[Introducción al lenguaje ensamblador 5](#_Toc190276567)

[Lenguajes de programación 5](#_Toc190276568)

[Clasificación 5](#_Toc190276569)

[Según paradigma de programación 5](#_Toc190276570)

[Imperativo 5](#_Toc190276571)

[Declarativo 5](#_Toc190276572)

[Según forma de ejecución 5](#_Toc190276573)

[Compilados 5](#_Toc190276574)

[Interpretados 5](#_Toc190276575)

[Según nivel de abstracción 5](#_Toc190276576)

[Alto nivel 5](#_Toc190276577)

[Mediano nivel 5](#_Toc190276578)

[Bajo nivel 5](#_Toc190276579)

[Lenguajes de maquina 5](#_Toc190276580)

[Lenguaje ensamblador 5](#_Toc190276581)

[Evolución 5](#_Toc190276582)

[Generaciones 5](#_Toc190276583)

[Primera 5](#_Toc190276584)

[Segunda 5](#_Toc190276585)

[Tercera 6](#_Toc190276586)

[Cuarta 6](#_Toc190276587)

[Quinta 6](#_Toc190276588)

[Lenguajes orientados a la inteligencia artificial 6](#_Toc190276589)

[Sexta 6](#_Toc190276590)

[Lenguaje Ensamblador 6](#_Toc190276591)

[Aplicaciones típicas 6](#_Toc190276592)

[En los inicios 6](#_Toc190276593)

[En la actualidad 6](#_Toc190276594)

[Desambiguación 6](#_Toc190276595)

[Lenguaje de maquina 6](#_Toc190276596)

[Lenguaje ensamblador 6](#_Toc190276597)

[Características 6](#_Toc190276598)

[Ventajas 6](#_Toc190276599)

[Desventajas 6](#_Toc190276600)

[Organización de la maquina 7](#_Toc190276601)

[Arquitectura del computador 7](#_Toc190276602)

[Definición 7](#_Toc190276603)

[Arquitectura clásica 7](#_Toc190276604)

[Origen 7](#_Toc190276605)

[Características 7](#_Toc190276606)

[Estructura Básica 7](#_Toc190276607)

[CPU 7](#_Toc190276608)

[CU 7](#_Toc190276609)

[ALU 7](#_Toc190276610)

[Registros 7](#_Toc190276611)

[Memoria 8](#_Toc190276612)

[Memoria Principal 8](#_Toc190276613)

[Definición 8](#_Toc190276614)

[Segmentos 8](#_Toc190276615)

[Definición 8](#_Toc190276616)

[Segmentos de código 8](#_Toc190276617)

[Segmento de datos 8](#_Toc190276618)

[Segmento extra 8](#_Toc190276619)

[Segmento de pila 8](#_Toc190276620)

[Entrada / Salida 8](#_Toc190276621)

[Sistema de Interconexión (Buses) 8](#_Toc190276622)

[Buses de datos 9](#_Toc190276623)

[Bus de control 9](#_Toc190276624)

[Bus de alimentación 9](#_Toc190276625)

[Buses de direcciones 9](#_Toc190276626)

[Periféricos 9](#_Toc190276627)

[Organización de la maquina 9](#_Toc190276628)

[Registros 9](#_Toc190276629)

[Definición 9](#_Toc190276630)

[Clasificación 9](#_Toc190276631)

[Por el microprocesador 9](#_Toc190276632)

[Por el usuario 9](#_Toc190276633)

[Control y estado 9](#_Toc190276634)

[Memory Address Register 9](#_Toc190276635)

[Memory Data Register 9](#_Toc190276636)

[Instriuction Register 10](#_Toc190276637)

[Instruction Pointer 10](#_Toc190276638)

[Visibles al usuario 10](#_Toc190276639)

[De segmentos 10](#_Toc190276640)

[Segmentos de Código (CS) 10](#_Toc190276641)

[Segmento de Datos (DS) 10](#_Toc190276642)

[Segmento Extra (ES) 10](#_Toc190276643)

[Segmento de Pila (SS) 10](#_Toc190276644)

[De propósito general 10](#_Toc190276645)

[Acumulador (AX) 10](#_Toc190276646)

[Base (BS) 10](#_Toc190276647)

[Contador (CX) 10](#_Toc190276648)

[Datos (DX) 10](#_Toc190276649)

[De Apuntadores e Índices 11](#_Toc190276650)

[Apuntador de pila (SP) 11](#_Toc190276651)

[Apuntador Base (BP) 11](#_Toc190276652)

[Índice de Fuente (SI) 11](#_Toc190276653)

[Índice de Destino (DI) 11](#_Toc190276654)

[De banderas 11](#_Toc190276655)

[Acarreo (C) 11](#_Toc190276656)

[Paridad (P) 11](#_Toc190276657)

[Cero (Z) 11](#_Toc190276658)

[Signo (S) 11](#_Toc190276659)

[Interrupción (I) 11](#_Toc190276660)

[Desbordamiento (O) 11](#_Toc190276661)

[Direcciones 11](#_Toc190276662)

[Composición 11](#_Toc190276663)

[Direccion de memoria 11](#_Toc190276664)

[Programación en lenguaje ensamblador 12](#_Toc190276665)

[Diferenciación 12](#_Toc190276666)

[Turbo Assembler 12](#_Toc190276667)

[Estructura 12](#_Toc190276668)

[Elementos 12](#_Toc190276669)

[Palabras reservadas 12](#_Toc190276670)

[Modelo 12](#_Toc190276671)

[Tipos 12](#_Toc190276672)

[Tiny 12](#_Toc190276673)

[Small 12](#_Toc190276674)

[Compact 12](#_Toc190276675)

[Médium 12](#_Toc190276676)

[Large 13](#_Toc190276677)

# Introducción al lenguaje ensamblador

## Lenguajes de programación

### Clasificación

#### Según paradigma de programación

##### Imperativo

Se ordena a la computadora como realizar una tarea siguiendo una serie de pasos o instrucciones

##### Declarativo

Se le indica a la computadora que se desea obtener o que se está buscando

#### Según forma de ejecución

##### Compilados

Se ejecuta pasando por uno o varios procesos de traducción para que sea comprensible por la computadora

##### Interpretados

Aquellos que se ejecutan instrucción por instrucción, haciendo una traducción de un lenguaje determinado a un idioma que la computadora comprenda

#### Según nivel de abstracción

##### Alto nivel

Lenguajes que no dependen de la arquitectura del computador, entonces si son capaces de migrar de un computador a otro

##### Mediano nivel

*Punto medio entre los lenguajes de alto nivel y de bajo nivel*, ciertas cualidades lo acercan al lenguaje humano (alto nivel)

##### Bajo nivel

Lenguajes que son dependientes de la arquitectura del computador, no pueden migrarse o utilizarse en otras computadoras sin antes recodificar

###### Lenguajes de maquina

Da órdenes directas a la máquina, operaciones fundamentales en código binario (código maquina) que consiste únicamente en ceros y unos

###### Lenguaje ensamblador

Derivado del lenguaje máquina y formado por abreviaturas de letras y números. Trabajado a nivel de instrucciones, ósea programas al más fino detalle.

### Evolución

#### Generaciones

##### Primera

Lenguaje de maquina

##### Segunda

Lenguaje ensamblador

##### Tercera

Lenguajes de alto nivel funcionales

##### Cuarta

Lenguajes de alto nivel parametrizables

##### Quinta

###### Lenguajes orientados a la inteligencia artificial

##### Sexta

Lenguajes orientados a la computación cuántica

## Lenguaje Ensamblador

### Aplicaciones típicas

#### En los inicios

* Sistemas operaticvos IBM PC DOS
* Lotus 1, 2, 3
* Atari 800
* Mega Drive / Genesis
* Super Nintendo Entertainment System
* Sega Saturn

#### En la actualidad

* Controladores de dispositivos
* Sistemas embebidos
* Sistemas de tiempo real
* Programación de hardware
* Hacking

### Desambiguación

#### Lenguaje de maquina

Basado en el código binario “0” y “1”, el único que una computadora es capaz de interpretar

#### Lenguaje ensamblador

Basado en “nemónicos” que permite una programación por parte del usuario en un lenguaje más comprensivo por la persona, pero ligado al microprocesador o microcontrolador

### Características

|  |  |
| --- | --- |
| Ventajas | Desventajas |
| * Mayor velocidad de ejecución * Poca demanda de recursos * Control preciso de tareas * Ejecutada directamente sobre el hardware * Única traducción | * Programación compleja para el usuario * Programas extensos * Poca portabilidad * Errores irreparables |

# Organización de la maquina

## Arquitectura del computador

### Definición

Es el modelo y descripción funcional de lo que se requiere y el diseño que se implementara, en varias partes de la computadora, con interés especial en el como el CPU trabaja y accede a las direcciones de memoria

### Arquitectura clásica

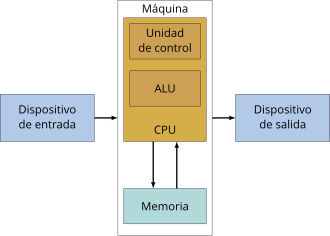
#### Origen

Basado en el trabajo del matemático John Von Neumann y divulgado en 1945 en la Moore School de la Universidad de Pensilvania, USA

#### Características

* Procesamiento de datos
* Almacenamiento de datos
* Transferencia de datos

#### Estructura Básica



##### CPU

Unidad central de procesos que dirige todas las funciones de la computadora, como un cerebro

###### CU

Unidad de control de leer la memoria de las instrucciones a ejecutar y las secuencias en el acceso de datos y operaciones a realizar

###### ALU

Unidad aritmético – lógica que transforma los datos a través de una serie de módulos que realizan operaciones aritméticas y lógicas

###### Registros

Almacén de la configuración interna del CPU, información de la última operación del ALY y resultado de la ejecución de instrucciones

##### Memoria

Responsable del almacenamiento de datos, como los recuerdos. Los datos e instrucciones son códigos binarios interpretadas por el CU

###### Memoria Principal

Definición

Es como una matriz de celdas en las que se almacena de forma aleatoria organizadas en “palabras” (conjunto de celdas, cada celda almacena un bit) que definen las instrucciones

Segmentos

Definición

Área específica de la memoria, determinada por el modelo del programa (64k bytes usualmente) destinada a un fin determinado.

Segmentos de código

Sección de memoria que tiene las instrucciones y procedimientos utilizados por el programa

El registro del este segmento (cs) define la direccion inicial en que inicia el segmento de código

Segmento de datos

Sección de memoria que contiene la mayor parte de los datos que el programa utilizara, se tiene acceso al segmento estableciendo la direccion en la que inicia el mismo en su respectivo registro (ds)

Segmento extra

Sección de memoria utilizada ocasionalmente para algunas instrucciones de cadena o cuando se trabaja con registros extendidos. Se ubica en el respectivo registro (es) al necesitarse, varios microprocesadores también cuentan con otros registros para segmentos adicionales (fs, gs)

Segmento de pila

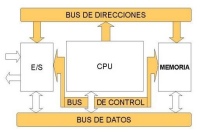
Sección de memoria destina al arreglo de pila con que el microprocesador trabajara, la ubicación inicial de entrada a la pila se determina por el registro apuntador de pila (sp)

##### Entrada / Salida

Transferencia de datos entre el interior y exterior de la computadora, como los sentidos. Ahí se encuentran los controladores de periféricos que forman la interfaz entre ellos, la memoria y el procesador

##### Sistema de Interconexión (Buses)

Permite el flujo de datos entre el CPU, la memoria y los módulos de entrada/salida, como los nervios.



###### Buses de datos

Sirve para transmitir daros entre los distintos dispositivos del computador

###### Bus de control

Sirve para seleccionar el emisor y receptor de la transacción del bus

###### Bus de alimentación

Sirve para proporcionar los distintos voltajes a los dispositivos

###### Buses de direcciones

Sirve para indicar la posición del dato a la que se quiere acceder

##### Periféricos

Dispositivos necesarios para brindar datos a la computadora o visualizar los resultados, como los órganos sensoriales. Estos se conectan con un bus especial a su controlador o al módulo de entrada/salida

# Organización de la maquina

## Registros

### Definición

Estructura con capacidades de almacenamiento temporal, trabaja directamente con el CU y se mueve a la velocidad del ALU

### Clasificación

#### Por el microprocesador

* MAR
* MDR
* IR
* IP

#### Por el usuario

* De propósito general
* De apuntadores e índices
* De banderas
* De segmento

### Control y estado

#### Memory Address Register

Registro con alta velocidad y capacidad limitada, conectado al bus de direcciones y conteniendo la direccion del dato que se quiere leer o escribir

#### Memory Data Register

Registro con alta velocidad y mayor capacidad, conectado al bus de datos y conteniendo lo que el CPU lee o escribe en la memoria de o hacia un puerto de entrada o salida

#### Instriuction Register

Contiene la instrucción que se está ejecutando, cada instrucción se carga, decodifica, prepara y luego ejecuta, este es un registro de la UC

#### Instruction Pointer

Registro apuntador (visible pero no modificable por el usuario) que contiene la posición de memoria de la próxima instrucción a ejecutar

### Visibles al usuario

#### De segmentos

##### Segmentos de Código (CS)

Contiene la direccion en que inicia el segmento de código

##### Segmento de Datos (DS)

Contiene la direccion en que inicia el segmento de datos

##### Segmento Extra (ES)

Contiene la direccion en que inicia el segmento extra

##### Segmento de Pila (SS)

Contiene la direccion en que inicia el segmento de pila

#### De propósito general

##### Acumulador (AX)

Usualmente conserva el resultado temporal después de una operación lógica o aritmética

* EAX
* AH
* AL

##### Base (BS)

Usualmente conserva el desplazamiento de los datos que hay en la memoria

* EBX
* BH
* BL

##### Contador (CX)

Usualmente contiene el conteo de ciertas instrucciones para:

* Corrimientos (CL)
* Rotaciones del número de bytes (CX)
* Contador de LOOP (CX y ECX)

##### Datos (DX)

De uso general que contiene la parte más significativa del producto luego de una multiplicación o del dividendo antes de una división

* EDX
* DH
* DL

#### De Apuntadores e Índices

##### Apuntador de pila (SP)

Se utiliza para direccionar datos a una pila de memoria lifo (ESP)

##### Apuntador Base (BP)

Apuntador de uso general (EBP)

##### Índice de Fuente (SI)

Se usa para direccionar datos fuente de forma directa y utilizarlo con las instrucciones string o arrays (ESI)

##### Índice de Destino (DI)

Se usa para direccionar datos destino de forma indirecta y usarlos como instrucciones de string o array (EDI)

#### De banderas

##### Acarreo (C)

Indica un acarreo después de una suma o un “préstamo” en la resta

##### Paridad (P)

Es un “0” para una impar y “1” para par

##### Cero (Z)

Indica el resultado de una operación aritmética o lógica, z=1 es que el resultado es 0, z=0 es que el resultado no es 0

##### Signo (S)

Indica el signo aritmético del resultado después de una suma o resta, si s=1 el resultado es negativo, s=0 el resultado es positivo

##### Interrupción (I)

Indica si se llevó a cabo o se ignoró por una entrada externa

##### Desbordamiento (O)

Indica que el resultado de la operación aritmética excede la capacidad de la maquina

## Direcciones

### Composición

#### Direccion de memoria

Una direccion de segmento y una direccion de desplazamiento

Una direccion ubicada en un registro de segmento define la direccion inicial de cualquier segmento de memoria de 64K bytes

La direccion de desplazamiento selecciona una localidad dentro de esos 64K bytes

# Programación en lenguaje ensamblador

## Diferenciación

### Turbo Assembler

Ensamblador para programas con datos de 8 y 16 bits compatible con Windows y con enlazador turbo linker, procesadores intel 8086 – 80286

## Estructura

### Elementos

* Directivas
* Identificadores
* Instrucciones

#### Palabras reservadas

1. Nemónicos generales
   1. MOV
   2. LEA
   3. etc
2. Directivas
   1. .model
   2. .stack
   3. Etc
3. Operadores
   1. ADD
   2. SUB
   3. Etc
4. Simbolos predefinidos
   1. @DATA
   2. Etc

#### Modelo

Identificador especifico de palabras reservadas que indica al ensamblador el tamaño de los segmentos de código y datos (si son necesarios) y define la estructura del programa en lenguaje ensamblador

##### Tipos

###### Tiny

Datos y código en un solo segmento de **64K bytes en total**

###### Small

Datos y código en segmentos independientes de **64K bytes c/u**

###### Compact

Un segmento único de código, mientras que los datos crean nuevos segmentos conforme se estén llenando. **64K bytes c/u**

###### Médium

Un segmento único de datos, mientras que el código crea nuevos segmentos conforme se están llenando. **64K bytes c/u**

###### Large

Datos y código crean nuevos segmentos conforme se van llenando. **64K bytes c/u**

#### Directivas

* Indican al ensamblador la estructura del programa
* Inicializa los segmentos
* Se conforman de palabras reservadas
  + .data
  + .model
  + .stack
  + .80286

#### Identificadores

Nombres que se le asignas a un elemento especifico del programa por el usuario

#### Instrucciones

* Nemónicos e identificadores que en conjunto crean y ejecutan una acción del programa (no distingue mayúsculas de minúsculas)

#### Algoritmo

##### Pasos

1. Definición del modelo
   1. .model
2. Definición del segmento de Datos y/o Código
   1. .data
   2. .code
3. Definición del segmento de pila
   1. .stack
4. Finalización del programa
   1. end