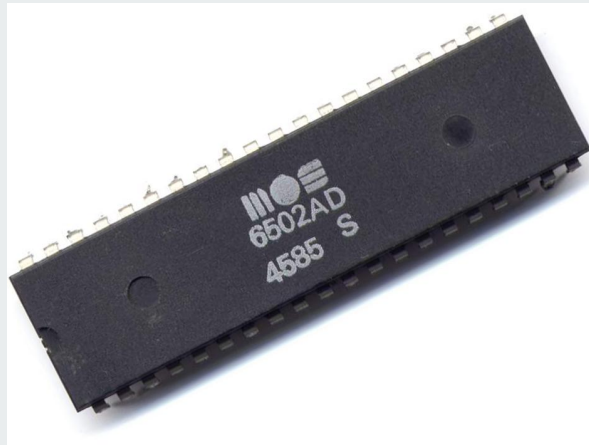




# Assembler 6502



*Make Classic Games*



# Índice

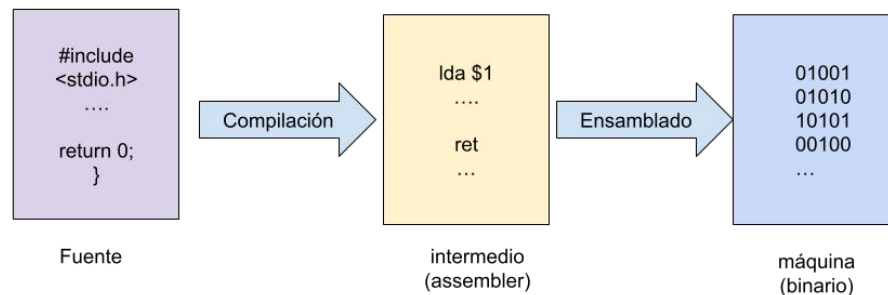
1. Introducción
2. 6502
  - a. Procesador 6502
  - b. Arquitectura del procesador
  - c. Consolas y Computadoras que utilizan 6502
3. Conjunto Instrucciones 6502
  - a. Aritméticas
  - b. memoria
  - c. salto
  - d. rotación y desplazamiento
  - e. llamadas y retorno
4. Entorno de Desarrollo
  - a. Editor de código
  - b. CC65
  - c. Emuladores
5. Bibliografía y referencias

# 1. Introducción

Vamos a recordar qué es el ensamblador, y por qué es importante...

El código ensamblador es un código nemotécnico que permite dar instrucciones al propio procesador; siendo un código muy óptimo.

Cuando “compilamos” un programa, el compilador genera un código intermedio; muchas veces es pasado a ensamblador (no siempre el código intermedio es ensamblador) y posteriormente, ensamblado para generar un binario.

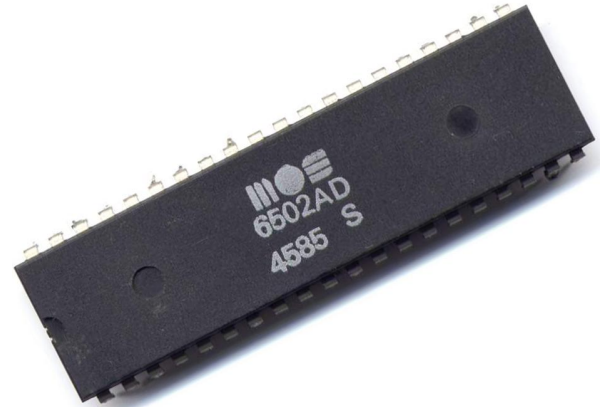


---

## 2. 6502

El procesador MOS 6502 es un procesador de 8 bits que se diseñó en el año 1975. En un principio se diseñó para ser compatible con el procesador Motorola 6800 (diseñado por parte del equipo original); pero tuvo que cambiarse debido a problemas legales.

Posteriormente, tuvo varias iteraciones y ha sido utilizado en infinidad de proyectos; tanto de forma doméstica como otros usos (se utiliza para marcapasos).





## 2. 6502

El procesador 6502, es un procesador de 8 bits, con un bus de dirección de 16 bits. Tiene 3510 transistores.

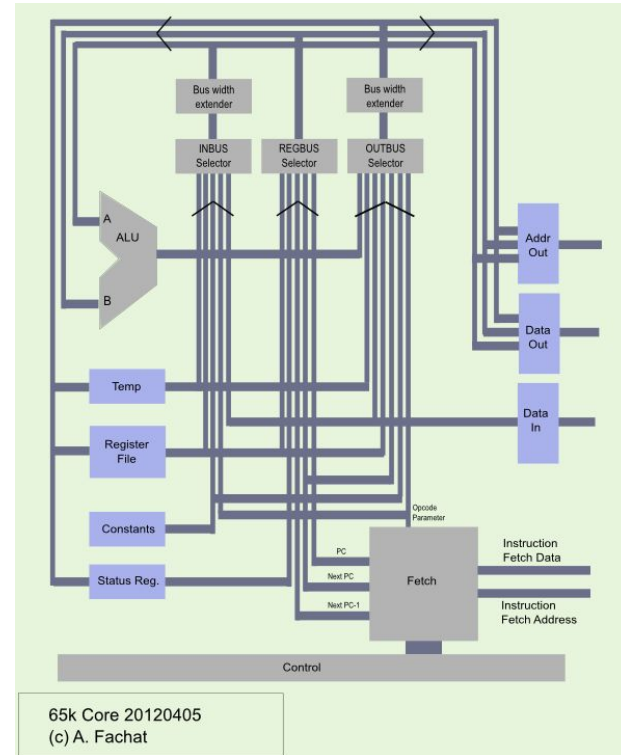
Tiene pocos registros. Registro acumulador *A* de 8 bits, registros índices *X* e *Y*, puntero a pila *SP* de 8 bits, registro de estado *SR* y por último el contador de programa *PC* de 16 bits.



## 2. 6502

### Arquitectura

Vamos a ver la arquitectura de este procesador.





## 2. 6502

### Registros:

- **A:** Acumulador
- **PC:** Contador Programa
- **X,Y:** Registros Índice
- **SP:** Puntero a pila
- **SR:** Registro de estado



## 2. 6502

### Registro Estado del procesador:

- **C:** Acarreo
- **Z:** Zero
- **I:** Interrupt Disable
- **D:** Decimal Mode
- **B:** Break Command
- **V:** OverFlow Command
- **N:** Negative Flag

Hay algunas instrucciones que modifican el estado del procesador.





## 2. 6502

### Modos Direccionamiento

- Implícita (Dirección implícita).
- Inmediata (Dirección inmediata).
- Absoluta (Dirección Absoluta dependiendo del valor).
- Relativa (Dirección Relativa a la dirección actual).
- Acumulador (Dirección establecida a partir del valor en el registro acumulador).
- Indirecto X, Indirecto Y (Utiliza la dirección de la instrucción actual más el registro índice).
- Indexado X, Indexado Y (Utiliza la dirección especificada más el registro índice).

---

## 2. 6502

### Máquinas y consolas con 6502

- Apple 2/Apple 3
- Atari 2600
- Atari 7800
- NES
- Commodore 64 / Commodore PET...
- Oric 1/Oric Atmos





## 3. Conjunto de Instrucciones

Podemos dividir el conjunto de instrucciones del procesador 6502, en los siguientes apartados:

- Operaciones de Memoria
- Transferencia de Registros
- Operaciones de Pila
- Operaciones Lógicas
- Aritméticas
- Incrementos y Decrementos
- Desplazamientos
- Salto y Llamadas
- Ramas
- Cambios Status Flag
- Funciones del Sistema



## 3. Conjunto de Instrucciones

### Operaciones de Memoria

LDA: Carga en acumulador (afecta N y Z)

LDX: Carga en X (afecta N y Z)

LDY: Carga en Y (afecta N y Z)

STA: Almacena Acumulador en memoria

STX: Almacena X en memoria

STY: Almacena Y en memoria.



## 3. Conjunto de Instrucciones

### Transferencia entre registros

TAX: Transfiere Acumulador a X (afecta N y Z)

TAY: Transfiere Acumulador a Y (afecta N y Z)

TXA: Transfiere X a Acumulador (afecta N y Z)

TYX: Transfiere Y a Acumulador (afecta N y Z)



## 3. Conjunto de Instrucciones

### Operaciones de Pila

El 6502 tiene 256 bytes para almacenar en pila entre las posiciones \$0100 y \$01FF Además tiene un registro especial S para saber el espacio de pila.

TSX: Transfiere el puntero a pila a X (Afecta a N y Z).

TXS: Transfiere el valor de X al puntero a Pila

PHA: Almacena el acumulador en la pila

PHP: Almacena el estado del procesador en la pila

PLA: Obtiene el último valor de la pila y lo guarda en el acumulador (Afecta a N y Z).

PLP: Obtiene el estado del procesador de la pila y lo guarda en el puntero a pila (Afecta a todo el estado).



## 3. Conjunto de Instrucciones

### Operaciones Lógicas

Todas las operaciones son entre el acumulador y un valor en memoria

AND: Operación Lógica Y (Afecta a N y Z).

EOR: Exclusive Or (Afecta a N y Z).

ORA: Or Lógico (Afecta a N y Z).

BIT: Bit test (Afecta a N, V y Z)



## 3. Conjunto de Instrucciones

### Operaciones Aritméticas

ADC: Suma con acarreo (Afecta a N,V,Z y C).

SBC: Resta con acarreo (Afecta a N,V,Z y C).

CMP: Compara el acumulador (Afecta a N,Z y C).

CPX: Compara el registro X (Afecta a N,Z y C).

CPY: Compara el registro Y (Afecta a N,Z y C).





## 3. Conjunto de Instrucciones

### Incrementos y Decrementos

INC: Incrementa una localización de memoria (Afecta N y Z).

INX: Incrementa el registro X (Afecta N y Z).

INY: Incrementa el registro Y (Afecta N y Z).

DEC: Decrementa una localización de memoria (Afecta N y Z).

DEX: Decrementa el registro X (Afecta N y Z).

DEY: Decrementa el registro Y (Afecta N y Z).



## 3. Conjunto de Instrucciones

### Desplazamiento

ASL: Desplazamiento aritmético a la izquierda (Afecta N, Z y C).

LSE: Desplazamiento Lógico a la derecha (Afecta N, Z y C).

ROL: Rotar a la izquierda (Afecta N, Z y C).

ROE: Rotar a la derecha (Afecta N, Z y C).



## 3. Conjunto de Instrucciones

### Llamadas y Saltos

JMP: Salta a una localización en concreto.

JSR: LLama a una subrutina.

RTS: Termina subrutina y vuelve a la anterior llamada.



## 3. Conjunto de Instrucciones

Ramas (Saltos condicionales)

BCC: Salto si no hay acarreo.

BCS: Salto si hay acarreo.

BEQ: Salta si el resultado es 0.

BMI: Salta si el resultado es negativo.



## 3. Conjunto de Instrucciones

### Ramas (Saltos condicionales) II

BNE: Salta si NO es 0.

BPL: Salta si No es negativo.

BVC: Salta si no hay desbordamiento.

BVS: SALta si hay desbordamiento.



## 3. Conjunto de Instrucciones

### Cambios de Estado

CLC: Borra el bit de acarreo.

CLD: Borra el bit de modo decimal.

CLI: Borra el bit de deshabilitar interrupción.

CLV: Borra el bit de desbordamiento.

SEC: Establece el bit de acarreo.

SED: Establece el bit de modo decimal.

SEI: Establece el bit de deshabilitar interrupciones.



## 3. Conjunto de Instrucciones

### Funciones del Sistema

BRK: Forzar interrupción

NOP\*: No operación (no hacer nada).

RTI: Retornar desde interrupción.

\* En algunas implementaciones el uso de esta instrucción podría dar problemas.



## 4. Entorno de Desarrollo

Como hemos comentado en otras ocasiones, es importante utilizar un entorno de desarrollo que nos permita implementar de forma fácil y cómoda los programas que crearemos; para ello podremos utilizar:

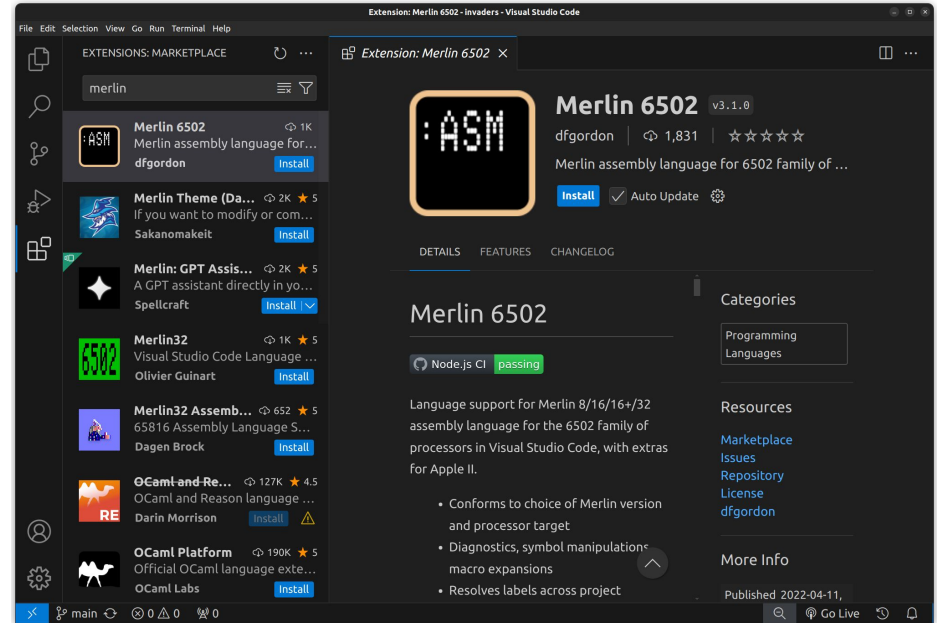
- Editor de Código
- Programa Ensamblador.
- Emulador/es



## 4. Entorno de Desarrollo

### Editor de Código

Tener un editor de código que nos ayude es importante; por ello recomendamos utilizar vscode con le extensión *merlin 6502*.





## 4. Entorno de Desarrollo

### Programa ensamblador

Es necesario tener un programa que pase del código que escribamos a código binario; para ello recomendamos la suite CC65 que no solo incluye un ensamblador sino también compilador de C, depurador, enlazador (linker), etc.. todo para la familia del 6502.

Más información: <https://cc65.github.io/>



## 4. Entorno de Desarrollo

### Emulador

Aunque esto dependerá de la máquina con la que vayamos a trabajar, existen muchos emuladores para diferentes máquinas con el procesador 6502; dejamos información de algunas:

- VICE (Commodore 64)
- POWER64 (Commodore 64)
- APPLEWIN (Apple II)
- PERDITA (Durango)



## 4. Entorno de Desarrollo

### Emulador

Aunque esto dependerá de la máquina con la que vayamos a trabajar, existen muchos emuladores para diferentes máquinas con el procesador 6502; dejamos información de algunas:

- VICE (Commodore 64)
- POWER64 (Commodore 64)
- APPLEWIN (Apple II)
- PERDITA (Durango)



## 5. Bibliografía y Recursos

- Programación retro para el Commodore 64 Vol.1: [Compra en Amazon.](#)
- Wikipedia: [https://es.wikipedia.org/wiki/MOS\\_6502](https://es.wikipedia.org/wiki/MOS_6502)
- 6502.org: <http://www.6502.org/>
- CC65: <https://cc65.github.io/>
- Merlin 6502: [Merlin 6502 - Visual Studio Marketplace](#)
- Emuladores Apple II: <https://www.zophar.net/apple2.html>
- Durango: <https://durangoretro.com>