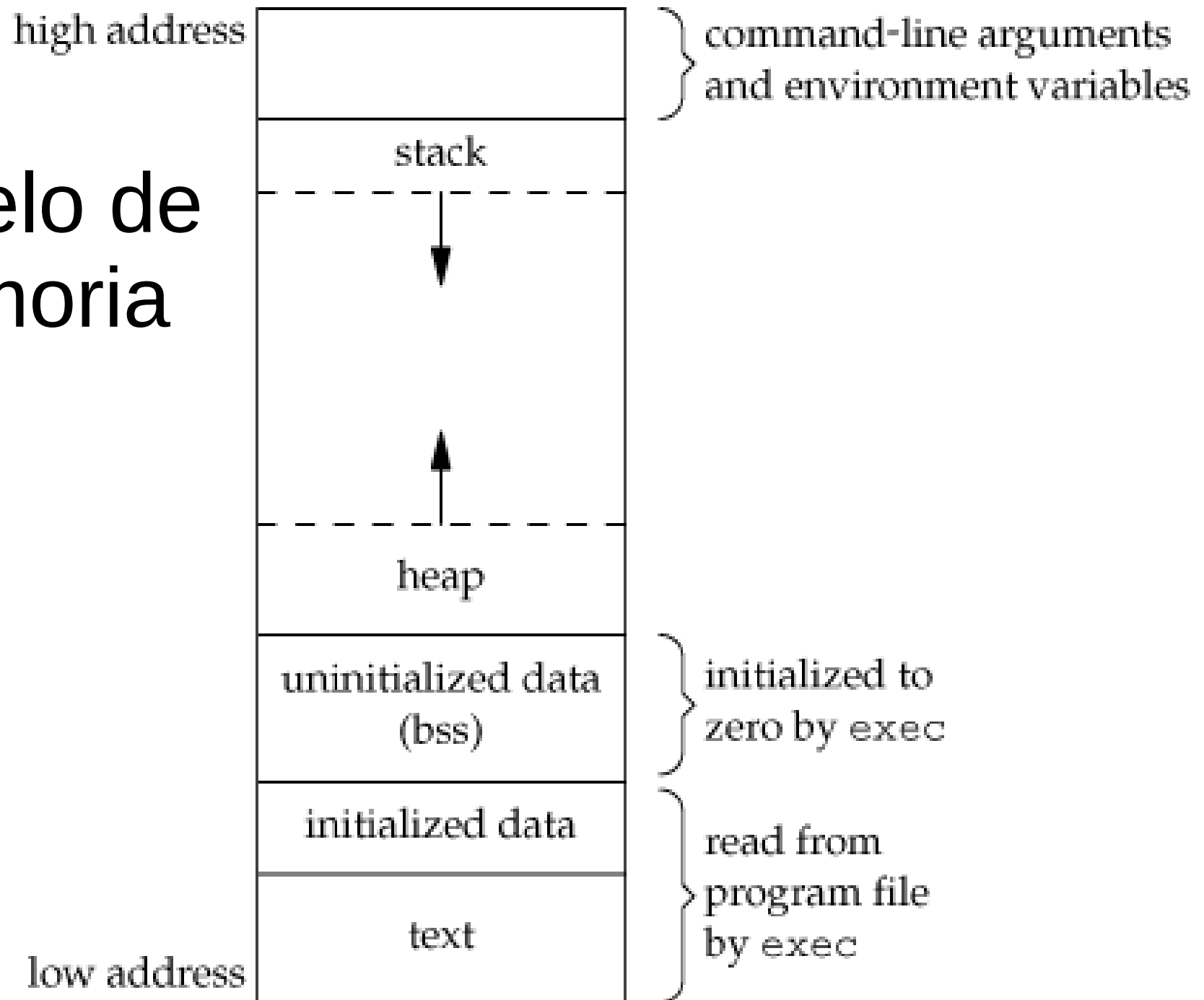
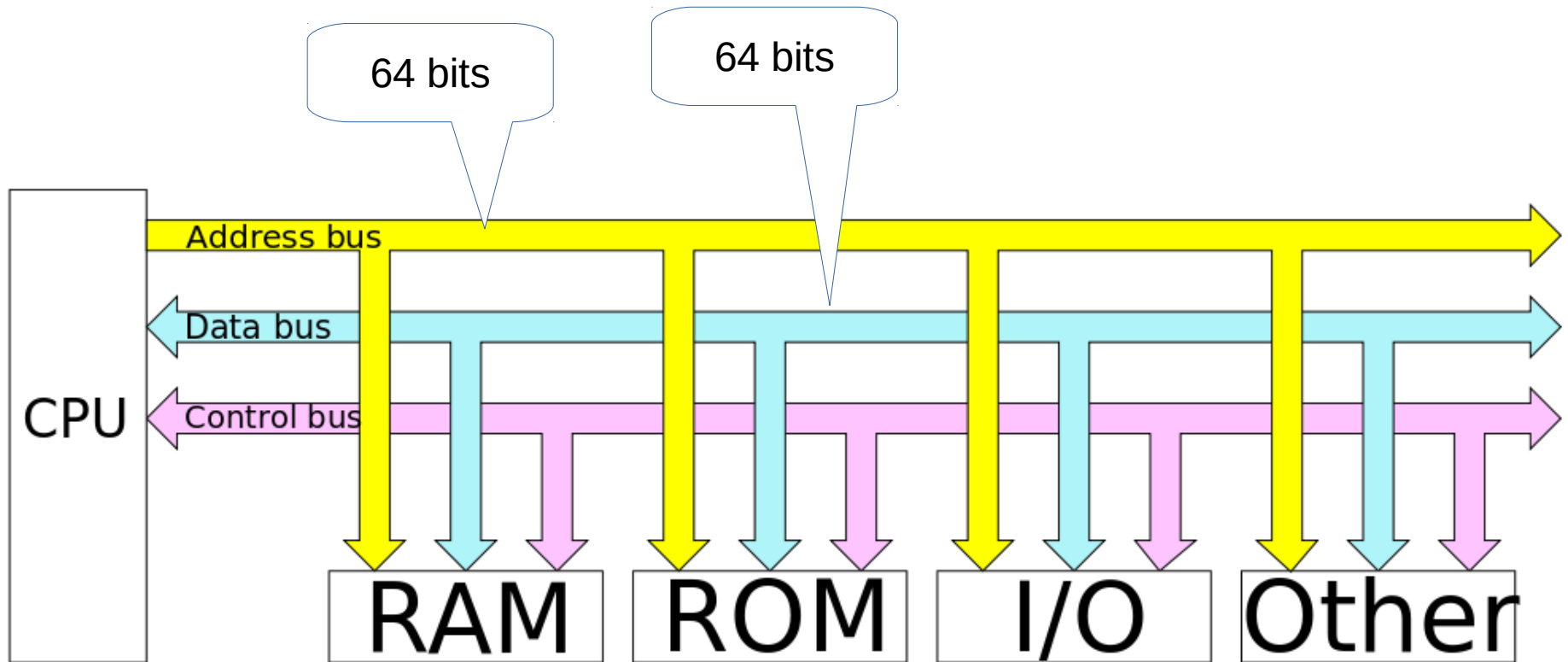


Modelo de memoria



Arquitectura X86_64



Registros: 64 bits

Re-extendidos del viejo 8080

Nuevos

Generales

rax: acumulador
rbx: base
rcx: contador
rdx: datos

Segmento ¡16 bits!

cs: code segment
ss: stack segment
ds: data segment
es: extra segment
fs: f segment
gs: g segment

Punteros

A cadenas:

rsi: source index
rdi: destination index

A la pila:

rbp: base pointer
rsp: stack pointer

Contador de programa

rip

Banderas y control

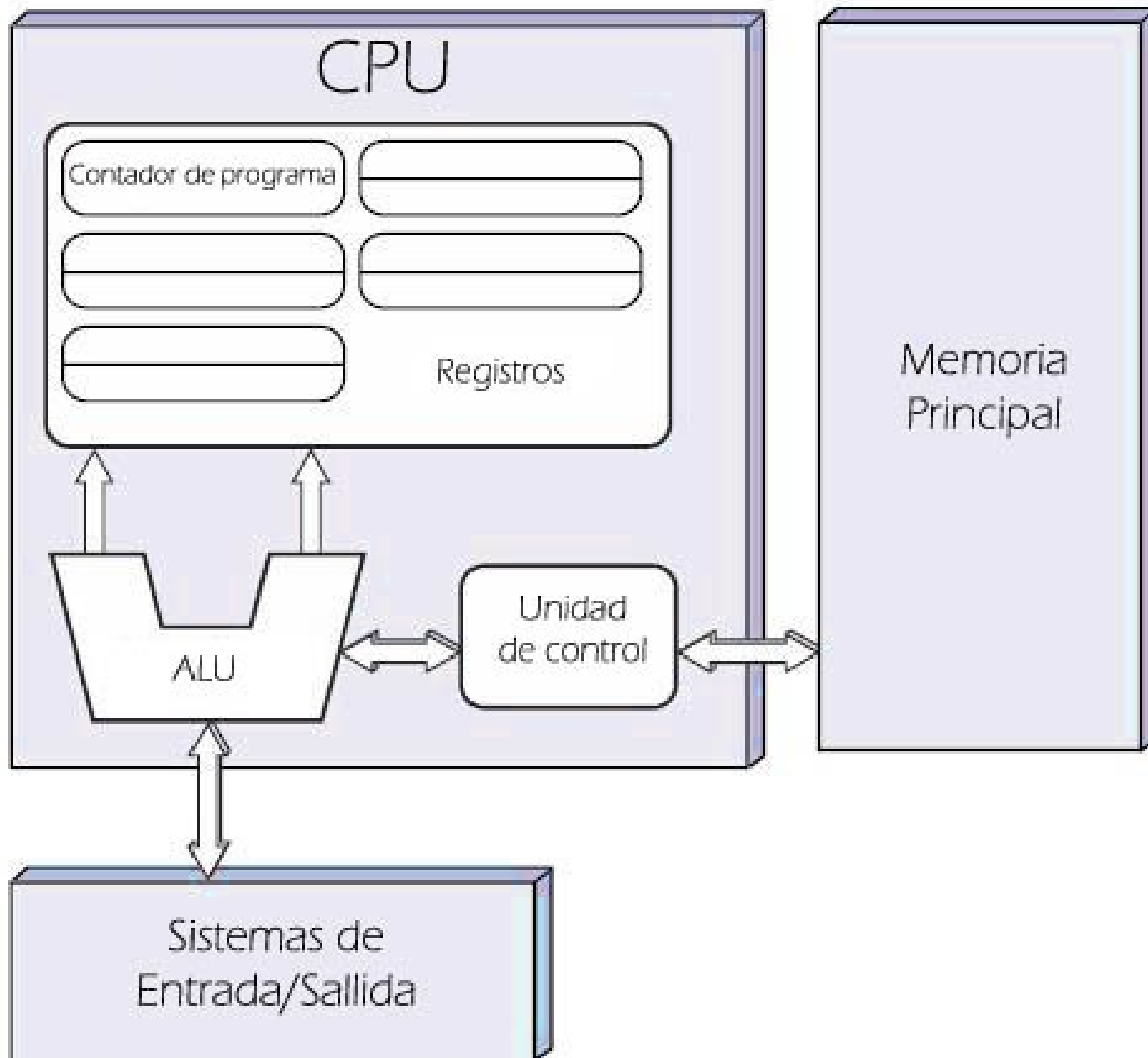
rflags

Punto flotante, MMX, ...

Generales

r8
r9
r10
r11
r12
r13
r14
r15

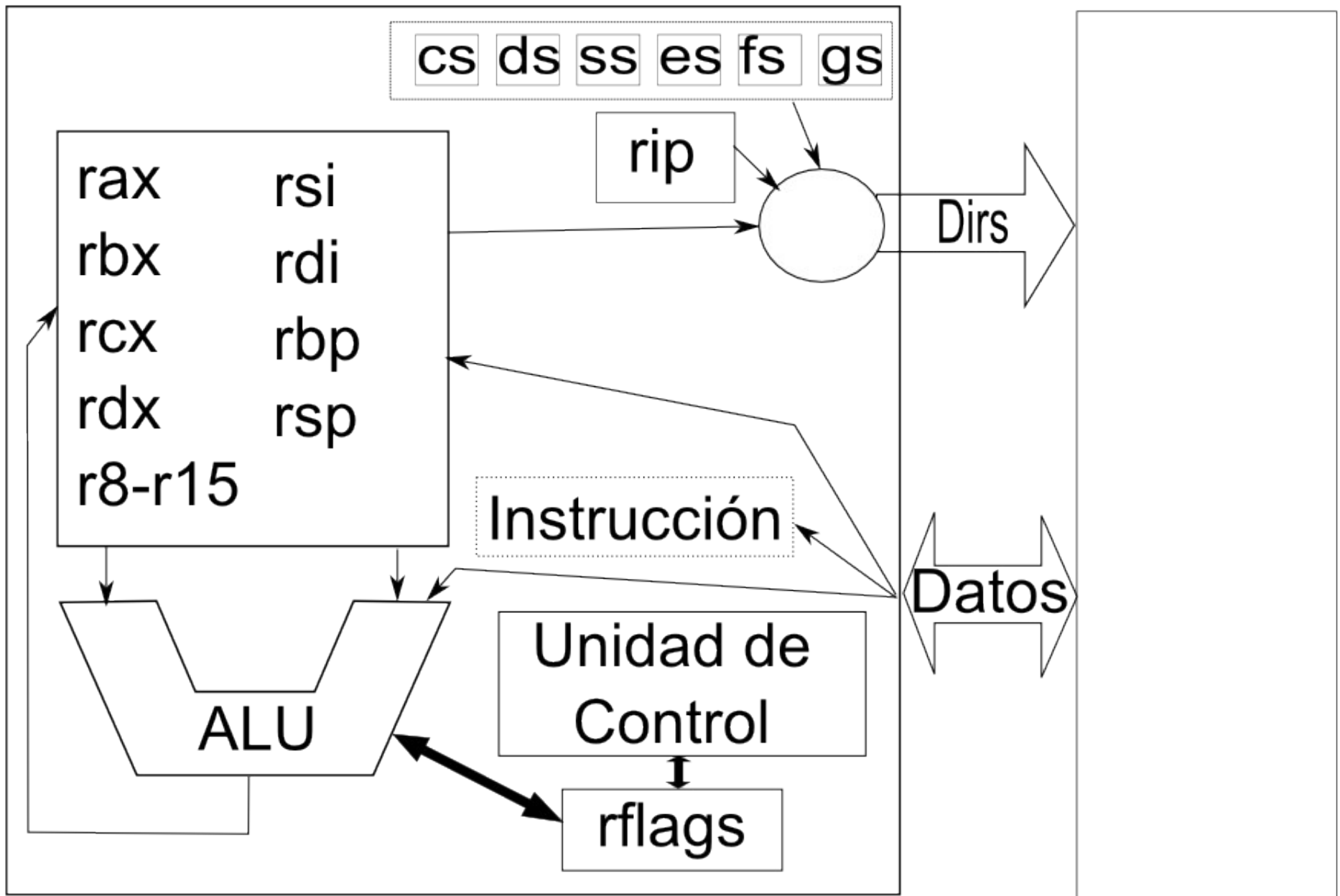
Punto flotante, ...



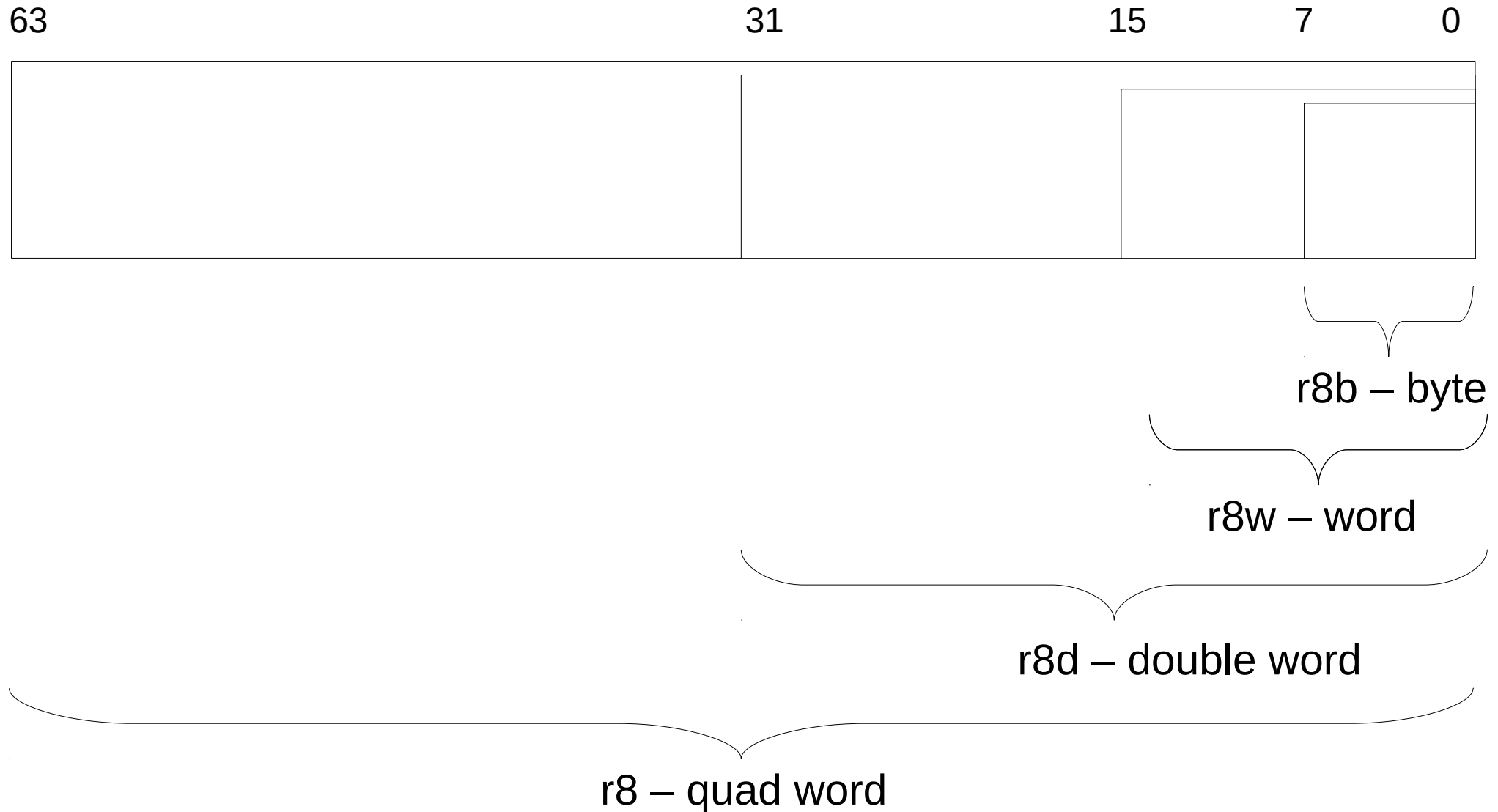
Arquitectura x86_64

CPU

Memoria

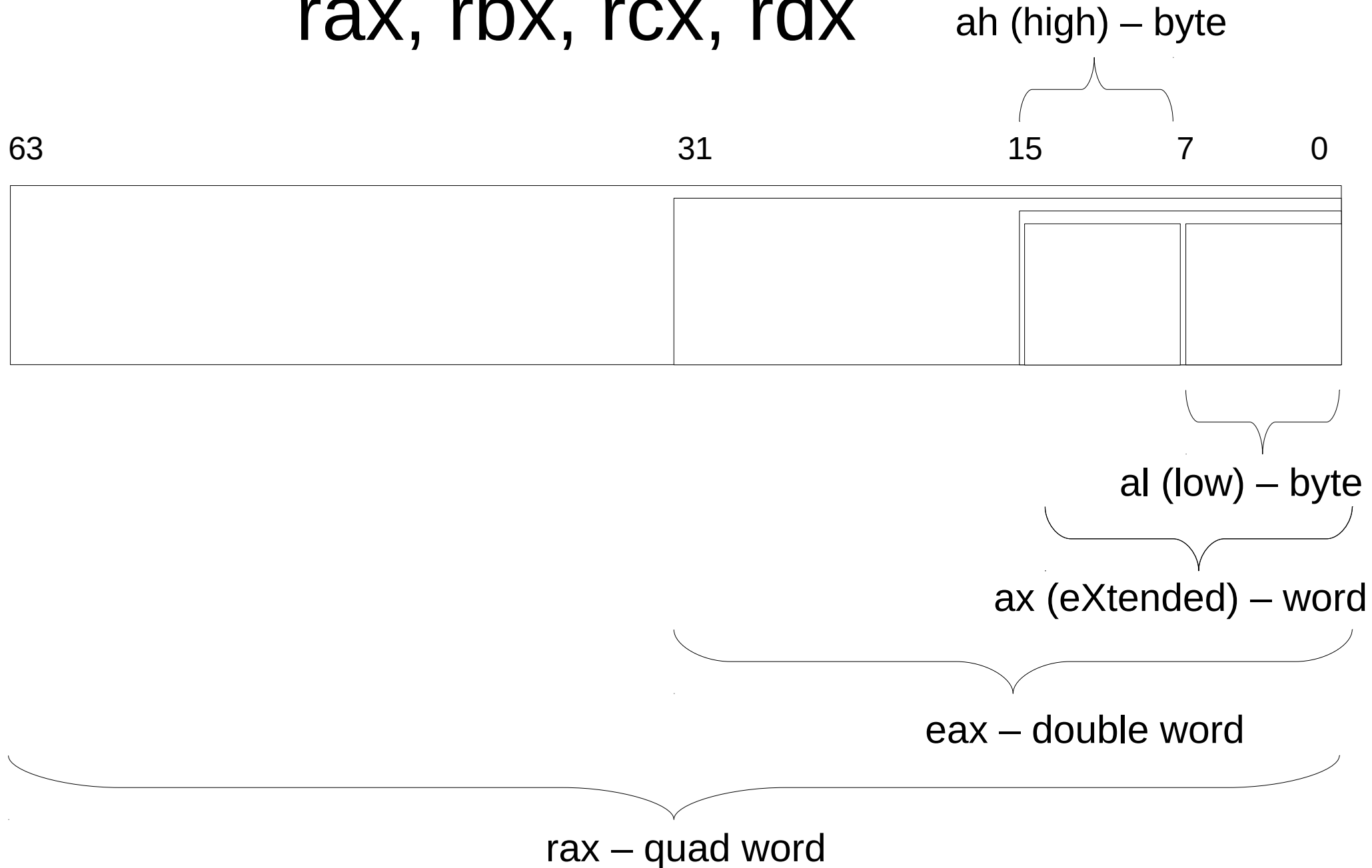


Tamaño de datos y sub-registros r8 a r15



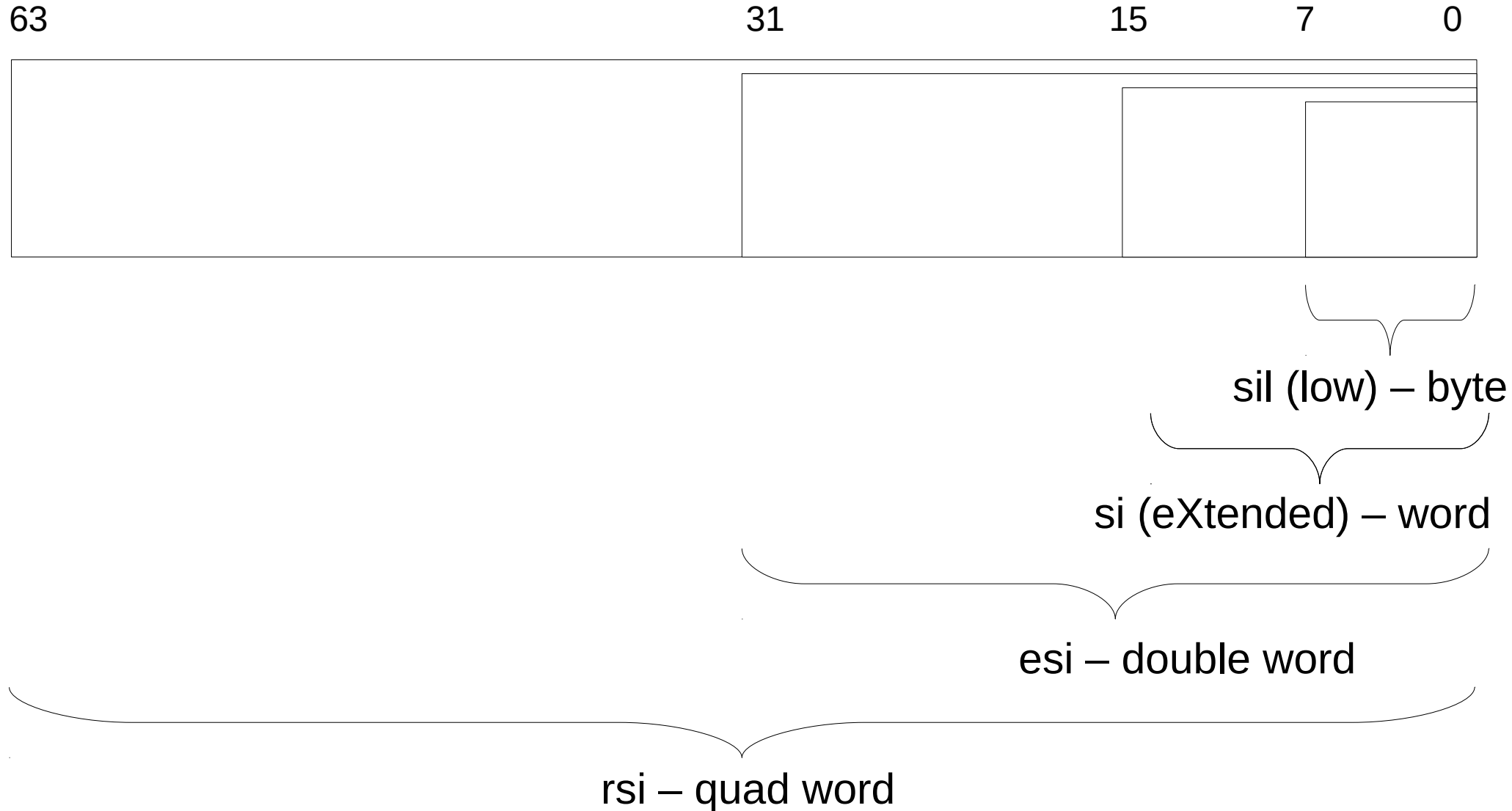
Tamaño de datos y sub-registros

rax, rbx, rcx, rdx



Tamaño de datos y sub-registros

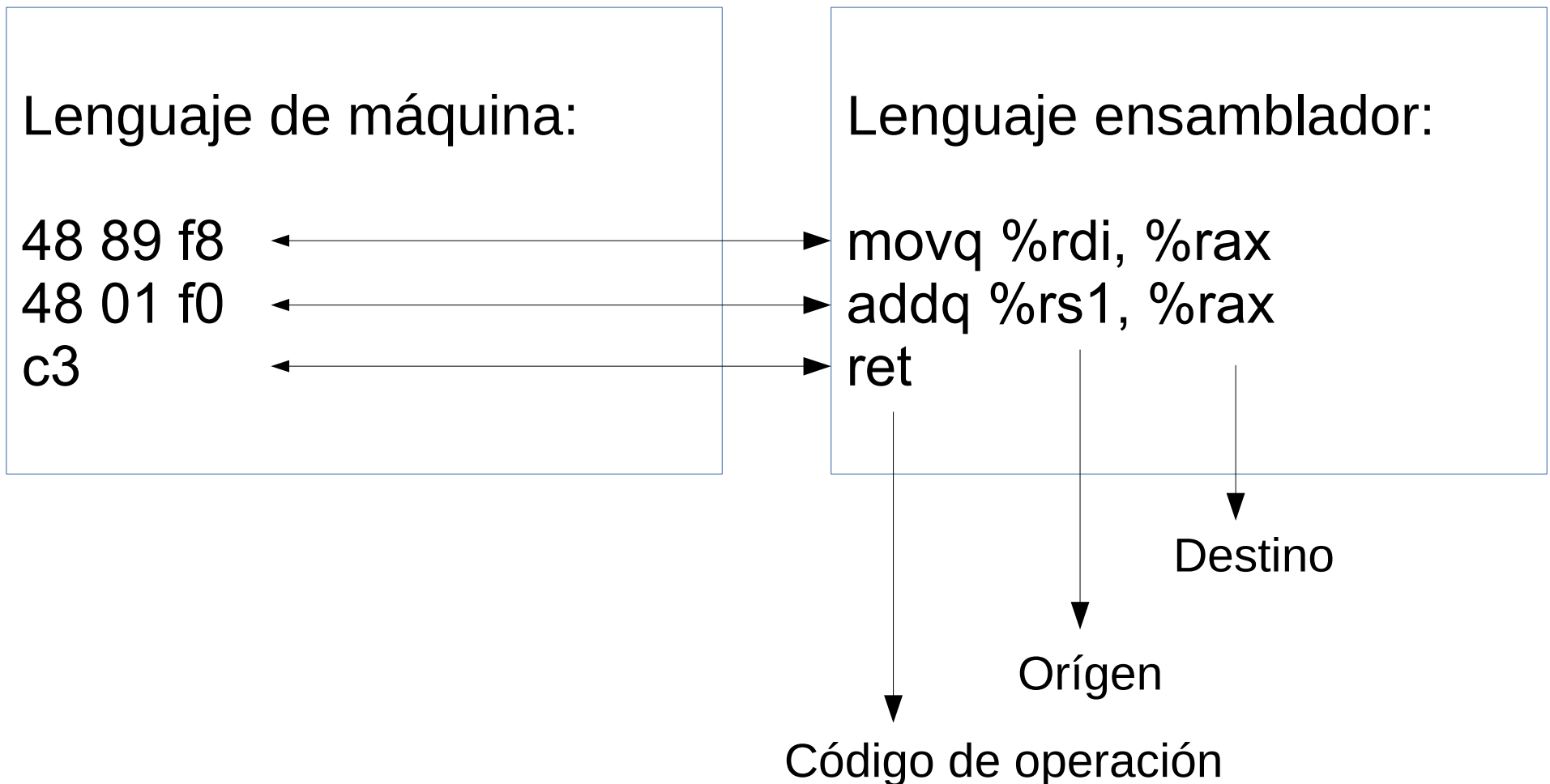
rsi, rdi, rbp y rsp



Big Endian/Little Endian

`demo_endianness.c`

Lenguaje de máquina y lenguaje ensamblador



Lenguaje ensamblador

esto es un comentario

% registro, ej.: %rax

\$ constante, ej.: \$5

Número: dirección de memoria. ej.: 3000

Sufijos: b, s o w, l q, t (ten bytes)

(expresion): lo apuntado por expresión

- Sintáxis completa en el apunte

Lenguaje ensamblador, directivas

Comienzan con un punto: '.'

- .data, .text: de aquí en adelante lo que viene irá a ese segmento
- .global etiqueta
- .ascii, .asciz: sigue en esta línea una lista de cadenas ascii (sin/con nulo final). Cada elemento de la lista se separa con espacios. Ej.:
 .ascii "Hola mundo"
 .asciz "a" "b"

Más directivas para valores

Todas pueden tomar de uno a más elementos.

- `.byte 'a' 97`
- `.double`
- `.float`
- `.short, .long, .quad`
- `.space 123 # lugar, sin inicializar`
- `.space 123,0 # lugar, relleno con ceros`