI	ags			
4																				
	Ejecuc	ión														FI	ags			
5																				
	Fiecuc	ión														FI	ags			П

Ejecución

Ejecución

Ejecución

Ejecución

3

4

5

	ores P		FEF	R0 000	R1 0 00			R3 0000	R4 0000	R5 0 0000	R6	R7)	Z C 0		N 0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	3 +9	+A	+B	+C	+D	+E	+F
000	0 19CB	0004	B000	0005	0FE0	39A7	C000	0000	000	0 0000	0000	0000	0000	0000	0000	0000
001	0 0000	0000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	0000	0000	0000
	PC	SP	[SP+1]	IR	Inst	rucción	- 1er pa	labra	PC	2da	PC	3ra	PC		strucció	
										palabra		palabra		de	codifica	da
1	0000	FFEF	0000													
	Ejecución													Flags		
2			1											1		

Flags

Flags

Flags

Ejecución

Ejecución

Ejecución

Ejecución

Ejecución

3

4

5

Va	lores P	CS	P	R0	R1	R	2 F	3	R4	R5	R6	R7		Z C	V	N
ini	ciales 00	000 F	FEF	0000	000	00 00	000 0	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
000	00 19CB	0004	B000	0005	FE0	39A7	C000	0000	000	0000	0000	0000	0000	0000	0000	0000
003	10 0000	0000	0000	0000	0000	0000	0000	0000	000	0000	0000	0000	0000	0000	0000	0000
	PC	SP	[SP+1]	IR	Instr	rucción - 1er palabra			PC	2da palabra	PC	3ra palabra	PC		strucció codifica	
1	0000	FFEF	0000	19CB												

Flags

Flags

Flags

Flags

Ejecución

Ejecución

Ejecución

Ejecución

3

4

5

Valores PC SP iniciales 0000 FFEF				R0 R1			R2 I		R4	R5 0 0000	R6	R7)	Z C 0	V 0	N 0
	+0															
		0004	B000			39A7	C000	0000			0000	0000	0000	0000	0000	0000
001	.0 0000	0000	0000	0000	0000	0000	0000	0000	000	0000	0000	0000	0000	0000	0000	0000
	PC	SP	[SP+1]	IR	Incti	rucción	ler pa	lahra	PC	2da	PC.	3ra	PC	In	strucció	n
	1 C	31	[51 +1]	"	111361	uccion -	Tei pa	labia	1 C	palabra	10	palabra	1.0	de	codifica	da
1	0000	FFEF	0000	19CB	000	1 1001 1	100 100	00								
	Ejecución													Flags		
2															•	

Flags

Flags

Flags

3

4

5

Ejecución

Ejecución

Ejecución

Valores PC SP iniciales 0000 FFEF				R0	R0 R1 0000		R2 R3		R4 0000	R5 0000	R6 0000	R7	0	Z C 0	-	N 0
000	+0 00 19CB	+1	+2 B000	+3 0005	+4 0FE0	+5 39A7	+6 C000	+7	+8		+A 0000	+B 0000	+C	+D	+E	+F
000		0004	0000		0000	0000	0000	0000			0000	0000	0000	0000	0000	0000
													_			
	PC	SP	[SP+1]	IR	Insti	rucción ·	- 1er pa	labra	PC	2da palabra	PC	3ra palabra	PC		strucció codifica	
1	0000	FFEF	0000	19CE	3 000	1 1001 1	100 100	00	0001							
	Ejecución													Flags		
2															•	
	Ejecución													Flags		

Flags

Flags

Ejecución

Ejecución

Ejecución

Ejecución

3

4

5

Va	lores P	CS	SP	R0	R1	R	2 F	3	R4	R5	R6	R7		Z C	V	N
ini	ciales 00	000 F	FEF	0000	0 00	00 00	000	0000	0000	0000	0000	000) [0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	3 +9	+A	+B	+C	+D	+E	+F
00	00 19CB	0004	B000	0005	0FE0	39A7	C000	0000	000	0 0000	0000	0000	0000	0000	0000	0000
00	10 0000	0000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	0000	0000	0000
	PC	SP	[SP+1]	IR	Instr	rucción -	- 1er pa	labra	PC	2da palabra	PC	3ra palabra	PC		strucció codifica	
1	0000	FFEF	0000	19CE	3 0001	1 1001 1	100 100	00	0001	0004	0002					
	Ejecución													Flags		

Flags

Flags

Flags

Va	alores P		SP	R0	R1	R	2 F	7 3	R4	R5	R6	R7		Z C	V	N
in	iciales 00	000 F	FEF	000	00 00	00 00	000 0	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
00	00 19CB	0004	B000	0005	0FE0	39A7	C000	0000	000	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	1 IR	Inct	rucción	- 1er pa	labra	PC	2da	PC	3ra	PC	- In	nstrucci	ón
	1.0	31	[51 +1	, ",	IIISLI	uccion	Tei pa	labia	1 C	palabra	10	palabra	1.0	d€	ecodifica	ıda
1	0000	FFEF	0000	19C	B 000	1 1001 1	100 100	00	0001	0004	0002			MOV	R7, [0x	0004]
	Ejecución													Flags		
2																
	Ejecución													Flags		\Box
3																
	Ejecución													Flags		\Box
4																
	Ejecución												Flags		П	
5																
	Ejecución													Flags		

			SP	R0	R1			R3	R4	R5	R6	R7		Z		N
ın	iciales 0	000 F	FEF	000	00 00	00 0	0000	0000	0000	0000	0000	000	U [0 0	0	0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00 19CB	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
	200	T 65	[CD . 41	Τ	.,		1		2da		3ra	- D.C		nstrucc	ión	
	PC	SP	[SP+1]	IR	Insti	rucción	- 1er pa	labra	PC	palabra	PC	palabra	PC	d	ecodific	ada
1	0000	FFEF	0000	19C	B 000	1 1001	1100 10	00	0001	0004	0002			MOV	R7, [0>	:0004]
	Ejecución	R7=[0	×0004]=0	0x0FE0										Flags		
2																
	Ejecución													Flags		
3																
	Ejecución													Flags		
4																
	Ejecución													Flags		
5																
	Ejecución													Flags		

V	alores	P	CS	SP	R0	R1		R2	R3	R4		R5	R6	R7		Z	С	V	N	
in	iciales	00	000 F	FEF	000	00 00	00	0000	0000	000	0	0000	0000	000	0	0	0	0	0	1
										-										
		⊢0	+1	+2	+3	+4	+5	+6	5 +7	+	8	+9	+A	+B	+C	+	-D	+E		+F
00	00 19	CB	0004	B000	0005	0FE0	39A	7 C00	000 000	0 00	00	0000	0000	0000	0000	00	000	000	0 0	0000
00	10 00	00	0000	0000	0000	0000	0000	000	000	0 000	00	0000	0000	0000	0000	00	000	000	0 0	0000
_																				
	PC		SP	[SP+1]	ı lr	Inst	rucció	n - 1er	palabra	PC		2da	PC	3ra	PC			struc		
					<u> </u>				•		<u> </u>	alabra		palabra	1			codifi		
1	0000		FFEF	0000	19C		1 100	1 1100	1000	0001	0	004	0002			M	OV F	R7, [0	×000	4]
	Ejecu	ión	R7=[0	×0004]=	0x0FE0											Fla	ags			
2	0002										Т									
	Ejecud	ión														Fla	ags			
3																				
	Ejecud	ión														FI	ags			\Box
4																				
	Ejecud	ión														FI	ags			\Box
5																				
	Ejecu	ción														Fla	ags			\Box

5

Ejecución

			FEF	R0 000	R1			R3 0000	R4 0000	R5 0 0000	R6 0000	R7 0 000	0	Z C 0	V 0	N 0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00 19CB	0004	B000	0005	0FE0	39A7	C000	000	000	0 0000	0000	0000	0000	0000	0000	0000
00	10 0000	0000	0000	0000	0000	0000	0000	000	000	0 0000	0000	0000	0000	0000	0000	0000
	PC	SP	[SP+1]			rucción ·			PC	2da palabra	PC	3ra palabra	PC	de	struccio	da
1	0000	FFEF	0000	19CI		1 1001 1	1100 10	00	0001	0004	0002				R7, [0x0	0004]
	Ejecución	R7=[0	×0004]=0											Flags		
2	0002			B000	0											
	Ejecución													Flags		
3													•			
	Ejecución													Flags		
4																
	Ejecución													Flags		

Flags

Ejecución

Ejecución

Va	ores P	CS	SP	R0	R1		R2	R3	R4	R5	R6	R7		Z	C	V	N
inio	ciales 0	000 F	FEF	000	00	00	0000	0000	0000	0000	0000	000	0	0	0	0	0
	10	1		- 2						2 10		- D			D	+E	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	3 +9	+A	+B	+C	+	D	+=	+F
000	0 19CB	0004	B000	0005	0FE0	39A	7 C00	000	000	0000	0000	0000	0000	00	00	0000	0000
001	0 0000	0000 0000 0000 0000		0000	0000	000	000	000	0000	0000	0000	0000	00	00	0000	0000	
	PC	SP	[SP+1] IR	Inst	rucció	n - 1er	palabra	PC	2da palabra	PC	3ra palabra	PC			strucci codifica	
1	0000	FFEF	0000	19C	B 000	1 1001	1 1100	L000	0001	0004	0002			M	OV F	7, [0x	0004]
	Ejecución	R7=[0	×0004]=	0x0FE0										Fla	igs		
2	0002			B00	0 101	1 0000	0000 (0000									
	Ejecución													Fla	igs		
3																	
Ejecución									Fla	igs							

Flags

Flags

Valor	es F	C	SP	R0	R1	. R	2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les 0	000	FFEF	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	+0 19CB	+1 0004	+2 B000	+3 0005	+4 0FE0	+5 39A7	+6 C000	+7 0000	+8 0000	+9 0000	+A 0000	+B 0000	+C		-D	+E	+F

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

Valor	es	PC	SP		R0	R1	R	2 I	R3	R4	R5	R6	R7		Z	C	V	N
inicia	les [0000	FFEF		0000	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	19C	B 000	4 B00) (0005	OFE0	39A7	C000	0000	0000	0000	0000	0000	0000	00	00	0000	0000
0010	0000	0 000	0 0000	(0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	000	0000	0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da palabra	PC	3ra palabra	PC	Instri decod		
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					-
	Ejecución										Flags		
3													
	Ejecución										Flags		
4													
	Ejecución										Flags		
5													
	Ejecución										Flags		

0000 0000

0000

0000

0000 0000

0010 0000

Valor	res	PC	SP		R0	R1	.	R2	R3	R4	R5	R6	R7		Ζ	С	V	N
inicia	les	0000	FFE	F	000	0 00	00 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	19CE	000	4 B0	000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	00	00	0000	0000

0000 0000 0000 0000

0000 0000 0000

0000 0000

	PC	SP	[SP+1]	IR	Instrucción - 1er palabra	PC	2da	PC	3ra	PC	Instrucc	ión
	1.0	31	[31 +1]	111	mistruccion - Ter parabra	1.0	palabra	1.0	palabra	1.0	decodific	ada
1	0000	FFEF	0000	19CB	0001 1001 1100 1000	0001	0004	0002			MOV R7, [0:	(0004]
	Ejecución	R7=[0:	×0004]=0>	OFE0							Flags	
2	0002			B000	1011 0000 0000 0000	0003	0005	0004			CALL 0×000	5
	Ejecución										Flags	
3												•
	Ejecución										Flags	
4												•
	Ejecución										Flags	
5												•
	Ejecución										Flags	

V	alores	P	CS	SP	R0	R1	.	R2	R3	R4	R5	,	R6	R7		Z	C	V	N	
in	iciales	00	000 F	FEF	000	00 00	00 (0000	0000	0000	00	00	0000	000	0	0	0	0	0	1
																_				
	1	-0	+1	+2	+3	+4	+5	+6	+7	+8	3 +	9	+A	+B	+C	+	-D	+E	-	+F
00	00 19	CB	0004	B000	0005	0FE0	39A7	C000	000	000	00 00	00	0000	0000	0000	00	000	000	0 0	000
00	10 00	00	0000	0000	0000	0000	0000	0000	000	000	00 00	00	0000	0000	0000	00	000	000	0	000
	PC		SP	[SP+1]	l IR	Inst	rucción	1 - 1er p	alabra	PC	2da	- 1	PC	3ra	PC			struc		
					·						palab	ra		palabra				codifi		
1	0000		FFEF	0000	19C	B 000	1 1001	1100 1	000	0001	0004		0002			M	OV F	R7, [0	×000	4]
	Ejecuc	ión	R7=[0	×0004]=	0×0FE0											FI	ags			
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005		0004			C	ALL (0×000	5	
	Ejecuc	ión	[0xFFE	F]=0x00	004 SP:	=0xFFE	E PC=	0×0005								FI	ags			\Box
3																				
	Ejecuc	ión														FI	ags			\Box
4																				
	Ejecuc	ión														FI	ags			
5																				
	Ejecuc	ión										FI	ags							

V	alores	PC		SP	R0	R:	L	R2	R3	R4	R5	R6	R7		Z	C	V	N
in	iciales	00	00 F	FEF	00	00 00	00	0000	0000	0000	0000	0000	000	0	0	0	0	0
										·								
		-0	+1	+2	+3	+4	+5	+6	5 +7	+8	+9	+A	+B	+C	+	D	+E	+F
00	00 19	CB	0004	B000	0005	0FE0	39A	7 C00	000 000	000	0000	0000	0000	0000	00	00	0000	0000
00	10 00	00	0000	0000	0000	0000	0000	000	0 000	000	0 0000	0000	0000	0000	00	00	0000	0000
	,														•			
_											2da		3ra					
	PC		SP	[SP+1	1 IR	Inst	rucció			strucc								
			٥.	[51 11	,		luccio	101	parabra	PC	palabra	PC	palabra	PC		de	codific	:ada
1	0000		FFEF	0000	190	B 000	1 100	1 1100 1	1000	0001	0004	0002			M	OV F	R7, [0:	×0004]
	Ejecuc	ión	R7=[0	×0004]=	0×0FE)									Fla	igs		
2	0002				B00	0 101	1 0000	0 0000 0	0000	0003	0005	0004			CA	LL (000×0	5
	Ejecuc	ión	[0×FFE	F]=0x00	004 SP	=0xFFE	E PC=	=0×0005	5						Fla	igs		
3	0005																	
	Ejecuc	ión												_	Fla	igs		
4																		
	Ejecuc	ión													Fla	igs		
5																		
	Ejecuc	ión													Fla	igs		\top

V	alores	PC		SP	R0	R1		R2	R3	R4	R5	R6	R7		Z	C	V	N
in	iciales	00	000 F	FEF	000	00 00	00	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
00	00 19	CB	0004	B000	0005	0FE0	39A7	7 C00	000	000	0000	0000	0000	0000	00	000	0000	0000
00	10 00	00	0000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	00	000	0000	0000
	PC		SP	[SP+1]	ı lr	Inst	rucciói	n - 1er p	palabra	PC.	2da	PC.	3ra	PC			struc	
				[41.1-]	<u> </u>			,			palabra		palabra				codifi	
1	0000		FFEF	0000	19C	B 000	1 1001	1100 1	000	0001	0004	0002			M	OV F	₹7, [0	×0004]
	Ejecuc	ión	R7=[0	×0004]=	0x0FE0										Fla	ags		
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005	0004			CA	ALL (0×000	5
	Ejecuc	ión	[0xFFE	F]=0x00	004 SP	=0xFFE	E PC=	0×0005							Fla	ags		
3	0005		FFEE															
	Ejecuc	ión													Fla	ags		
4																		
	Ejecuc	ión													Fla	ags		
5																		
	Ejecuc	ión													Fla	ags		

Va	alores	PC	C S	SP		R0	R1		R2	2	R3	R4		R5	R6	R7		Z	C	V		N	
in	iciales	00	00 F	FEF	Г	0000	0 000	00	00	00	0000	0000	0	0000	0000	000	0	0	0	0	1	0	
																							_
	+	-	+1	+2		+3	+4	+5	\rightarrow	+6	+7	+8		+9	+A	+B	+C		-D	+	\rightarrow	+F	_
00	00 190	CB	0004	B000	00	005	0FE0	39A	7	C000	0000	000	00	0000	0000	0000	0000	0	000	000	00	0000	i
00	10 000	00	0000	0000	00	000	0000	000	0	0000	0000	000	00	0000	0000	0000	0000	0	000	000	00	0000)
																							_
					_								_	0.1				_		_	.,		_
	PC		SP	[SP+1	1	IR	Instr	Instrucción - 1er palabra PC 2da PC 3ra PC												stru			
					,								+ •	alabra		palabra				codit			_
1	0000		FFEF	0000	_	19CE	3 0001	100	1 1	100 10	00	0001	0	0004	0002			M	OV I	₹7, [0x0	004]	
	Ejecuci	ón	R7=[0:	×0004]=	0×0)FE0												FI	ags				
2	0002				Т	B000	1011	000	0 0	000 00	00	0003	0	0005	0004			C	ALL	0x00	05		٦
	Ejecuci	ón	[0xFFE	$[F]=0\times0$	004	SP=	0×FFEE	E PC:	=0×	0005								FI	ags				
3	0005		FFEE	0004	Т	39A7	7						Г										٦
	Ejecuci	ón																FI	ags				
4																							٦
	Ejecuci	ón																FI	ags				
5																							
	Ejecuci	ón																FI	ags				7

Ejecución

			FEF	R0	R1			R3 0000	R4	R5	R6	R7	0	Z C	V	N 0
					-						1000		ا ل	- -	1.	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00 19CB	0004	B000	0005	0FE0	39A7	C000	0000	000	0 0000	0000	0000	0000	0000	000	0000
00	10 0000	0000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	0000	000	0000
_																
	PC	SP	[SP+1]	l IR	Inst	rucción -	- 1er pa	alabra	PC	2da	PC	3ra	PC		struc	
		-	[]							palabra		palabra		de	codifi	cada
1	0000	FFEF	0000	19CI	3 000	1 1001 1	100 10	00	0001	0004	0002			MOV	R7, [0	×0004]
	Ejecución	R7=[0	×0004]=0	0x0FE0										Flags		
2	0002			B000	0 101	1 0000 0	000 00	00	0003	0005	0004			CALL	0×000	5
	Ejecución	[0×FFE	F]=0×00	04 SP=	=0×FFE	E PC=0	×0005							Flags		
3	0005	FFEE	0004	39A	7 001	1 1001 1	010 01	11								
	Ejecución													Flags		
4																
	Ejecución															
5																

Flags

Va	alores	P	C S	SP	R0	R1	L I	R2	R3	R4		R5	R6	R7		Z	C	V	N	
in	iciales	00	000	FEF	000	00 00	00 (0000	0000	000	00	0000	0000	000	0	0	0	0	0	
		-0	+1	+2	+3	+4	+5	+6	+7	' H	-8	+9	+A	+B	+C	+	-D	+6		+F
00	00 19	CB	0004	B000	0005	0FE0	39A7	C00	000	0 00	000	0000	0000	0000	0000	00	000	000	10 C	0000
00	10 00	00	0000	0000	0000	0000	0000	000	0 000	0 00	000	0000	0000	0000	0000	00	000	000	0 C	0000
											_									
	PC		SP	ISP+1	1 IR	Inst	Instrucción - 1er palabra PC 2da PC 3ra PC												ción	
				1.	,							palabra		palabra	1				icada	
1	0000		FFEF	0000	190	B 000	1 1001	1100 1	1000	0001	. (0004	0002			M	OV F	7, [0	000×C	4]
	Ejecuc	ión	R7=[0)×0004]=	0x0FE0)										Fla	ags			
2	0002				B00	0 101	1 0000	0000	0000	0003	3 (2000	0004			CA	ALL (x00	05	
	Ejecuc	ión	[0×FFI	EF]=0×00	004 SP	=0×FFE	E PC=	0×0005	5							Fla	ags			
3	0005		FFEE	0004	39A	7 001	1 1001	1010 ()111	0006	j									
	Ejecuc	ión														Fla	ags			
4																				
	Ejecuc	ión														Fla	ags			
5																				
	Ejecuc	ión														Fla	ags			T

Ejecución

	alores iciales		FEF	R0 000	R1 0 00			R3 0000	R4 0000	R5 0000	R6 0000	R7 0 000	0	Z C 0	-	N 0
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00 19CI	3 0004	B000	0005	0FE0	39A7	C000	0000	000	0 0000	0000	0000	0000	0000	0000	0000
00	10 0000	0000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	0000	0000	0000
	PC	SP	[SP+1]	IR	Insti	rucción -	- 1er pa	labra	PC	2da palabra	PC	3ra palabra	PC		strucció codifica	
1	0000	FFEF	0000	19CE	3 000	1 1001 1	100 100	00	0001	0004	0002			MOV	R7, [0×0	0004]
	Ejecució	n R7=[0	×0004]=0	x0FE0										Flags	Ť	П
2	0002			B000	101	1 0000 0	000 000	00	0003	0005	0004			CALL	0×0005	
	Ejecució	n [0xFFE	F]=0x000	04 SP=	0×FFEI	E PC=0	×0005							Flags		
3	0005	FFEE	0004	39A7	001	1 1001 1	.010 011	11	0006					SUB R	6, R7	
	Ejecució	n												Flags		
4	-															
	Ejecució	n												Flags		
5								- 1								

Flags

V	alores	P	CS	SP	R0	R1	.	R2	R3	R4	R5	R6	R7		Z	С	V	N
in	iciales	00	000 F	FEF	000	00 00	00 (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
00	00 19	CB	0004	B000	0005	0FE0	39A7	C00	000	000	0000	0000	0000	0000	00	000	0000	0000
00	10 00	00	0000	0000	0000	0000	0000	0000	000	000	0 0000	0000	0000	0000	00	000	0000	0000
_																		
	PC		SP	[SP+1]	l IR	Inst	rucciór			struco								
				[0. 12]	·		acc.o.	. 20. p	Jaiabia	PC	palabra	PC	palabra	PC			codific	
1	0000		FFEF	0000	19C	B 000	1 1001	1100 1	000	0001	0004	0002			M	OV F	R7, [0	×0004]
	Ejecuc	ión	R7=[0	×0004]=	0×0FE0										Fla	ags		
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005	0004			CA	ALL (000×0	5
	Ejecuc	ión	[0xFFE	F]=0x00	004 SP:	=0xFFE	E PC=	0×0005							Fla	ags		
3	0005		FFEE	0004	39A	7 001	1 1001	1010 0	111	0006					SL	JB R	6, R7	
	Ejecuc	ión	R6=0×	(0000-0x	OFEO=)xF020									Fla	ags		
4																		
	Ejecuc	ión																\Box
5																		
	Ejecuc	ión													Fla	ags		\Box

	alores P		SP	R0	R1			23	R4	R5	R6	R7		Z C	V		N	
in	iciales 00	000 F	FEF	000	00 00	00 00	000	0000	0000	0000	F020	000	0 [0 0	0		0	
_								_				_	_			_		_
	+0	+1	+2	+3	+4	+5	+6	+7	+8		+A	+B	+C	+D	+	_	+	
00	00 19CB	0004	B000	0005	0FE0	39A7	C000	0000	000	0000	0000	0000	0000	0000	00	00	000	00
00	10 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	00	00	000	00
																	_	_
	PC.	SP	ICD 11	l IR	Lucia		4	Labora	PC.	2da	PC.	3ra	PC	lr	stru	cció	n	\neg
	PC	SP	[SP+1]	I IK	Insti	rucción -	- 1er pa	iabra	PC	palabra	PC	palabra	PC	de	codi	fica	da	
1	0000	FFEF	0000	19C	B 000	1 1001 1	100 100	00	0001	0004	0002			MOV	R7,	0x0	004]	П
	Ejecución	R7=[0	×0004]=	0x0FE0									•	Flags			T	П
2	0002			B00	0 101	1 0000 0	000 000	00	0003	0005	0004			CALL	0×00	005		ヿ
	Ejecución	0×FFE	F]=0×00	004 SP=	=0×FFEI	E PC=0	×0005							Flags			П	П
3	0005	FFEE	0004	39A	7 001	1 1001 1	010 01	11	0006					SUB F	6, F	7		ᅱ
	Ejecución	R6=0×	(0000-0x	OFE0=0)xF020									Flags	0	1	0	1
4																		ᅱ
	Ejecución																П	\neg
5																		ヿ
	Ejecución													Flags				

	alores Piciales 00		FEF	R0 000	R1			R3 0000	R4 0000	R5 0000	R6	R7	0	Z C 0	0	0	N)	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E		+F	=]
00	00 19CB	0004	B000	0005	0FE0	39A7	C000	0000	0000	0000	0000	0000	0000	0000	000	0	000	0
00	10 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0	000	0
		2da 3ra																_
	PC	SP	SP [SP+1] IR Instrucción - 1er palabra PC 2da palabra PC 3ra palabra													ciór cad	-	
1	0000	FFEF	0000	19C	B 000:	1 1001 1	100 100	00	0001	0004	0002			MOV	R7, [0	×00	004]	٦
	Ejecución	R7=[0	×0004]=0	x0FE0										Flags		Т	\top	\neg
2	0002			B00	0 101	1 0000 0	000 000	00	0003	0005	0004			CALL	0×000	15		\neg
	Ejecución	[0×FFE	F]=0×00	04 SP=	=0×FFEI	E PC=0	×0005							Flags		Т	Т	П
3	0005	FFEE	0004	39A	7 001	1 1001 1	.010 011	11	0006					SUB R	6, R7			\neg
	Ejecución	R6=0×	0000-0×0)FE0=0)×F020									Flags	0	1	0	1
4	0006																	
	Ejecución													Flags		T		
5																		
	Ejecución													Flags		Т	\top	

	alores P		FEF	R0 000	R1 0 000			R3 0000	R4 0000	R5 0000	R6	R7	0	Z C 0 0	V 0	ļ.,))	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	:	+1	F
00		0004		0005	0FE0	39A7	C000	0000			0000	0000	0000	0000	000	_	000	
00		0000		0000	0000	0000	0000	0000			0000	0000	0000	0000	000	-	000	
00	10 0000	0000	0000	0000	0000	0000	0000	0000	1 000	7 0000	0000	0000	0000	0000	000	U	000	0
	PC.	SP	[SP+1]	IR	Inati		1	مساما	PC.	2da	PC	3ra	PC	Ir	struc	ciói	1	
	PC	3P	[5P+1]	l ik	Instr	rucción -	- 1er pa	iabra	PC	palabra	PC	palabra	PC	de	codif	cac	la	
1	0000	FFEF	0000	19CF	3 0001	1 1001 1	100 100	00	0001	0004	0002			MOV	R7, [0)×0	004]	П
	Ejecución	R7=[0:	×0004]=0	×0FE0										Flags			Ť	\neg
2	0002			B000	0 1011	1 0000 0	000 000	00	0003	0005	0004			CALL	0×000)5		コ
	Ejecución	[0xFFE	F]=0×00	04 SP=	=0×FFE	E PC=0	×0005							Flags				\neg
3	0005	FFEE	0004	39A7	7 0011	1 1001 1	010 01	11	0006					SUB F	6, R	7		コ
	Ejecución	R6=0×	0000-0×0	FE0=0	xF020								_	Flags	0	1	0	1
4	0006			C000)								Т					ヿ
	Ejecución												_	Flags		Т		\neg
5													Т					ヿ
	Ejecución													Flags		Т		

	alores P		FEF	R0 000	R1 0 000	00 00		R3 0000	R4 0000	R5 0000	R6	R7	0	Z C 0	V 0	0))	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E		+F	=
00																_		_
00		0004		0005	0FE0	39A7	C000	0000			0000	0000	0000	0000	000	-	000	
00	10 0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	000	0	000)0
			SP [SP+1] IR Instrucción - 1er palabra PC 2da PC 3ra F															_
	PC	SP	[SP+1]	l IR	Insti	ucción -	. 1er na	lahra	PC.	2da	PC		PC		struc			
		J.	[5, 11]	"`	111361	accion	ICI Pa	labia	١ ٠	palabra	' ~	palabra	' `	de	codifi	cad	a	
1	0000	FFEF	0000	19CE	3 0001	1001 1	100 100	00	0001	0004	0002			MOV	R7, [0)×00	004]	
	Ejecución	R7=[0:	×0004]=0	×0FE0										Flags		Т		
2	0002			B000	1011	0000 0	000 000	00	0003	0005	0004			CALL	0×000)5		コ
	Ejecución	[0xFFE	F]=0x00	04 SP=	=0xFFE	PC=0	×0005							Flags		Т		
3	0005	FFEE	0004	39A7	7 0011	1001 1	.010 01	11	0006					SUB F	6, R7	,		ヿ
	Ejecución	R6=0×	0000-0x0	FE0=0	×F020									Flags	0	1	0	1
4	0006			C000	1100	0000 0	000 000	00										ヿ
	Ejecución													Flags				\neg
5																		7
	Ejecución													Flags			Т	٦

V	alores	P	C S	SP	R0	R1	.	R2	R3	R4	F	R5	R6	R7		Z	С	V		N	
in	iciales	00	000 F	FEF	000	00 00	00	0000	0000	0000) (0000	F020	000	0	0	0	0	1	0	
		_													_ `					_	
		-0	+1	+2	+3	+4	+5	+6	5 +7	+8	В	+9	+A	+B	+C	+1	D	+E	Ξ	+	F
00	00 19	CB	0004	B000	0005	0FE0	39A7	7 C00	000 000	000	00 (0000	0000	0000	0000	000	00	000	0	000	00
00	10 00	00	0000	0000	0000	0000	0000	000	0 000	000	00 (0000	0000	0000	0000	000	00	000	0	000	00
											•					_					
_																					
	PC		SP	[SP+1]	l IR	Inst			truc												
			٥.	[0. 12]	·			. 10.	parabra	PC	pal	labra	PC	palabra	PC		dec				
1	0000		FFEF	0000	19C	B 000	1 1001	1100	1000	0001	000	04	0002			MC	V R	7, [0x0	004]	
	Ejecuc	ión	R7=[0	×0004]=0	0×0FE0											Fla	gs				
2	0002				B00	0 101	1 0000	0000 (0000	0003	000	05	0004			CA	LL 0	×00	05		
	Ejecuc	ión	[0×FFE	F]=0×00	004 SP	=0×FFE	E PC=	0×0005	5							Fla	gs			П	
3	0005		FFEE	0004	39A	7 001	1 1001	1010 (0111	0006						SUI	B R6	, R	7		
	Ejecuc	ión	R6=0>	<0000-0x0	FE0=0	0xF020										Fla	gs	0	1	0	1
4	0006				C00	0 110	0000	0000	0000	0007											
	Ejecuc	ión														Fla	gs				
5																					\neg
	Ejecuc	ión														Flag	gs				

Va	alores	es 0000 FFE		P	R0	R1	- 1	R2	R3	R4	R5	R6	R7		Z	C	V		N
in	iciales	0000) F	FEF	000	00 00	00 (0000	0000	0000	0000	F020	000	0	0	0	0	T	0
																		二	
	+	-	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+1	Ξ	+F
00	00 190	CB 0	004	B000	0005	0FE0	39A7	C00	0000	000	0000	0000	0000	0000	00	00	000	10	0000
00	10 000	0 0	000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	00	00	000	0	0000
			SP [SP+1] IR Instrucción - 1er palabra PC 2da PC 3ra																
	PC		SP	[SP+1]	l IR	Inst	rucciór	1 - 1er r	alabra	PC		PC.		PC			struc		
	. 0		<u> </u>	[0. 12]			ucc.o.	. 20. p	, a labi a		palabra		palabra			dec	odif	icad	da
1	0000	F	FEF	0000	19C	B 000	1 1001	1100 1	000	0001	0004	0002			MC	OV F	7, [0x0	004]
	Ejecuci	ón R	7=[0:	×0004]=0	x0FE0										Fla	gs		П	
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005	0004			CA	LL C)×00	05	
	Ejecuci	ón [C)xFFE	F]=0x00	04 SP	=0xFFE	E PC=	0×0005							Fla	gs			
3	0005	F	FEE	0004	39A	7 001	1 1001	1010 0	111	0006					SU	B R	5, R	7	
	Ejecuci	ón R	6=0x	0000-0×0	FE0=()xF020									Fla	gs	0	1	0 1
4	0006				C00	0 110	0000	0000 0	000	0007					RE	Т			
	Ejecuci	ón													Fla	gs			
5																			
	Ejecuci	ón													Fla	gs		\Box	

Va	lores	PC	S	P	R0	R1	. F	22	R3	R4	R5	R6	R7		Z	C	V		N	
ini	ciales	000	0 F	FEF	000	00 00	00 0	000	0000	0000	0000	F020	000	0	0	0	0	1	0	
							'												_	
	+	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+1	Ε	+	F
000	00 190	CB C	0004	B000	0005	0FE0	39A7	C00	0 0000	000	0000	0000	0000	0000	00	00	000	00	00	00
003	10 000	00 0	0000	0000	0000	0000	0000	0000	0000	000	0000	0000	0000	0000	00	00	000	00	00	00
																				_
			SP [SP+1] IR Instrucción - 1er palabra PC 2da PC 3ra																	_
	PC		SP	[SP+1]	l IR	Inst	rucción	- 1er r	palabra	PC		PC		PC			struc			
			-	[0. , -]							palabra		palabra	1			odif			
1	0000	F	FEF	0000	19C	B 000	1 1001	0001	0004	0002			M	OV F	7, [0x0	004]			
	Ejecuci	ón F	R7=[0:	×0004]=0	×0FE0										Fla	gs			\neg	
2	0002				B00	0 101	1 0000	0000 0	0000	0003	0005	0004			CA	LL C)×00	05		
	Ejecuci	ón [0×FFE	F]=0x00	04 SP=	=0×FFE	E PC=0	0×0005							Fla	gs			\Box	
3	0005	F	FEE	0004	39A	7 001	1 1001	1010 0	111	0006					SU	B R	5, R	7		
	Ejecuci	ón F	₹6=0x	0000-0×0	FE0=0)×F020									Fla	gs	0	1	0	1
4	0006				C00	0 110	0 0000	0000 0	0000	0007					RE	Т				\neg
	Ejecuci	ón F	PC=[0	xFFEF]=	0x0004	4 SP=0:	<ffef< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Fla</td><td>gs</td><td></td><td></td><td>\Box</td><td></td></ffef<>								Fla	gs			\Box	
5																				
	Ejecuci	ón													Fla	gs			\Box	

Va	alores	PC	S	P	R0	R1	F	R2	R3	R4	R5	R6	R7		Z	C	V	Ν	
ini	iciales	00	00 F	FEF	000	00 00	00 C	0000	0000	0000	0000	F020	000	0	0	0	0	0	
	+	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	Ι.Λ	+B	+C	+1		+E		+F
		-		-								+A							
00	00 190	св	0004	B000	0005	0FE0	39A7	C000	000 0	000 0	0000	0000	0000	0000	000	00	0000)	0000
00	10 000	00	0000	0000	0000	0000	0000	0000	000	000	0 0000	0000	0000	0000	000	00	0000) [0000
									. ,										
	PC		SP	[SP+1]	IR	Inst	rucción	- 1er p	alabra	PC	2da palabra	PC	3ra palabra	PC		deco	rucc		
1	0000		FFEF	0000	19C	B 000	1 1001	1100 1	000	0001	0004	0002			MC	V R7	', [0:	×00	04]
	Ejecuci	ión	R7=[0:	×0004]=0	0×0FE0										Fla	gs		Т	
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005	0004			CA	LL 0>	:000	5	
	Ejecuci	ión	[0xFFE	F]=0x00	04 SP:	=0xFFE	E PC=	0×0005							Fla	gs		Т	\top
3	0005		FFEE	0004	39A	7 001	1 1001	1010 0	111	0006					SUI	B R6	R7		
	Ejecuci	ión	R6=0x	0000-0×0	FE0=0)×F020									Fla	gs () 1	П	0 1
4	0006				C00	0 110	0000	0000 0	000	0007					RE	Г			
	Ejecuci	ión	PC=[0xFFEF]=0x0004 SP=0xFFEF												Fla	gs		Т	\top
5	0004																		
	Eiecuci	ión													Fla	gs		Т	

V	alores	P	CS	SP	R0	R1	F	R2	R3	R4	R5	R6	R7		Ζ	C	V		N	
in	iciales	00	000 F	FEF	000	00 00	00 0	0000	0000	0000	0000	F020	000	0	0	0	0	1	0	
																			_	
	-	⊢0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+1	E	+F	F
00	00 19	CB	0004	B000	0005	0FE0	39A7	C000	0000	000	0000	0000	0000	0000	00	00	000	00	000	00
00	10 00	00	0000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	00	00	000	00	000	00
																				_
_				SP [SP+1] IR Instrucción - 1er palabra PC 2da PC 3ra																
	PC	.	SP	[SP+1]	l IR	Inst	rucción	- 1er n	alahra	PC		PC		PC			struc			
				[51 11]	·		decion	ICI P	alabia		palabra		palabra				odif			
1	0000		FFEF 0000 19CB 0001 1001 1100 1000 0001 0004 0002											M	OV F	7, [0x0	004]		
	Ejecu	ión	R7=[0	×0004]=	0x0FE0										Fla	igs				
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005	0004			CA	LL ()×00	05		
	Ejecud	ión	[0×FFE	F]=0x00	004 SP=	=0×FFE	E PC=	0×0005							Fla	igs			П	
3	0005		FFEE	0004	39A	7 001	1 1001	1010 0	111	0006					SU	B R	5, R	7		
	Ejecud	ión	R6=0×	(0000-0x	OFE0=0)xF020									Fla	igs	0	1	0	1
4	0006				C00	0 110	0000	0000 0	000	0007					RE	Т				\neg
	Ejecud	ión	PC=[0	xFFEF]=	=0×0004	1 SP=0>	FFEF								Fla	igs			\Box	
5	0004		FFEF	0000																╗
	Ejecud	ión													Fla	igs			\Box	

Va	alores	PC	S	SP	R0	R1		R2	R3	R4	R5	R6	R7		Ζ	C	V		N	
in	iciales	000	0 F	FEF	000	00 00	00	0000	0000	0000	0000	F020	000	0	0	0	0	(0	
																	_	_		
	+	-	+1	+2	+3	+4	+5	+6	+7	+8	3 +9	+A	+B	+C	+	D	+6	Ξ	+F	:
00	00 190	CB (0004	B000	0005	0FE0	39A	7 C00	000	000	0000	0000	0000	0000	00	00	000	00	000	0
00	10 000	00 (0000	0000	0000	0000	0000	000	0 000	000	0000	0000	0000	0000	00	00	000	0	000	0
																				_
	PC		SP	[SP+1]	l IR	Inst	rucció	n - 1er ı	nalahra	PC	2da	PC	3ra	PC			struc			
	. 0		٠.	[5, 12]				. 20.	parabra		palabra		palabra			dec	codif	icac	da	
1	0000	F	FFEF	0000	19C	B 000	1 1001	. 1100 1	1000	0001	0004	0002			M	OV F	ί7, [(0x0	004]	
	Ejecuci	ón F	R7=[0:	×0004]=0	0×0FE0)									Fla	gs		П	П	
2	0002				B00	0 101	1 0000	0000 0	0000	0003	0005	0004			CA	LL C)×00	05		
	Ejecuci	ón [0xFFE	F]=0x00	04 SP	=0×FFE	E PC=	0×0005	5						Fla	gs		П		
3	0005	F	FFEE	0004	39A	7 001	1 1001	1010 0)111	0006					SU	B R	5, R	7		Т
	Ejecuci	ón F	R6=0×	0000-0x0	FE0=	0xF020									Fla	gs	0	1	0	1
4	0006				C00	0 110	0 0000	0000 0	0000	0007					RE	Т				
	Ejecuci	ón F	PC=[0	xFFEF]=	0×000	4 SP=0	FFEF							Fla	gs			П		
5	0004	F	FFEF	0000	0FE	.0														
	Ejecuci	ón													Fla	gs		П		_

Va	alores	PC	: S	SP	R0	R1	. F	R2	R3	R4	R5	R6	R7		Z	C	V		N
in	iciales	000	00 F	FEF	000	00 00	00 (0000	0000	0000	0000	F020	000	0	0	0	0	T	0
																		二	
	+	-	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+1	Ξ	+F
00	00 190	CB	0004	B000	0005	0FE0	39A7	C000	0000	000	0000	0000	0000	0000	00	00	000)0	0000
00	10 000	00	0000	0000	0000	0000	0000	0000	0000	000	0 0000	0000	0000	0000	00	00	000	0	0000
	PC		SP	[SP+1]	l IR	Inst	rucción	- 1er p	alahra	PC	2da	PC.	3ra	PC			struc		
	. 0		٥.	[5, 12]			decion	- 10. p	diabia		palabra		palabra			dec	odif	icad	da
1	0000		FFEF	0000	19C	B 000	1 1001	1100 1	000	0001	0004	0002			MC	OV F	7, [0x0	004]
	Ejecuci	ón	R7=[0:	×0004]=0	0×0FE0										Fla	gs		П	
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005	0004			CA	LL ()×00	05	
	Ejecuci	ón	[0×FFE	F]=0x00	04 SP:	=0xFFE	E PC=	0×0005							Fla	gs			
3	0005		FFEE	0004	39A	7 001	1 1001	1010 0	111	0006					SU	B R	5, R	7	
	Ejecuci	ón	R6=0x	0000-0x0	FE0=0	0xF020									Fla	gs	0	1	0 1
4	0006				C00	0 110	0000	0000 0	000	0007					RE	Т			
	Ejecuci	ón	PC=[0	xFFEF]=	0×000	4 SP=0>	FFEF								Fla	gs			\Box
5	0004		FFEF	0000	0FE	0 000	0 1111	1110 0	000										
	Ejecuci	ón													Fla	gs		\Box	

Va	alores	P	C S	SP	R0	R1	. 1	R2	R3	R4	R5	R6	R7		Z	C	V		N	
ini	iciales	00	000 F	FEF	000	00 00	00 (0000	0000	0000	0000	F020	000	0	0	0	0	1	0	
						· ·													_	
		-0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+	Ε	+	·F
00	00 19	CB	0004	B000	0005	0FE0	39A7	C00	000	000	0000	0000	0000	0000	00	00	000	00	00	00
00	10 00	00	0000	0000	0000	0000	0000	000	000	000	0000	0000	0000	0000	00	00	000	00	00	00
		_												_			_	.,		
	PC		SP	[SP+1] IR	Inst	rucciór	- 1er i	palabra	PC	2da	PC	3ra	PC			truc			
\Box				,	1						palabra		palabra				odit			
1	0000		FFEF	0000	190		1 1001	1100 1	.000	0001	0004	0002			MC)V R	7, [0x0	004	
	Ejecuc	ión	R7=[0	×0004]=	0×0FE)									Fla	gs				
2	0002				B00	0 101	1 0000	0000	0000	0003	0005	0004			CA	LL 0	x00	05		
	Ejecuc	ión	[0×FFI	F]=0×0	004 SP	=0×FFE	E PC=	0×0005	,						Fla	gs				
3	0005		FFEE	0004	394	7 001	1 1001	1010 ()111	0006					SU	B R	5, R	7		
	Ejecuc	ión	R6=0:	<0000-0×	0FE0=	0×F020									Fla	gs	0	1	0	1
4	0006				C00	0 110	0 0000	0000	0000	0007					RE	T				
	Ejecuc	ión	PC=[0	xFFEF]=	=0×000	4 SP=0:	FFEF								Fla	gs				
5	0004		FFEF	0000	0FE	000	0 1111	1110 (0000	0005										
	Ejecuc	ión													Fla	gs				

V	alores	P	C	SP	R0	R1	.	R2	R3	R4	F	R5	R6	R7		Ζ	C	V	1	N	
in	iciales	00	000 F	FEF	000	00 00	00 (0000	0000	0000) (0000	F020	000	0	0	0	0	- (0	
	+	-0	+1	+2	+3	+4	+5	+6	+7	+8	3	+9	+A	+B	+C	+	D	+1	E	+	F
00	00 19	CB	0004	B000	0005	0FE0	39A7	C00	0 000	000	00 (0000	0000	0000	0000	00	00	000	00	00	00
00	10 00	00	0000	0000	0000	0000	0000	000	000	000	00 (0000	0000	0000	0000	00	00	000	00	00	00
																					_
				SP [SP+1] IR Instrucción - 1er palabra PC 2da PC 3ra																	
	PC		SP	ISP+1	ı lr	Inst	rucción	1 - 1er ı	palabra	PC	1		PC.		PC			stru			
					<u> </u>						<u> </u>			palabra	1			codit			
1	0000		FFEF	FFEF 0000 19CB 0001 1001 1100 1000 0001 0004 0002												M	OV F	27, [0x0	004]	
	Ejecuc	ión	R7=[0	×0004]=	0×0FE0											Fla	igs				
2	0002				B00	0 101	1 0000	0000 0	0000	0003	000	05	0004			CA	LL C)×00	05		
	Ejecuc	ión	[0xFFE	F]=0x00	004 SP	=0xFFE	E PC=	0×0005								Fla	igs				
3	0005		FFEE	0004	39A	7 001	1 1001	1010 0	111	0006						SU	B R	6, R	7		
	Ejecuc	ión	R6=0×	(0000-0x	0FE0=	0xF020										Fla	igs	0	1	0	1
4	0006				C00	0 110	0000	0000 0	0000	0007						RE	T				
	Ejecuc	ión	PC=[0	C000 1100 0000 0000 0000 0007 PC=[0xFFEF]=0x0004 SP=0xFFEF													igs				
5	0004		FEE 0000 0FE0 0000 1111 1110 0000 0005													Ins	truc	ción	Inv	álida	3
	Ejecuc	ión														Fla	igs				

V	alores	P	CS	SP	R0	R1	F	R2	R3	R4	R5	R6	R7		Z	C	V	1	N	
in	iciales	00	000 F	FEF	000	00 00	00 0	0000	0000	0000	0000	F020	000	0	0	0	0		0	
	-	-0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+	D	+	E	+	F
00	00 19	CB	0004	B000	0005	0FE0	39A7	C000	0000	000	0000	0000	0000	0000	00	00	000	00	000	00
00	10 00	00	0000	0000	0000	0000	0000	0000	0000	000	0000	0000	0000	0000	00	00	000	00	000	00
																				_
_																				
	PC		SP	[SP+1	1 IR	Inst	rucción	- 1er p	alahra	PC	2da	PC	3ra	PC			stru			
				[0. 12	<u>' </u>			20. p	a.ab.a		palabra		palabra				odi			
1	0000		FFEF	0000	19C	B 000	1 1001	1100 1	000	0001	0004	0002			M	OV F	27, [0×0	004]	
	Ejecuc	ión	R7=[0	×0004]=	0×0FE0										Fla	igs				
2	0002				B00	0 101	1 0000	0000 0	000	0003	0005	0004			CA	LL C)×00	05		\neg
	Ejecuc	ión	[0×FFE	F]=0x00	004 SP	=0×FFE	E PC=	0×0005							Fla	igs				
3	0005		FFEE	0004	39A	7 001	1 1001	1010 0	111	0006					SL	B R	6, R	7		
	Ejecuc	ión	R6=0×	(0000-0x	0FE0=0)xF020									Fla	igs	0	1	0	1
4	0006				C00	0 110	0000	0000 0	000	0007					RE	T				\neg
	Ejecuc	ión	PC=[0xFFEF]=0x0004 SP=0xFFEF													igs				\neg
5	0004		FFEF 0000 0FE0 0000 1111 1110 0000 0005												Ins	truc	ción	Inv	álida	3
	Ejecuc	ión	FIN D	E LA EJ	ECUCIO	ĺΝ									Fla	igs				

Ejercicio 2 - Solución

]	Plani	illa de	Se	egui:	mier	nto	F020								
			PC	SP		RC		R1	R2	P	13	R4	R5	R6	R7		Z	С	V	N		
٧	alores i	niciales:	0000	FFEF	:	000	0 0	000	0000	00	000	0000	0000	0000	0000		0	0	0	0	1	
	[+0	+1	+2	+3	+4	+5	+6		+7	+8	+9	+A	+B	+C	+E	-	Ε	+	F	1
Mer	noria:	0000	19C8	0004	B000	0005	OFEO	39A	7 C00	0 (0000	000	0000	0000	0000	0000	000	00 00	000	00	00	
		0001	0000	0000	0000	0000	0000	0000	000	0 (0000	000	0000	0000	0000	0000	000	00 00	000	00	00	
	PC		SP	[SP+1]	IR	Ins		in - 1° en bits	palab	ra	P	С	2 ^{do} palabra	PC	3 ^{cro} palabra	Pi	2		stru codi			_
1	0000 FFEF 0000 19C8 0001 1001 1100 1000 0001 0004 0002													MOV	R7,	[0x0	004	1]				
	Ejecución R7 = [0x0004] = 0x0FE0														Flags ¹	Z	С	٧	N			
2	Boo Boo													CALL	0x0	005						
	Е	jecució	n	[0xFFEF]	= 0x00	04 9	SP = 0x	FFEE	PC	= 0x	:0005							Flags ¹	Z	С	٧	N
3	000	5 FF	EE	0004	39A7	001	1 100	1 101	0 011	1	00	06						SUBF	R6, R	7		
	Е	jecució	n	R6 = 0x0	000 - 0	oFE0 =	0xF02	0										Flags ¹	z 0	c 1	v O	1 N
4	000	5			C000	110	000	000	0000		00	07						RET				
	Е	jecució	n	PC = [0x	FFEF] =	0x0004	SF	= 0xF	FEF									Flags ¹	Z	С	٧	м
5	0004 FFEF 0000 0FE0 0000 1111 1110 0000 0005												Instru	ıcció	n In	váli	la					
	Е	jecució	n	FIN DE LA	EJECUC	IÓN.												Flags ¹	Z	С	٧	N

Ejercicio 2 - Observaciones

 $\ensuremath{ \begin{subarray}{c} \ensuremath{ \ensuremath{ \ensuremath{ Q} \ensuremath{ u\acutee} \ensuremath{ \ensuremath{ e} \ensuremath{ \ensuremath{ c} \ensuremath{ \ensuremath{ \ensuremath{ Q} \ensuremath{ \ensuremath{ u\acutee} \ensuremath{

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.

Programación en Asembler

Acordando pautas...

- Se utilizará el lenguaje Assembler de la máquina 'Orga 1'
- Es un lenguaje ensamblador particular de esta arquitectura, que es una simplificación de una arquitectura Intel x86
- Posee instrucciones con 2 operandos, 1 operando, 0 operandos y saltos
- Posee diferentes modos de direccionamiento

Enunciado Ejercicio Nro 1

Estaría muy bueno manejar algunas operaciones en Assembler. Y la división tiene sus vericuetos....Por lo tanto vamos a escribir un programa que calcule la división entera entre dos enteros sin signo de 16 bits.

- R1 contiene la dirección de memoria donde se aloja el dividendo.
- R2 contiene la dirección de memoria donde se aloja el divisor.
- R3 debe ser el registro en el que se devuelva el cociente (el resultado de la división).

Como somos muy buenos...En caso de que el divisor sea 0, basta con que se retorne 0.

Pseudocódigo

```
\begin{tabular}{ll} resultado &= 0 \\ si & divisor == 0 \\ entonces \\ & listo \\ sino \\ & mientras \\ dividendo >= divisor \\ & hacer \\ & dividendo = dividendo - divisor \\ & resultado = resultado + 1 \\ & fin \\ & mientras \\ fin \\ si \\ \end{tabular}
```

Assembler!

```
inicio: MOV R3, 0x0000; R3 = 0
       MOV R4, [R1] ; R4 = dividendo
       MOV R5, [R2] ; R5 = divisor
       CMP R5. 0x00000 : divisor == 0?
       JE fin
ciclo: CMP R4, R5 ; dividendo < divisor?
       JCS fin ; uso JCS en vez de JL porque son enteros sin signo
       SUB R4, R5 ; R4 = R4-R5
       ADD R3, 0x0001; R3 = R3+1
       JMP ciclo
    (...)
fin:
```

Tarea: Retornar en R0 el resto de la division!

Enunciado Ejercicio Nro 2

Los docentes del turno tarde precisamos ingresos extras... compramos una pareja de conejos y queremos saber cuántos se podrían reproducir en un año a partir de la pareja inicial, teniendo en cuenta que de forma natural tienen una pareja en un mes, y que a partir del segundo mes se empiezan a reproducir.



Sucesión de Fibonacci

$$f_n = f_{n-1} + f_{n-2} \text{ con } f_0 = 1 \text{ y } f_1 = 1$$

Pseudocódigo

```
meses = 10; //(12-2) me da el término 12 de la suc.
anteUltMes = 1;
ultMes = 1;
cantParejas = ultMes;
mientras meses > 0
   cantParejas = cantParejas + anteUltMes;
   anteUltMes = ultMes;
   ultMes = cantParejas;
   meses = meses - 1;
fin mientras
```

Assembler!

Resumen de la clase

Hoy repasamos los siguiente conceptos:

- Tamaño de memoria
- Dirección de memoria
- Unidad direccionable
- Instrucción
- Codificación de una instrucción

Resumen de la clase

Hoy también 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

Resumen de la clase

Por último:

- Vimos como realizar pequeños programas en Assembler utilizando la Arquitectura 'Orga 1'
- Es importante utilizar pseudocódigo para resolver el problema y verificar que el algoritmo funcione
- Pueden utilizar un pseudocódigo informal, que pueda comprenderse y no sea ambiguo
- Una vez verificada la solución, realizar el código en el lenguaje
 Ensamblador de la materia

¿Preguntas?

¿Preguntas?



Avisos

Martes 2/10: Taller de Ciclo de Instrucción .

IMPORTANTE: Traer Cartilla Orga 1 y planillas de Seguimiento

IMPORTANTE 2: Hacer la guia de ejercicios!!!!

