

# **Programación en ASM: 8085**

Diego Enrique Fontán, CosasDePuma

2017-09-24

# Índice

|   |          |
|---|----------|
| <b>1. Operaciones de transferencia básicas</b>      | <b>2</b> |
| 1.1. Inmediato (MVI) . . . . .                      | 2        |
| 1.2. Por Registro (MOV) . . . . .                   | 2        |
| 1.3. Directa . . . . .                              | 4        |
| 1.4. Indirecta (MOV M) . . . . .                    | 4        |
| <b>2. Operaciones aritméticas básicas</b>           | <b>4</b> |
| 2.1. Por Registro . . . . .                         | 5        |
| 2.1.1. Suma/Adición (ADD) . . . . .                 | 5        |
| 2.1.2. Resta/Sustracción (SUB) . . . . .            | 6        |
| 2.2. Inmediato . . . . .                            | 6        |
| 2.2.1. Suma/Adición (ADI) . . . . .                 | 6        |
| 2.2.2. Resta/Sustracción (SUI) . . . . .            | 6        |
| 2.3. Directa . . . . .                              | 6        |
| 2.4. Indirecta . . . . .                            | 7        |
| 2.4.1. Suma/Adición (ADD M) . . . . .               | 7        |
| 2.4.2. Resta/Sustracción (SUB M) . . . . .          | 7        |
| <b>3. Operaciones aritméticas unitarias básicas</b> | <b>7</b> |
| 3.1. Inmediato . . . . .                            | 8        |
| 3.1.1. Incremento (INR) . . . . .                   | 8        |
| 3.1.2. Decremento (DCR) . . . . .                   | 8        |
| <b>4. Ejemplos de código</b>                        | <b>9</b> |
| 4.1. Máscaras . . . . .                             | 9        |

## **Licencia y propósito de este documento**

Este documento, creado por Diego Enrique Fontán (CosasDePuma), pretende ser un apoyo para todo aquel que esté estudiando o esté interesado en aprender a cómo programar en ensamblador 8085. Se da por hecho que el lector ha adquirido una placa similar al procesador 8085 Intel o que se ha apropiado de un software simulador.

No calificaré de manual a este escrito, dado que tiene una finalidad más modesta. Es por ello que lo descrito aquí no serán más que un par de apuntes y recordatorios que el autor compartirá con cualquier interesado.

La distribución y copia de este documento, así como la edición del mismo, está permitida siempre y cuando el autor original reciba las menciones correspondientes y sin que nadie se lucre a costa de este documento.

# 1. Operaciones de transferencia básicas

Las operaciones de transferencia son aquellas que nos permiten mover información.

Aquí es importante ver que podemos operar de varias maneras:

| Tipo de operación | Destino  | Origen            |
|-------------------|----------|-------------------|
| Inmediato         | Registro | Valor             |
| Por Registro      | Registro | Registro          |
| Directa           | Registro | Dirección         |
| Indirecta         | Registro | Puntero (Memoria) |

## 1.1. Inmediato (MVI)

El método de transferencia inmediato se usa para definir el valor de un Registro asignándole un valor.

Este comando ocupa dos bytes: el primero para el opcode de la instrucción y el segundo para el valor en hexadecimal que queremos que sea asignado.

| Instrucción | Opcode |
|-------------|--------|
| MVI A,valor | 3E     |
| MVI B,valor | 06     |
| MVI C,valor | 0E     |
| MVI D,valor | 16     |
| MVI E,valor | 1E     |
| MVI H,valor | 26     |
| MVI L,valor | 2E     |
| MVI M,valor | 36     |

## 1.2. Por Registro (MOV)

El método de transferencia por Registro sirve para copiar el valor de un Registro A a un Registro B, siendo el primero el Registro Destino y, el se-

gundo, el Registro Origen.

Este comando ocupa un byte y tiene un opcode diferente según los Registros implicados.

| Instrucción | Opcode | Instrucción | Opcode | Instrucción | Opcode |
|-------------|--------|-------------|--------|-------------|--------|
| MOV A,A     | 7F     | MOV B,A     | 47     | MOV C,A     | 4F     |
| MOV A,B     | 78     | MOV B,B     | 40     | MOV C,B     | 48     |
| MOV A,C     | 79     | MOV B,C     | 41     | MOV C,C     | 49     |
| MOV A,D     | 7A     | MOV B,D     | 42     | MOV C,D     | 4A     |
| MOV A,E     | 7B     | MOV B,E     | 43     | MOV C,E     | 4B     |
| MOV A,H     | 7C     | MOV B,H     | 44     | MOV C,H     | 4C     |
| MOV A,L     | 7D     | MOV B,L     | 45     | MOV C,L     | 4D     |

| Instrucción | Opcode | Instrucción | Opcode | Instrucción | Opcode |
|-------------|--------|-------------|--------|-------------|--------|
| MOV D,A     | 57     | MOV E,A     | 5F     | MOV H,A     | 67     |
| MOV D,B     | 50     | MOV E,B     | 58     | MOV H,B     | 60     |
| MOV D,C     | 51     | MOV E,C     | 59     | MOV H,C     | 61     |
| MOV D,D     | 52     | MOV E,D     | 5A     | MOV H,D     | 62     |
| MOV D,E     | 53     | MOV E,E     | 5B     | MOV H,E     | 63     |
| MOV D,H     | 54     | MOV E,H     | 5C     | MOV H,H     | 64     |
| MOV D,L     | 55     | MOV E,L     | 5D     | MOV H,L     | 65     |

| Instrucción | Opcode |
|-------------|--------|
| MOV L,A     | 57     |
| MOV L,B     | 50     |
| MOV L,B     | 51     |
| MOV L,B     | 52     |
| MOV L,B     | 53     |
| MOV L,B     | 54     |
| MOV L,B     | 55     |

### 1.3. Directa

Las operaciones de transferencia directas que existen en 8085 son algo complicadas, así que se dejará para apartados más avanzados.

### 1.4. Indirecta (MOV M)

Las operaciones de transferencia indirectas son aquellas que nos permiten interactuar con la dirección de Memoria a la que apunta el Registro HL.

Este comando ocupa un byte y tiene un opcode diferente según los Registros implicados.

| Instrucción | Opcode | Instrucción | Opcode |
|-------------|--------|-------------|--------|
| MOV A,M     | 7E     | MOV M,A     | 77     |
| MOV B,M     | 46     | MOV M,B     | 70     |
| MOV C,M     | 4E     | MOV M,C     | 71     |
| MOV D,M     | 56     | MOV M,D     | 72     |
| MOV E,M     | 5D     | MOV M,E     | 73     |
| MOV H,M     | 66     | MOV M,H     | 74     |
| MOV L,M     | 56     | MOV M,L     | 75     |

## 2. Operaciones aritméticas básicas

Las operaciones aritméticas básicas son aquellas que nos permiten operar el Acumulador.

Dentro de esta sección definiremos operaciones tales como Suma/Adición y Resta/Sustracción.

De nuevo, estas operaciones pueden ser de varios tipos:

| Tipo de operación | Destino    | Origen            |
|-------------------|------------|-------------------|
| Inmediato         | Acumulador | Valor             |
| Por Registro      | Acumulador | Registro          |
| Directa           | Acumulador | Dirección         |
| Indirecta         | Acumulador | Puntero (Memoria) |

Dado que muchas de estas operaciones implican operaciones con bits de estado (flags), se mostrarán solamente las más básicas, dejando el resto para apartados más avanzados.

## 2.1. Por Registro

Las operaciones aritméticas por Registro nos permiten sumar o restar el valor de un Registro al Acumulador.

Estas operaciones ocupan un byte y el opcode varía según el Registro implicado.

### 2.1.1. Suma/Adición (ADD)

| Instrucción | Opcode |
|-------------|--------|
| ADD A       | 87     |
| ADD B       | 80     |
| ADD C       | 81     |
| ADD D       | 82     |
| ADD E       | 83     |
| ADD H       | 84     |
| ADD L       | 85     |

### 2.1.2. Resta/Sustracción (SUB)

| Instrucción | Opcode |
|-------------|--------|
| SUB A       | 97     |
| SUB B       | 90     |
| SUB C       | 91     |
| SUB D       | 92     |
| SUB E       | 93     |
| SUB H       | 94     |
| SUB L       | 95     |

## 2.2. Inmediato

Las operaciones aritméticas inmediatas nos permiten añadir o sustraer una cantidad determinada en el Acumulador.

Este comando ocupa dos bytes: el primero para el opcode de la instrucción y el segundo para el valor en hexadecimal que queremos que sea añadido.

### 2.2.1. Suma/Adición (ADI)

| Instrucción | Opcode |
|-------------|--------|
| ADI valor   | C6     |

### 2.2.2. Resta/Sustracción (SUI)

| Instrucción | Opcode |
|-------------|--------|
| SUI valor   | D6     |

## 2.3. Directa

Las operaciones aritméticas directas son algo complicadas, así que como ya hemos comentado, se dejará para apartados más avanzados.



## 2.4. Indirecta

Las operaciones aritméticas indirectas son aquellas que nos permiten interactuar con la dirección de Memoria a la que apunta el Registro HL.

Estas operaciones ocupan un byte.

### 2.4.1. Suma/Adición (ADD M)

| Instrucción | Opcode |
|-------------|--------|
| ADD M       | 86     |

### 2.4.2. Resta/Sustracción (SUB M)

| Instrucción | Opcode |
|-------------|--------|
| SUB M       | 96     |

## 3. Operaciones aritméticas unitarias básicas

Las operaciones aritméticas unitarias son aquellas que modifican en una unidad el valor del Registro con el que se esté trabajando.

Aquí podemos distinguir dos operaciones distintas: incremento y decremento.

Sólamamente existen dos tipos de comandos según su interacción:

| Tipo de operación | Destino           |
|-------------------|-------------------|
| Inmediato         | Registro          |
| Indirecta         | Puntero (Memoria) |

### 3.1. Inmediato

Las operaciones aritméticas unitarias inmediatas son las modificar un Registro directamente.

Ocupan un byte y tiene diferentes opcodes según el Registro implicado.

#### 3.1.1. Incremento (INR)

| Instrucción | Opcode |
|-------------|--------|
| INR A       | 3C     |
| INR B       | 04     |
| INR C       | 0C     |
| INR D       | 14     |
| INR E       | 1C     |
| INR H       | 24     |
| INR L       | 2C     |

#### 3.1.2. Decremento (DCR)

| Instrucción | Opcode |
|-------------|--------|
| DCR A       | 3D     |
| DCR B       | 05     |
| DCR C       | 0D     |
| DCR D       | 15     |
| DCR E       | 1D     |
| DCR H       | 25     |
| DCR L       | 2D     |

## 4. Ejemplos de código

### 4.1. Máscaras

**Enunciado:** A partir de la posición de memoria 0E300h, cada uno de los 25 bytes siguientes representa la información de un alumno en una determinada materia. Cada byte contiene la siguiente información:

- El bit de mayor peso, b7, indica si es primera matrícula o no.
- Los siguientes dos bits, b6 y b5, representan el curso más alto en el que está matriculado (00 = 1º, 01 = 2º, 10 = 3º y 11 = 4º).
- Los 5 bits de menor peso (b4, b3, b2, b1 y b0) representan el valor numérico entero de su nota en la materia (de 0 a 10).

**Ejercicio:** cuente el número de alumnos en primera matricula, cuyo curso sea 3º o inferior y tengan una nota igual a 5.

**Código:**

```
# ORG E000
# BEGIN E000

; requisitos del alumno
MASCARA: MVI B, C5H

; lugar en memoria de los alumnos
INICIO:    MVI D,05
          LXI H,FC00H

; comprobamos si cumple los requisitos
CONTAR:    MOV A,M
          CMP B
          JNZ SIGUIENTE
          INR C

; pasamos al siguiente si quedan mas
SIGUIENTE: INX H
          DCR D
          JNZ CONTAR

; modificamos los requisitos
          MOV A,B
```

```
SUI 20H
MOV B,A
CPI 65H
JNZ INICIO
HLT

# ORG FC00H
# DB 75H, 85H, A5H, 19H, C5H
```