

Programación en Ensamblador Documentación

1. Transparencias del tema (web)
2. Descripción del emulador 88110 (web+publicaciones)
3. Subrutinas: paso de parámetros y marco de pila (web+publicaciones)
4. Enunciados de problemas (web+publicaciones)

http://www.datsi.fi.upm.es/docencia/Estructura_09

Fundamentos de los computadores
 Pedro de Miguel, Paraninfo/Thomson-2006 (capítulo 13)

Estructura de computadores: problemas y soluciones
 García Clemente y otros, RAMA-2000 (capítulo 2)

Estructura de computadores: problemas resueltos
 García Clemente y otros, RAMA-2006 (capítulo 3)

Solución de problemas:
<http://www.datsi.fi.upm.es/88110>

Dpto. Arquitectura y Tecnología de Sistemas Informáticos, Universidad Politécnica de Madrid

Lenguaje Máquina

- Programa está compuesto por datos e instrucciones almacenados en memoria
- **Instrucción máquina:** Es la función básica elemental que puede ejecutar un computador.
- Son cadenas de 1 y 0 (almacenadas en binario) y particulares de cada computador
- **Propiedades:**
 - Realizan una única y sencilla función
 - Tienen un número fijo de operandos
 - Autocontentidas: Contienen todo lo necesario para su ejecución (operación, operandos, dir. Resultado y dir. Sig instrucción)

Cada instrucción realiza una sola función de suma, resta...

Dpto. Arquitectura y Tecnología de Sistemas Informáticos, Universidad Politécnica de Madrid

Juego de instrucciones

- Conjunto de instrucciones que ejecuta directamente el computador.
- La codificación de las instrucciones deben encajar en pocos formatos. *para poder usarlos distintos*
- Formato de instrucción: Representación de una instrucción y especificación de cada campo:
 - **Código de operación** - *los operandos se designan*
 - **Operandos (direcciones)** - *los operandos de posición del computador (puede llevar + de los)*
 - **Operador** - *quiere ser subrutinado*
 - **Modos de direccionamiento** - *los operandos se designan*

*Le 0 binario - a solo contiene el cod de operación
 1 binario - instrucción que es de campo y datos
 2 binario - 1 binario y campo y el 0 binario
 3 binario - 1 binario campo y datos*

Dpto. Arquitectura y Tecnología de Sistemas Informáticos, Universidad Politécnica de Madrid



**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70**

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, starburst-like background. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Tipos de datos

- Tipos de datos que maneja una instrucción:
 - Palabras: Es el tamaño privilegiado del computador (4 bytes) *32 bits*
 - Medias palabras (2 bytes)
 - Bytes: Cadenas de caracteres.
- Acceso a memoria:
 - Direccionable a nivel de palabra. Cada palabra tiene una dirección.
 - Direccionable a nivel de byte. Cada byte de memoria tiene una dirección.
 - Dos palabras consecutivas están separadas por el tamaño en bytes de la palabra.
 - Es el utilizado habitualmente.

Tipos de datos

*standard IEEE C94
función ensamblador*

- Alineamiento a palabra. La dirección para el acceso a palabra debe ser múltiplo del tamaño de la misma.
- Ordenación de los bytes de una palabra en memoria.
 - Little-Endian: Byte menos significativo de una palabra en la dirección menos significativa.
 - Big-Endian: Byte menos significativo de una palabra en la dirección más significativa.

Palabra: 0x10203040

Little-Endian

100	101	102	103
40	30	20	10

- significativa

Big-Endian

100	101	102	103
10	20	30	40

- significativa

12.00

standard IEEE

• Instrucciones ensamblador

- código numérico
- nombres simbólicos

*↕ equivalencia
↕ directa*

*de código en programa
ensamblador*

p. fuente (simbólico) → prog. objeto (ejecutable)

• Instrucciones máquina (Binario)

ADD 1

*alternativa
instrucción*

Modos de direccionamiento

- Forma en la que se accede a una instrucción o dato.
- OBJETO:** Instrucción o dato al que se desea acceder
- DIRECCIÓN:** Lugar en el que reside el objeto. Puede estar almacenado en:
 - La instrucción.
 - Registro
 - Memoria

Direccionamiento Inmediato

Se representa con #4; ej: 44

- El objeto está contenido en la propia instrucción.

Instrucción

Objeto

- ADD .R1, #4 R1 ← R1 + 4

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, starburst-like background that tapers to the right. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Direccionamiento Directo Absoluto a Registro

• El objeto del direccionamiento está contenido en un registro. La instrucción contiene el registro que contiene el objeto del direccionamiento.

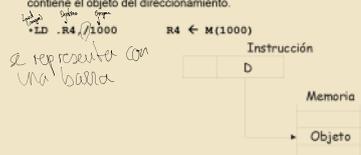
• **ADD R4, R5** $R4 \leftarrow R4 + R5$



Direccionamiento Directo Absoluto a Memoria

• El objeto del direccionamiento está contenido en una dirección de memoria. La instrucción contiene la dirección completa de memoria que contiene el objeto del direccionamiento.

• **LD R4, 1000** $R4 \leftarrow M(1000)$



Direccionamientos Relativos

• El objeto del direccionamiento está contenido en una dirección de memoria. La instrucción contiene la dirección especificada en *partes*.

• Dependiendo de cómo se especifique la dirección:

- Relativo a registro base
- Relativo a PC
- Relativo a registro índice

Direccionamiento Relativo a Registro Base

• La dirección de memoria viene especificada en dos partes:

- **Registro Base:** Registro de propósito específico o general que contiene una dirección a memoria.
- **Desplazamiento:** Valor entero con signo.

• La dirección efectiva se calcula:

$$\text{Dir_Efectiva} = \text{Registro_Base} + \text{Desplazamiento}$$

- Si el registro está entre corchetes contiene una dirección de memoria

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, dark blue font. The text is set against a light blue, abstract background that resembles a stylized 'C' or a wave. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

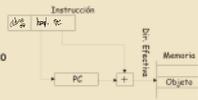
**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Direccionamiento Relativo a PC

- Es un direccionamiento relativo a registro base en el que el registro base es el PC (contador de programa)
- El objeto de este direccionamiento suele ser direccionar instrucciones.
- Permite alcanzar instrucciones "cercanas" a la que se está ejecutando.

Ejecución de saltos "cortos"



BR \$10
Suma 10 al PC



IEEE 194

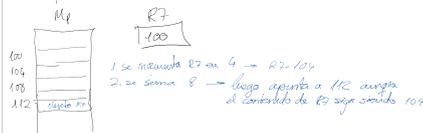
- El desplazamiento se suma al PC ya incrementado
- ej: BR \$9

Direccionamiento Relativo a Registro Índice

- Es un direccionamiento relativo a registro base en el que el registro base se modifica:
 - Preincremento: $ID .R1, \#8[+, R7] \quad R7 \leftarrow R7+4 \quad R1 \leftarrow MEM(R7+8)$
 - Postdecremento: $ID .R1, \#8[-, R7] \quad R7 \leftarrow R7-4 \quad R1 \leftarrow MEM(R7+8)$
 - Postincremento: $ID .R1, \#8[., R7++] \quad R1 \leftarrow MEM(R7+8) \quad R7 \leftarrow R7+4$
 - Postdecremento: $ID .R1, \#8[., R7--] \quad R1 \leftarrow MEM(R7+8) \quad R7 \leftarrow R7-4$
- El tamaño del incremento/decremento es igual al tamaño del objeto transferido
- Útil para recorrer vectores y matrices

1. se recorren los preincrementos o decrementos
2. se calcula la dirección
3. se calculan los preincrementos o decrementos

ej. palabra de 32 bit con redondeamiento a mitad de byte
 por eso la dist entre palabra y palabra varía 4 o 8
 útil para recorrer vectores o matrices.



Direccionamiento Indirecto a Registro

- La instrucción contiene la dirección donde está contenida la dirección donde se almacena el objeto.

a Registro

- La instrucción contiene la especificación del registro que contiene la dirección de memoria donde está almacenado el objeto

$ID .R1, [R4]$
 $R1 \leftarrow MEM(R4)$



la instrucción contiene la dirección de la dirección del objeto (se podría redirigir n veces).

si [R4] cada estado es un nivel de indirecto

- Pueden usarse autoincrementos o autodecrementos y combinarse el direccionamiento indirecto a nuevos

Direccionamiento Indirecto a Memoria

- La instrucción contiene una dirección de memoria donde está contenida la dirección donde se almacena el objeto.

$ID .R1, [1000]$
 $R1 \leftarrow MEM(MEM(1000))$



• P. Pila -> apunta a la dirección de Mp que actúa como cima de pila

POP .R1 -> saca la info de la pila y la carga en R1
 PUSH .R1 -> pone en la pila la info de R1

puede ser cualquier otro direccionamiento

• cuando SP apunta a la cima de la pila -> cuando SP apunta a la cima de la pila

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

la 1ª direct. libre

Cartagena99

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Juego de instrucciones

- Conjunto de instrucciones del computador.
- "Herramientas" con las que construir programas
- Debe ser completo y eficaz.
- Tipos:
 - Transferencia
 - Aritméticas
 - Lógicas
 - Bifurcaciones
 - Desplazamiento y rotación
 - De bit

Instrucciones

LD destino, Origen cargar
ST origen, destino guardar
Move origen, destino mover

PUSH Origen

POP Origen

Carga Origen en el pñtero de pñla

Saca el 1º deca de la pñla y la carga en origen.

Transferencia de datos

- Mueven datos entre registros, registros y posiciones de memoria y entre posiciones de memoria.
- No modifican los biestables de estado.

• LD, ST y MOVE
• LD R2, #4(.R4) R2 ← MEM(R4+4)
• ST R2, #4(.R4) MEM(R4+4) ← R2
• MOVE R2, R4 R4 ← R2
• MOVE (.R2), (.R4) MEM(R4) ← MEM(R2)
• PUSH y POP
• PUSH R1 SP ← SP - 4; MEM(SP) ← R1
• POP R1 R1 ← MEM(SP); SP ← SP + 4

Bifurcaciones = Saltos

- Modifican la secuencia del programa. Lo habitual es que la siguiente instrucción que se ejecuta sea la siguiente en secuencia. En este caso no es así.
- No modifican los biestables de estado.

Incondicionales

• BR. Le sigue un direccionamiento a memoria cuyo objeto es la siguiente instrucción a ejecutar
• BR /1000 PC ← 1000 *El PC no avanza, apunta directamente a 1000*
• BR #4(.R4) PC ← R4+4
• BR \$10 PC ← PC + 10 *El PC apunta a la dirección de la siguiente instrucción !!*

• Cambia la ejecución del programa
• no cambia el registro de estado

→ Se genera siempre, sin condiciones

Bifurcaciones Condicionales

- Bcc dir. Dir es un direccionamiento a memoria.

Si cc = 1 entonces se ejecuta la bifurcación
Si no se continúa en secuencia

• Cc: Z, NZ, C, NC, V, NV, P, N o alguna combinación de estos biestables.

• BZ /1000 SI Z = 1 PC ← 1000

• BNC #4(.R4) SI C = 0 PC ← R4+4

• BP \$10 SI S = 0 PC ← PC + 10

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, starburst-like background that tapers to the right. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Bifurcaciones Con retorno

• En la pila:

•CALL /1000 $SP \leftarrow SP - 4$; $MEM(SP) \leftarrow PC$;
 $PC \leftarrow 1000$

•RET $PC \leftarrow MEM(SP)$; $SP \leftarrow SP + 4$

•Permite llamadas anidadas

•Si en la subrutina se trabaja con la pila, hay que tener en cuenta el modo de crecimiento de la pila y que la dirección de retorno ha quedado en la cima de la pila.

Aritméticas y Comparación

•ADD .R1, .R2 $R1 \leftarrow R1 + R2$; mod. flags

•SUB .R1, .R2 $R1 \leftarrow R1 - R2$; mod. flags

•MUL .R1, .R2 $R1 \leftarrow R1 \cdot R2$; mod. flags

•DIV .R1, .R2 $R1 \leftarrow R1 / R2$; mod. flags

•ADDC .R1, .R2 $R1 \leftarrow R1 + R2 + c$; mod. flags

•SUBC .R1, .R2 $R1 \leftarrow R1 - R2 - c$; mod. flags

•CMP .R1, .R2 $R1 - R2$; mod. flags

Aritméticas y Comparación

• Fijan el modelo de ejecución del computador.

•Modelo Registro-Registro.

•ADD .R1, .R2 $R1 \leftarrow R1 + R2$; mod. Flags

•ADD .R1, #4 $R1 \leftarrow R1 + 4$; mod. Flags

•Modelo Registro-Memoria.

•ADD .R1, [.R2] $R1 \leftarrow R1 + MEM(R2)$; mod. Flags

•Modelo Memoria-Memoria.

•ADD [.R1], #4[.R2] $MEM(R1) \leftarrow MEM(R1) + MEM(R2+4)$;

mod. Flags. *En los operandos sólo puede haber 2 direcciones de memoria (cualquier tipo de almacenamiento)*

•Definen el número de direcciones del computador.

Lógicas

•Realizan la operación lógica indicada por el mnemónico bit a bit.

•Establecen el modelo de ejecución del computador

•AND .R1, .R2 $R1 \leftarrow R1 \text{ AND } R2$; mod. Flags

•XOR .R1, #4 $R1 \leftarrow R1 \text{ XOR } 4$; mod. Flags

•OR .R1, .R2 $R1 \leftarrow R1 \text{ OR } R2$; mod. Flags

•NOT .R1 $R1 \leftarrow \text{NOT } R1$; mod. Flags

•Se utilizan para trabajar con máscaras *función lógica*

•AND .R1, #1; Si Z = 1 el número es par

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, blue, serif font. The text is set against a light blue, abstract background that resembles a stylized 'C' or a wave. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

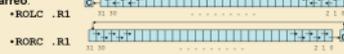
Rotación

• Realizan rotaciones de bits a izquierda/derecha.

• Rotación.



• Rotación con acarreo



Otras instrucciones:

- NOP, no operación → durante uno o varios ciclos la máquina no hace nada.
- HALT, para la ejecución.
- WAIT, la máquina entra en copesa durante una serie de ciclos

I/O

- IN, A, /n A ← (n) carga un puerto en el registro
- OUT, A, /n (n) ← A carga un registro en el puerto

Sufijo o Extensiones

- Referidos a condiciones del registro de estado
 - I / bit
 - Z / bit
 - S / bit
 - U / unsigned (operacion sin signo)
- Referidos a la representación de datos
 - b / byte
 - w / word (doble palabra)
 - s / short (media palabra)
 - d / decimal
 - float (coma flotante)

De bit

• Ejecutan operaciones con un bit: pone a 1 o a 0 un bit.

• CLR.I #3, .R1 ; Pone a 0 el bit 3 de R1

• SET.I #3, .R1 ; Pone a 1 el bit 3 de R1

• TEST.I #3, .R1 ; Z ← NOT(R1(3))



COMPUTADORES CISC Y RISC

CISC - Complex instruction set computer (juegos de instrucciones complejos)
- Su modelo de ejecución es de registro a registro

RISC - Reduced instruction set computer (juegos de instrucciones reducidos)
- Tienen alrededor de solo 30 instrucciones, lo que hace que los programas sean más largos para que hagan lo mismo, así así como son instrucciones más sencillas, la ejecución es mucho más rápida

ERRORES.

- Palabra y dirección de 32 bits.
- 30 registros generales.
- 500 instrucciones.
- 8R #R1, #R2
- El Código de Operación ocupa 9 bits (combinación necesaria para obtener a...
- El R1 ocupa 5 bits y el desplazamiento ocupa el resto (8 bits)
- ADDI #R1, #R2
- El código ocupa 9 bits
- Inmediato (operando) ocupa 18 bits
- ST, #R1, #R2
- El código ocupa 9 bits
- El R1 ocupa 5 bits
- Los operandos ocupan 18 bits
- la dirección necesita muchos bits es otra palabra a parte de 32 bits
- LD, #R1, #R2
- El C.O. son 9 bits
- R1 y R2 son 5 bits cada uno
- El desplazamiento son 13 bits

ERRORES 2.

Para un computador de dos direcciones con modelo de ejecución registro a registro dispone únicamente de los siguientes modos de direccionamiento: inmediato, directo, instrucciones correspondientes para las instrucciones que se muestran a continuación: RT1, y RT2 como registros auxiliares.

1. Accede a 1000 y copia ese dato
2. Leer la segunda palabra de la instrucción (1000)
3. Accede a 2000 y copia el dato
4. Guarda la información en 1000

Puede haber varias soluciones válidas:

```
LD RT1, #1000
LD RT2, #2000
LD RT1, RT2
LD RT2, RT2
ADD RT1, RT2
ST RT1, RT1
```

c. CALL #R1, RT1 (Carga el PC con esa dirección)

Como es direccionamiento a R. Base no se modifica.

```
MOVE RT1, RT1
ADD RT1, #4
CALL RT1
```

d. SUB RT1, #R1, RT1 (Hace un Salto)

Hay 2 direcciones de memoria implicadas: R1 ← R1 - (MMAR * 4)

R1 ← R1 + 4 (Post increment)

```
MOVE RT1, RT1
LD RT1, RT1 // Acceso a memoria para obtener la dirección
LD RT1, RT1 // acceso para obtener el operando
SUB RT1, RT1
ADD RT1, #4
```

PROBLE 1

En un computador con palabras y direcciones de 16 bits, direccionamiento a nivel de palabra de cada instrucción se indica como comentario. Considere que a partir de...

Indique los valores sucesivos...

```
LD RT1, #1000 ; palabras
LD RT2, #1008 ; palabras
CMP RT1, RT2
BNEZ S, 6
MOVIE RT1, RT1+1
HALT
```

RT1: 1000, RT2: 1008, RT1: 1009, RT1: 1010, RT1: 1011

PROBLEMA 2

En un computador con palabras y direcciones de 16 bits, direccionamiento a nivel de...

Indique los valores sucesivos...

```
LD RT1, #R1
LD RT2, #R2
ADD RT1, RT2
LD RT1, RT1
LD RT2, RT2
CALL RT1
```

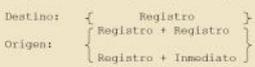
RT1: R1, RT2: R2, RT1: R1+R2, RT1: R1, RT2: R2, RT1: R1

ARQUITECTURA 88110 Procesador

• Máquina de 3 direcciones (0: add 1: add B 2: add A,B)

```
add rD, rS1, rS2
rD ← rS1 + rS2
```

• Modelo de ejecución registro-registro (ALU, no jmp/ld/st)



• Palabra de 32 bits

• ALU opera en complemento a 2

• Emulador: ejecución serie / superescalar

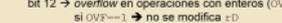
ARQUITECTURA 88110 Banco de Registros

• Banco general de 32 registros: r0...r31

r0 cableado a 0 (siempre tiene valor 0)
r1 guarda dirección de retorno de subrutina accesibles por pares en operaciones de 64 bits

• PC

• PSR (Processor Status Register, registro de estado) / registro de control



bit 12 → overflow en operaciones con enteros (OVF) si OVF=-1 → no se modifica rD

bit 28 → acarreo (carry = borrow)

• Registros que decidimos reservar para función particular:
r30 puntero de pila y r31 puntero de marco de pila

lo por eso los prog suelen ir de r2 a r31.

INSTRUCIÓN DE COMENTARIOS: Ejercicios y Problemas (Instrucciones y Direccionamiento) Página 3 de 3

8 Para un computador de dos direcciones con modelo de ejecución registro a registro con palabras de 32 bits y direccionamiento a nivel de byte que dispone únicamente de los siguientes modos de direccionamiento: inmediato, directo o registro a registro, indique los registros correspondientes a las instrucciones que se muestran a continuación para el computador indicado. Si es necesario utilizar los registros RT1, RT2 y RT3 como registros auxiliares.

```
ADD #1000, #2000
CALL #R1, RT1
```

9 Dado un computador con palabras y direcciones de 32 bits, 38 registros de propósito general y 380 instrucciones. Indique el formato de formato de instrucciones que contenga las siguientes instrucciones:

```
MOV RT1, RT1
ADDI #R1, #4
ST RT1, #2000
LD RT1, #R1, RT1
```

10 Para un computador con palabras y direcciones de 32 bits, 38 registros de propósito general y 380 instrucciones. Dado el código objeto de formato de instrucciones que se muestra a continuación:

```
BNEZ RT1, #R1
ADD RT1, RT1, #4
LD RT1, #R1, RT1
ST RT1, #2000
```

11 Considere un computador de 32 bits y direccionamiento a nivel de byte, en el que la pila crece hacia direcciones decrecientes y el puntero de pila apunta a la primera palabra libre. Dado los subrutinas del computador con una palabra y la dirección de retorno de la subrutina se muestran en la pila. Dado el programa que se muestra a continuación, complete la siguiente tabla. Indique para cada instrucción principal sus abreviatura y parte de la dirección (y la subrutina a partir de la cual) que el SP tiene el valor 5000, y las direcciones de las direcciones RT1 a RT3 con 1, 2, 3, 4 y 5.

Programa principal	Subrutina	Instrucción 1	Instrucción 2
mov RT1, #R1	ld r1, #R1, RT1	r1 ← 0	r1 ← 0
ld r2, #1000	ld r2, #R2, RT1	r2 ← 1000	r2 ← 1000
ld r3, #R2	ld r3, #R2, RT1	r3 ← RT1 + 4	r3 ← RT1 + 4
push r2	add r1, r1, r2	r1 ← r1 + r2	r1 ← r1 + r2
push r3	mov r2, #4	r2 ← 4	r2 ← 4
push r1	mov r3, #R2	r3 ← R2	r3 ← R2
ld r4, #R3	ld r4, #R3, RT1	r4 ← R3	r4 ← R3
add r1, r1	ret		
pop r1	halt		

1000	1004	1008	1012	1016
1	2	3	4	5



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

menos 500 número(instrucciones)

rio, palabras y direcciones de 32 bits y direccionamiento a byte que
recto a registro o indirecto a registro Realice los programas (conq. De
cción para el computador indicado. Si es necesario utilizar los registros R1,

ndría 3 palabras

an el contenido de los registros

por direccionamiento a byte?)

l da palabra, se ejecuta el siguiente fragmento de código en el que el
la dirección de memoria #1008 se encuentran almacenados los siguientes
100C)

*• Los 4 son el (contenido)
y se le confunde*



se en el que utiliza el registro R3 como puntero de palabra (R1+R3 apuntando
a las siguientes secuencias de instrucciones suponiendo que el contenido de

- 1) R3+R3, R1+R2
- 2) R1+R1, R3+R3 (para que siga apuntando a la siguiente posición libre)
- 3) R1+R2, R3+R3
- 4) R1+R2, R3+R3
- 5) R1+R2, R3+R3
- 6) R1+R2, R3+R3
- 7) R1+R2, R3+R3
- 8) R1+R2, R3+R3
- 9) R1+R2, R3+R3
- 10) R1+R2, R3+R3
- 11) R1+R2, R3+R3
- 12) R1+R2, R3+R3
- 13) R1+R2, R3+R3
- 14) R1+R2, R3+R3
- 15) R1+R2, R3+R3
- 16) R1+R2, R3+R3
- 17) R1+R2, R3+R3
- 18) R1+R2, R3+R3
- 19) R1+R2, R3+R3
- 20) R1+R2, R3+R3
- 21) R1+R2, R3+R3
- 22) R1+R2, R3+R3
- 23) R1+R2, R3+R3
- 24) R1+R2, R3+R3
- 25) R1+R2, R3+R3
- 26) R1+R2, R3+R3
- 27) R1+R2, R3+R3
- 28) R1+R2, R3+R3
- 29) R1+R2, R3+R3
- 30) R1+R2, R3+R3
- 31) R1+R2, R3+R3
- 32) R1+R2, R3+R3
- 33) R1+R2, R3+R3
- 34) R1+R2, R3+R3
- 35) R1+R2, R3+R3
- 36) R1+R2, R3+R3
- 37) R1+R2, R3+R3
- 38) R1+R2, R3+R3
- 39) R1+R2, R3+R3
- 40) R1+R2, R3+R3
- 41) R1+R2, R3+R3
- 42) R1+R2, R3+R3
- 43) R1+R2, R3+R3
- 44) R1+R2, R3+R3
- 45) R1+R2, R3+R3
- 46) R1+R2, R3+R3
- 47) R1+R2, R3+R3
- 48) R1+R2, R3+R3
- 49) R1+R2, R3+R3
- 50) R1+R2, R3+R3

PC	SP	Z	R1	R2	R3	5000	4996	4992	4988
0	5000	1	0	1000	5	1000	5	0	32
4	4996								
8	4992								
12	4988								
16	4984								
20									
24									
28									
32									
36									
40									
44									
48									
52									
56									
60									
64									
68									
72									
76									
80									
84									
88									
92									
96									
100									

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



ARQUITECTURA 88110

Memoria principal

- Almacena: Instrucciones + Datos
- Direccional a nivel de byte
1 palabra → 4 direcciones de memoria
- Bus de direcciones de 32 bits
Máxima capacidad (teórica) → 2^{32} bytes = 4 GB
 $\left\{ \begin{array}{l} 0x00000000 \\ 0xFFFFFFFF \end{array} \right.$
- Capacidad del emulador:
 2^{18} direcciones = 2^{18} bytes = 256 KB
 $\left\{ \begin{array}{l} 0x00000000 \\ 0x0003FFFF \end{array} \right.$

ARQUITECTURA 88110

Modos de direccionamiento

- Sí tiene:
 - Directo a registro: `.Ri`
 - Inmediato: `#aaaa`
 - Relativo a registro base: `#disp[.Ri]`
 - Relativo a PC: `$xx`
 - Indirecto a registro: `[.Ri]`
- NO tiene:
 - ~~Absoluto: `/dir`~~
 - ~~Relativo a registro índice: `rr, --`~~
 - ~~Indirecto a memoria: `[/dir]`~~
 - ~~Relativo a pila: `push / pop`~~

ARQUITECTURA 88110

Direccionamiento directo a registro

```
add r1, r2, r3 ; r1 ← r2 + r3
```

equivalente en máquina de 2 direcciones, IEEE 694:

```
ADD .R7, .R9 ; ( add r7, r7, r9 )
```

ARQUITECTURA 88110

Direccionamiento inmediato

- Puede ser con/sin signo, ambos de 16 bits: *p/10, cap 1, apartado 1.3 del manual*
 Con signo: SIMM16
 Sin signo: IMM16

- Se puede expresar en decimal o hexadecimal

Ejemplo:

```
add r1, r2, -13;
```

13 en binario: 0000 1101
 -13 en binario: - 0000 1101
 -13 en binario: 1111 0011

```
add r1, r2, 0xFFFF { 1111 1111 r2  
                    + FFFF FFF3 → r1  
                    (con signo)
```

```
addu r1, r2, 0xFFFF { 1111 1111 r2  
                    + 0000 FFF3 → r1  
                    (sin signo)
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

Cartagena99

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, abstract background that resembles a stylized 'C' or a wave. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

ARQUITECTURA 88110

Direccionamiento relativo a PC

- Ejemplo:
`br 7 ; PC ← PC + 7*4`
(desplazamiento: 26 bits / 16 bits)
- Ejemplo en formato del estándar IEEE:
`ADD .7, .5`
`BR $desp ; PC ← et1+desp`
`et1: LD .1, [-7]`
- Ejemplo en formato de 88110:
`add z7, z7, x5`
`et0: br D ; PC ← et0 + 4*D`
`ld r1, z7, 0`

← SIA ☆☆☆
CHANDELIER

ARQUITECTURA 88110

Direccionamiento indirecto a registro

En el 88110 solo existe en los saltos:

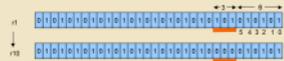
- Ejemplo en formato del estándar IEEE:
`JMP [R10] ; PC ← R10`
- Ejemplo en formato del 88110:
`jmp (r10) ; PC ← r10`

ARQUITECTURA 88110

Direccionamiento campos de bit

En ciertas instrucciones del 88110 se pueden seleccionar:
- bits individuales
- campos de bit

- Ejemplo en formato del estándar IEEE / 88110:
`CLR.L3 .R10, R1, 6 / clr r10, r1, 3<6>`



- Otro ejemplo (bits individuales):

`bb0 3, z8, 7 ; S1 (bit3 de z8) == 0 → PC ← PC + 4*7`

ARQUITECTURA 88110

Juego de instrucciones

Tipos de instrucciones en el 88110:

- Lógicas (`or`, `and`, `xor`, `mask`)
- Aritméticas (`add`, `sub`, `addu`, `subu`, `mula`, `mulu`, `divu`, `divu`, `cmp`)
- Bifurcaciones (`bb0`, `bbz`, `br`, `baz`, `jmp`, `jaz`)
- Transferencia (`ld`, `st`, `ldex`, `stex`, `xxmem`)
- Campos de bit (`clr`, `set`, `ext`, `extu`, `mak`, `rot`)
- Coma flotante (`fsadd`, `fsub`, `fsml`, `fdiv`, `fcvt`, `flt`, `int`, `fcmp`)

Instrucción específica del emulador: `stop`

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, dark blue font. The text is set against a light blue, irregular shape that resembles a map of the city of Cartagena. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

ARQUITECTURA 88110

Instrucciones aritméticas (I)

Aritméticas: add, sub, addu, subu, muls, mulu, divs, divu, cmp

INST	Operandos	Ext
add	rD, rS1, rS2	ci,co,cio
sub	rD, rS1, SIMM16	
addu	rD, rS1, rS2	ci,co,cio
subu	rD, rS1, IMM16	

ci: opera con acarreo de entrada (bit28 de PSR)
co: actualiza el flag de acarreo (bit28 de PSR)
cio: equivale a usar cico

ARQUITECTURA 88110

Instrucciones aritméticas (II)

Aritméticas: add, sub, addu, subu, muls, mulu, divs, divu, cmp

```

muls rD, rS1, rS2 ; excep. OVF
mulu { rD, rS1, rS2
      rD, rS1, IMM16
mulu.d rD, rS1, rS2 ; d.p. en rD
divs { rD, rS1, rS2 ; excep. div por 0
      rD, rS1, SIMM16
divu { rD, rS1, rS2 ; excep. div por 0
      rD, rS1, IMM16
divu.d rD, rS1, rS2 ; d.p. en rS1 y rD
    
```

ARQUITECTURA 88110

Instrucciones aritméticas (III)

Aritméticas: add, sub, addu, subu, muls, mulu, divs, divu, cmp

```

cmp { rD, rS1, rS2
     rD, rS1, SIMM16
rD [nh|he|nb|be|hs|ls|hi|ge|lt|le|gt|ne|eq|0|0
    15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
(resto de bits a '0')
eq: 1 si y solo si rS1 = rS2
ne: 1 si y solo si rS1 ≠ rS2
gt: 1 si y solo si rS1 > rS2 (con signo)
...
hi: 1 si y solo si rS1 > rS2 (sin signo)
...
    
```

ARQUITECTURA 88110

Bifurcaciones/saltos

Bifurcaciones (bb0, bbl, br, bsr, jmp, jsr)

INST	Operandos	Ext
bb0	B, rS1, D16	PC ← PC+4*D16 → 0
bbl	B, rS1, D16	si bit B de rS1 = 1
br	D26	PC ← PC+4*D26
bsr	D26	r1 ← PC+4; PC ← PC+4*D26
jmp	(rS1)	PC ← rS1 (alineado)
jsr	(rS1)	r1 ← PC+4; PC ← rS1 (alineado)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, starburst-like background that tapers to the right. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

ARQUITECTURA 88110

Campos de bit

Campos de bit (clr, set, ext, extu, mak, rot)

INST	Operandos
clr	
set	rD, rS1, #5<O5>
ext	
extu	rD, rS1, rS2
mak	
rot	rD, rS1, <O5> rD, rS1, rS2

ARQUITECTURA 88110

Instrucciones de coma flotante

Coma flotante (fadd, fsub, fmul, fdiv, fcvt, flt, int, fcmp)

INST	Operandos	explicación
fadd.xxx		
fsub.xxx		$x=s \bar{x}=d$
fmul.xxx	rD, rS1, rS2	$x=d \bar{x}=s$
fdiv.xxx		
fcvt.x \bar{x}		
flt.xs	rD, rS2	
int.sx		
fcmp.sxx	rD, rS1, rS2	

Pueden generar excepciones: Overflow/Underflow/NaN/Div0

ARQUITECTURA 88110

Ensamblador/Cargador

•Ensamblador: Programa que se encarga de "traducir" un programa escrito en lenguaje ensamblador a lenguaje máquina.

etiqueta: instrucción_ensamblador ; Comentarios

•Instrucción_ensamblador: Puede ser una instrucción-máquina o una pseudoinstrucción.

•Pseudoinstrucción:

- Instrucción para el programa ensamblador.
- No se traduce en una instrucción en memoria.
- Indica al ensamblador cómo debe generar el código-máquina.

ARQUITECTURA 88110

Pseudoinstrucciones

•Org n: Indica que el código que le sigue se almacene en la posición de memoria n.

•Res n: Indica que se reserven n bytes en memoria. N debe estar alineado a palabra.

•Data a, b, c,: Inicializa las posiciones de memoria con los valores a, b y c.

•Data "texto": Inicializa las posiciones de memoria con la cadena de bytes texto. Asegura que la siguiente palabra en memoria está alineada (véase ejemplo).

•Low(etiqueta o inmediato): Devuelve los 16 bits menos significativos de la dirección asociada a la etiqueta o dato inmediato.

•High(etiqueta o inmediato): Devuelve los 16 bits más significativos de la dirección asociada a la etiqueta o dato inmediato.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, starburst-like background that tapers to the right. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

ARQUITECTURA 88110

Ejemplo "data" (2)

```

practicas@avellanot mc88110 data.bin
PC=0  id 00000000  Tot. Inst: 0  Ciclo : 1
FL=1 FE=1 FC=0 FV=0 FE=0 (s.l.s.l.)
R01 = 00000000 h R02 = 00000000 h R03 = 00000000 h R04 = 00000000 h
R05 = 00000000 h R06 = 00000000 h R07 = 00000000 h R08 = 00000000 h
R09 = 00000000 h R10 = 00000000 h R11 = 00000000 h R12 = 00000000 h
R13 = 00000000 h R14 = 00000000 h R15 = 00000000 h R16 = 00000000 h
R17 = 00000000 h R18 = 00000000 h R19 = 00000000 h R20 = 00000000 h
R21 = 00000000 h R22 = 00000000 h R23 = 00000000 h R24 = 00000000 h
R25 = 00000000 h R26 = 00000000 h R27 = 00000000 h R28 = 00000000 h
R29 = 00000000 h R30 = 00000000 h R31 = 00000000 h
88110>

```

ARQUITECTURA 88110

Ejemplo "data" (3)

```

88110> e
Fin ejecución
PC=16  instrucción incorrecta  Tot. Inst: 4  Ciclo : 62
FL=1 FE=1 FC=0 FV=0 FE=0 (s.l.s.l.)
R01 = 00000000 h R02 = 00000000 h R03 = 01020304 h R04 = 00000000 h
R05 = 00000000 h R06 = 00000000 h R07 = 00000000 h R08 = 00000000 h
R09 = 00000000 h R10 = 00000000 h R11 = 00000000 h R12 = 00000000 h
R13 = 00000000 h R14 = 00000000 h R15 = 00000000 h R16 = 00000000 h
R17 = 00000000 h R18 = 00000000 h R19 = 00000000 h R20 = 00000000 h
R21 = 00000000 h R22 = 00000000 h R23 = 00000000 h R24 = 00000000 h
R25 = 00000000 h R26 = 00000000 h R27 = 00000000 h R28 = 00000000 h
R29 = 00000000 h R30 = 00000000 h R31 = 00000000 h
88110> v 400
      400  04030201  00000000  00000000  00000000
      416  55544444  31222222  35363738  39306162
      432  23646566  67890ABD  09000000  0F000000
      448  AFD70000  FFFFFFFF  00000000  00000000
      464  00000000  00000000  00000000  00000000
88110>

```

Es lo que hace el programa de la pág. 53

ARQUITECTURA 88110

Macroinstrucciones ("macros")

•Conjunto de sentencias a las que se le asigna un nombre y se le pasa un conjunto de argumentos.

•Cuando aparece la invocación de la macro se sustituye en fase de ensamblado la macro por el conjunto de sentencias declarado en la macro cambiando los parámetros declarados por los que se pasan en la invocación.

Nombre de macro: MACRO (arg1, arg2, ..., argn)

Conjunto de instrucciones
Que componen la macro

ENDMACRO

```

swap: MACRO (ra, rb)
    or r1,ra,ra
    or ra,rb,rb
    or rb,r1,r1

```

ENMAMACRO
Opis. Arquitectura y Tecnología de Sistemas Informáticos.
Universidad Politécnica de Madrid

ARQUITECTURA 88110

Macroinstrucciones ("macros") II

•Una macro debe haberse definido previamente.

•Se permiten macros anidadas.

•No se permite la definición de etiquetas dentro de una macro.

•Se utilizan para encapsular pequeños fragmentos de código para los que no merece la pena construir una subrutina.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, blue, serif font. The text is set against a light blue, abstract background that resembles a stylized 'C' or a wave. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, dark blue font. The text is set against a light blue, irregular shape that resembles a map of the city of Cartagena. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Subrutinas

•Tipos de variables que utiliza una subrutina:

•Variables **globales**. Se crean cuando arranca el programa y tienen validez durante toda la vida del mismo. Cualquier subrutina puede acceder a estas variables.

•Variables **locales** a una subrutina. Se crean cada vez que se activa la subrutina y se destruyen cuando se finaliza cada ejecución. Solo la subrutina que las crea puede acceder a ellas.

•Parámetro: Dato de entrada/salida de una subrutina que es necesario para su operación:

•Por **valor**. Se pasa el dato necesario para su operación.

•Por **dirección**. Se pasa la dirección de memoria en la que está contenido el dato/resultado necesario para la operación.

Subrutinas: Paso de parámetros En registros

•El llamante y la subrutina "acuerdan" un conjunto de registros de propósito general para intercambiar los datos y resultados.

•Rápido

•Limitado en el tipo y el número de parámetros.

•Llamante

```
ld r2, r20, 0
ld r3, r20, 4
brr suma
```

Subrutina

```
suma: add r2, r2, r3
      jmp (r1)
```

Subrutinas: Paso de parámetros En variables globales

•Llamante

```
num1: res 4
num2: res 4
brr suma
```

Subrutina

```
suma: or r20, r0, low(num2)
      or u r20, r20, high(num2)
      ld r5, r20, 0
      or r20, r0, low(num1)
      or u r20, r20, high(num1)
      ld r6, r20, 0
      add r5, r5, r6
      st r5, r20, 0
      jmp (r1)
```

Subrutinas: Paso de parámetros En variables globales

•El llamante y la subrutina "acuerdan" un conjunto de variables globales, puesto que son visibles desde todas las subrutinas, para intercambiar los datos y resultados.

•Sencillo.

•Limitado en subrutinas de librería y genera problemas de reentrancia

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, blue, serif font. The '99' is significantly larger and more prominent than the 'Cartagena' part. The text is set against a light blue background with a subtle gradient and a white shadow effect.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, abstract background that resembles a stylized 'C' or a speech bubble. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

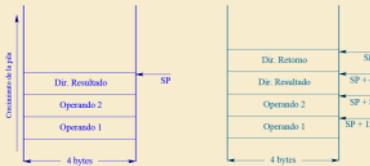
**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Subrutinas: Paso de parámetros En la pila

Llamante	Subrutina
RESULT: RES 4 parámetro	SUMA: LD .R5, #8[.SP] ;
PUSH .R1	LD .R6, #12[.SP]; parámetro
PUSH .R2	ADD .R5, .R6
LD .R1, #RESULT	LD .R6, #4[.SP]
PUSH .R1	ST .R5, [.R6] ; resultado
CALL ./SUMA	RET

Subrutinas: Paso de parámetros En la pila



88110: Gestión de la pila

- 88110 tiene limitaciones:
 - No tiene puntero de pila, por tanto no tiene PUSH ni POP: el número y tipos de parámetros es limitado.
 - Almacena la dirección de retorno en r1: el número de llamadas anidadas es limitado.
- Soluciones:
 - Se asigna un registro de propósito general como puntero de pila: r30.
 - Se crean dos macros: PUSH y POP
- El protocolo que se muestra en la siguiente transparencia es obligatorio si hay llamadas anidadas.

88110: Gestión de la pila

```

PUSH: MACRO (ra)
    subu r30, r30, 4
    st ra, r30, 0
ENDMACRO

result: res 4
...
PUSH (r1)
PUSH (r2)
or r1, r0, low(result)
or u r1, r1, high(result)
PUSH (r1)
bsr suma

POP: MACRO (ra)
    ld ra, r30, 0
    addu r30, r30, 4
ENDMACRO

suma: PUSH (r1)
    ld r5, r30, 8
    ld r6, r30, 12
    add r5, r5, r6
    ld r6, r30, 4
    st r5, r6, 0
    POP (r1)
    jmp (r1)
    
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, starburst-like background that tapers to the right. Below the text is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Marco de pila

- Se dedica un registro que apunta a una posición conocida del marco de pila: **puntero de marco de pila** (*frame pointer* o FP). En 88110 es r31.
- Si hay llamadas anidadas, cada una de las llamadas tendrá su propio FP.
- Al entrar en la subrutina hay que asegurar que el marco de pila de la subrutina llamante no se destruye.
- Se debe crear el espacio necesario para variables locales.
- Se deben realizar las inicializaciones de las variables locales.

FP = Marco de pila: puntero que apunta a la dirección de retorno de la subrutina; o sea, cuando se acaba de ejecutar la subrutina lo que hace es volver a la ejecución del programa principal justo donde lo había dejado.

Activación de la subrutina y creación del marco de pila

- Almacenar SI: de retroceso (esq en r3)
- Guardar FP antiguo ("r31")
- Crear marco de pila
 - o Establecer el nuevo valor de FP (de r31)
 - o Recuperar espacio en pila para las variables locales
 - o Inicializar variables locales
- Código de la subrutina

Los parámetros en la pila se borran igual que en los computadores.

Destrucción del marco de pila y retorno de la subrutina.

- Restaurar puntero de pila (r30)
- Recuperar FP antiguo (r31)
- ""Indicador de retorno
- Retorno subrutina

Marco de pila

```
void vector (int a[], int b [])
{
    int vector_local[2]; /* Vector de enteros
                          de una palabra (4 bytes) */
    short entero_16bits = 0; /* Entero de dos bytes
                              // Variable de un byte
    ... ..
```

Marco de pila: creación

```
vector: PUSH (r31)      ; Guarda la dir. de retorno
        PUSH (r31)     ; Guarda el puntero al marco
                          de pila del llamante
        or r31, r30, r0 ; Crea el nuevo marco de pila
                          ; a partir de SP (r30)
        subu r30, r30, 12 ; Reserva 12 bytes para
                          ; variables locales
        st.h r0, r31, -10 ; Inicializ. de entero_16bits
                          ; Inicio del código de la subrutina
        ld.r6, r31, 12   ; Accesa la dirección de
                          ; comienzo del vector b
        ... ..
```

Marco de pila



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, abstract background that resembles a stylized 'C' or a wave. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Marco de pila



Parámetros: otras consideraciones

- El orden en el paso de parámetros es un acuerdo entre la subrutina llamante y la llamada.
- En el ejemplo anterior el parámetro a está en la cima de la pila al realizar la llamada a subrutina, tal y como lo realizan los lenguajes de programación de alto nivel.
- El parámetro que está en la cima siempre ocupa la misma posición en la pila independientemente del número de parámetros. Utilizando este se puede acceder al resto.
- Este acuerdo es útil para subrutinas con número variable de parámetros: printf de la librería de C.

Parámetros: funciones

- Funciones: subrutinas que tienen un valor de retorno:
 - Indica el estado de ejecución de la función.
 - Es una función matemática y es su resultado.
- Estos valores de retorno se utilizan inmediatamente:
 - Comprobar el estado.
 - Introducir el resultado en una expresión.
- Por estas razones se utilizan registros y, habitualmente, el valor de retorno no suele ser un tipo complejo de datos (estructura).

Recursividad

- Una subrutina recursiva es la que en su ejecución tiene llamadas a sí misma.
- La recursividad se utiliza para resolver
 - funciones matemáticas cuya definición es recursiva (por ejemplo el factorial).
 - Problemas que requieren almacenamiento en estructuras de datos recursivas (árboles, sistemas de ficheros basados en directorios, etc.)
- Hay que tener en cuenta:
 - Caso general: incluye la llamada a la propia función con diferentes parámetros a los que se recibieron.
 - Caso particular: no se realiza la llamada recursiva.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, starburst-like background that tapers to the right. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**