

# Endiannes - Ejemplos

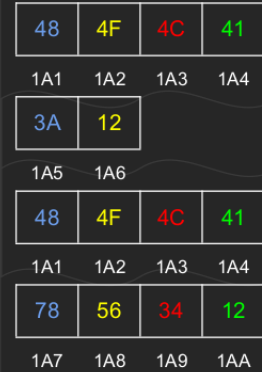
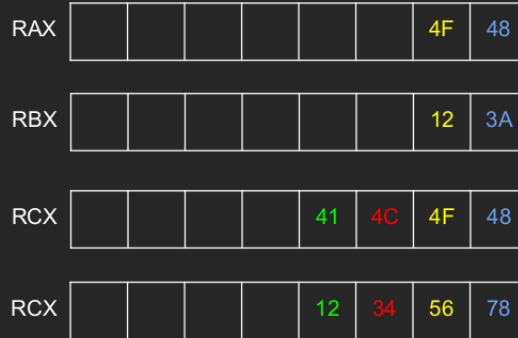
- **Caso 3:** Se ejecuta una copia de memoria a registro

mov AX, [msg]

mov BX, [num1]

mov ECX, [msg]

mov EDX, [num2]



La parte alta del registro contiene el byte de orden superior de memoria, y la parte baja del registro contiene el byte de orden inferior.

## Registros Generales

### Acumulador

operando de instrucciones aritméticas y lógicas

### Base

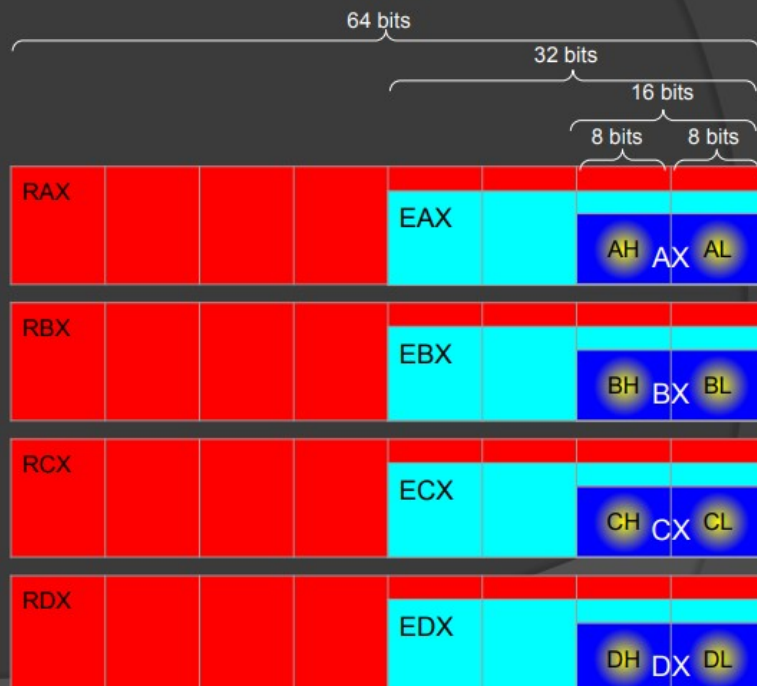
direccionamiento de operandos

### Contador

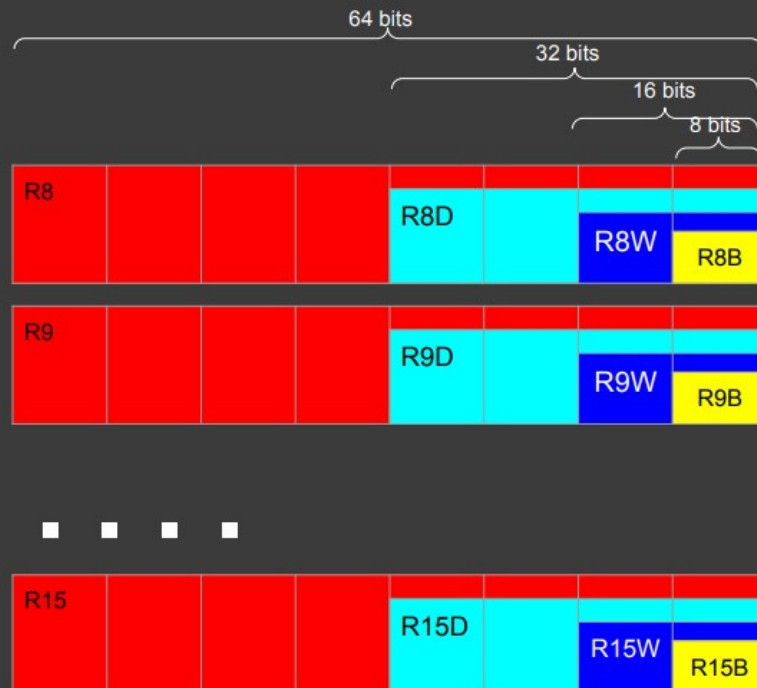
operaciones aritméticas o de string

### Data

operaciones que requieren duplas de registros



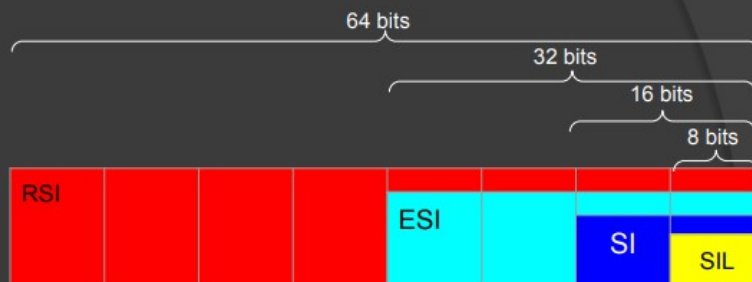
## Registros Generales



## Registros Indice

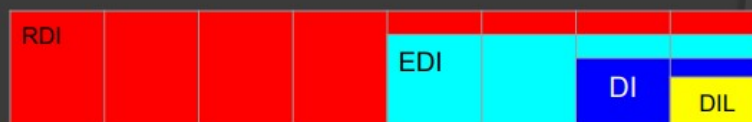
### Source

operaciones de manejo de cadenas para apuntar al operando "origen"

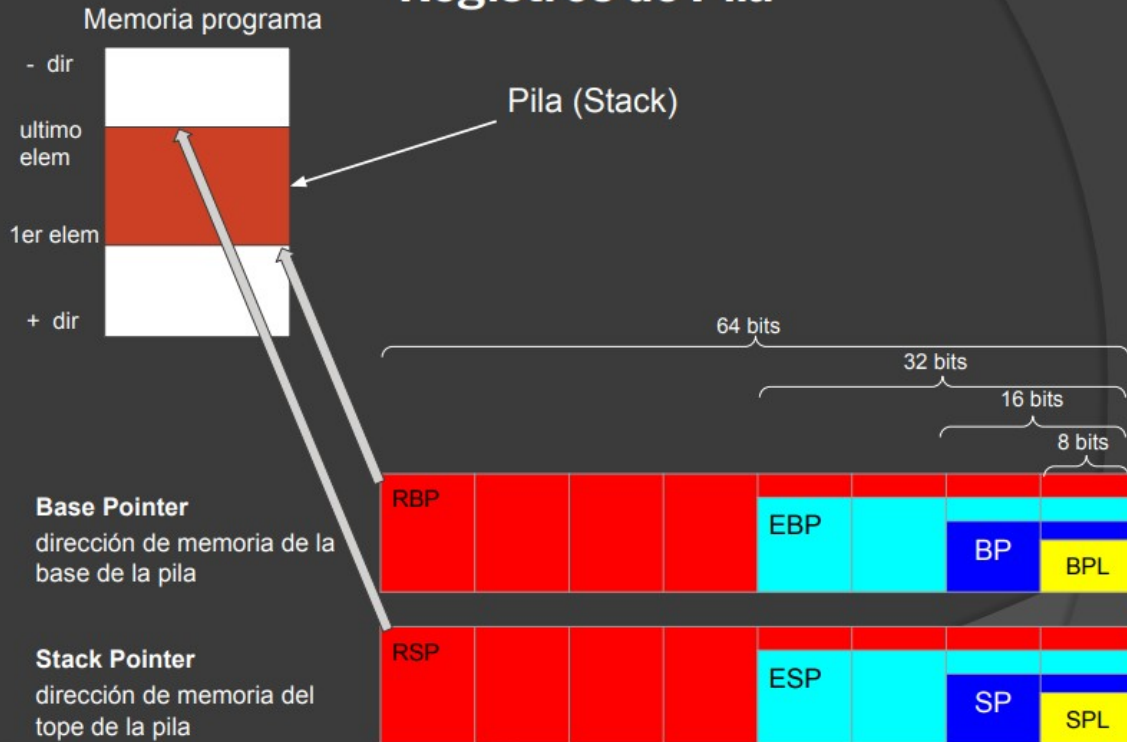


### Destination

operaciones de manejo de cadenas para apuntar al operando "destino"



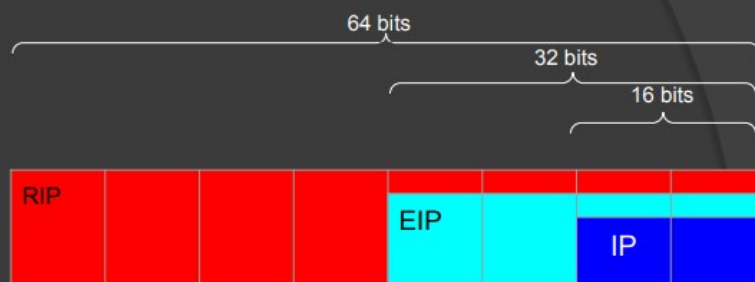
## Registros de Pila



## Registros de Instrucción y Control

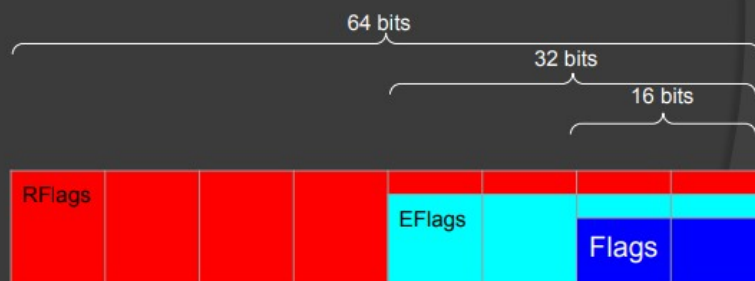
### Instruction Pointer

Actúa como registro contador de programa (PC) y contiene la dirección efectiva de la instrucción siguiente que se ha de ejecutar



### Flags

se usa para almacenar el estado general de la CPU; indica el resultado de la ejecución de cada instrucción



- **Celda de Memoria:** 1 Byte
- **Palabra :** 2 Bytes
- **Doble Palabra :** 4 Bytes
- **Cuádruple Palabra :** 8 Bytes

## Directivas al ensamblador / Pseudo-instrucciones

Son instrucciones para que el ensamblador tome alguna acción durante el proceso de ensamblado. No se traducen a código máquina.

|   |  |
|---|--|
| <code>section</code>                      | indica el comienzo de un segmento  |
| <code>global</code>                       | indica que una etiqueta declarada en el programa es visible para un programa externo                             |
| <code>extern</code>                       | indica que una etiqueta usada en el programa pertenece a un programa externo (donde habrá sido declarada global) |
| <code>db, dw, dd, dq, dt</code>           | sirven para definir áreas de memoria (variables) con contenido inicial   |
| <code>resb, resw, resd, resq, rest</code> | sirven para definir áreas de memoria (variables) sin contenido inicial   |
| <code>times</code>                        | repite una definición la cantidad de veces que se indica   |
| <code>%macro %endmacro</code>             | indican el inicio y final de un bloque para definir una macro  |
| <code>%include</code>                     | permite incluir el contenido de un archivo   |

### Definición y reserva de campos en memoria

- Con contenido inicial (en `section .data`)
  - `db` define byte (1 byte)
  - `dw` define word (2 bytes)
  - `dd` define double (4 bytes)
  - `dq` define quad (8 bytes)
  - `dt` define ten (10 bytes)
- Sin contenido inicial (en `section .bss`)
  - `resb` reserve byte (1 byte)
  - `resw` reserve word (2 bytes)
  - `resd` reserve double (4 bytes)
  - `resq` reserve quad (8 bytes)
  - `rest` reserve ten (10 bytes)

