Tema 2:
Segmentación y modos de direccionamiento

- Segmentación
- Modos de direccionamiento
- Direccionamiento relativo
- Instrucciones de transferencia de control
- Instrucciones aritméticas: comparación
- Ejemplo de acceso a una cadena mediante:
  - Direccionamiento relativo a base
  - Direccionamiento relativo mediante índice

Bibliografía básica

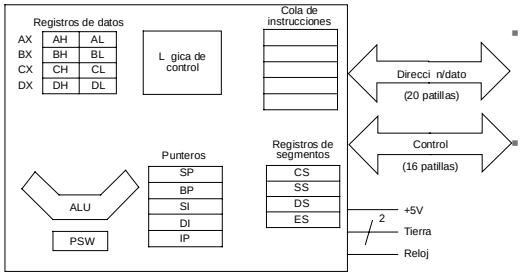
- 8088-8086/8087 programación ensamblador en entorno MS-DOS
  Miguel Angel Roselló.
  Ed. Anaya Multimedia

- Microprocesadores: el 8088 / 86
  Fernando Remiro Domínguez
  Agustín Martín García
  Ed. Akal-Bibiloteca tecnológica

- Lenguajes ensambladores
  R. Martínez Tomás. Ed. Paraninfo

- Lenguaje ensamblador de los 80x86
  Jon Beltrán de Heredia
  Editorial Anaya-Multimedia. 1996

Segmentación del i8086



- El microprocesador 8086 tiene catorce registros de 16 bits. Con 16 bits se puede acceder a 2^{16} o lo que es igual a 64 K
- El 8086 emplea un truco para acceder a 1 MB = 2^{20}
- El truco consiste en dividir el mega en trozos de 64 K que llama segmentos
- El programa en todo momento debe conocer en qué segmento están los datos o el código y cuál es la posición dentro del segmento

- El cálculo de la dirección física lo realiza según:
  - R.Base x 10h + desplazamiento
- Los registros de segmento son
  - CS: para el segmento de código
  - DS: para el segmento de datos
  - SS: para el segmento de pila
  - ES: segmento extra de datos
- Gracias a la segmentación se facilita la multiprogramación y existen zonas diferentes para el código, para los datos y para la pila

Modos de direccionamiento

Modos de direccionamiento			µP 8086/88	Ejemplos
Inmediato			Inmediato	MOV AX, 15H
	Directo	De registro	A registro	MOV AX, BX
		De memoria	(con segmentación)	
		De página base	Directo	MOV CX, ETIQUETA
	Relativo	Al contador de programa	(sólo saltos)	
		A un registro	Relativo a base	MOV [BX]+ARTÍCULO, AL
		A un registro índice	Mediante índice	MOV DL, VECTOR[SI]
			Mediante índice y base	MOV AH, [BX][SI]+ARRAY
		A pila	(Relativo a pila)	
Indirecto			(No existe)	
Implícito			Algunas instrucciones	

Direccionamiento relativo

- Se emplea para apuntar a direcciones de memoria dentro de un segmento
- Se emplean registros Base y registros Índices. Si BX se emplea como registro base, entonces el registro de segmento que se emplea es el registro DS. Si es BP el registro base, entonces el registro de segmento empleado es el de la pila SS
- Existen diferentes modos de direccionamiento indirecto en el 8086:

Ejemplos:

- ADD DX, [BX]
- MOV DL, [BP+6]
- XOR rojo[BX], DX

Modo de direccionamiento	Sintaxis	Descripción.
Indirecto a registro	[BX] [BP][DI][SI]	Los registros contienen la dirección efectiva .
Relativo a base	desplaz[BX] desplaz[BP]	La dirección efectiva se calcula a partir del registro y se le suma el desplazamiento.
Mediante índice	desplaz[DI] desplaz[SI]	
Mediante índice y base	[BX][DI] [BX][SI] [BP][DI] [BP][SI]	La dirección efectiva se calcula a partir del registro base y el registro índice.
Mediante índice y base con desplazamiento.	Desp[BX][DI] Desp[BX][SI] Desp[BP][DI] Desp[BP][SI]	La dirección efectiva se calcula a partir del registro base, del índice y del desplazamiento.

Instrucciones de transferencia de control (I)

- Modifican la secuencia normal de ejecución de un programa
- Las instrucciones que actúan sobre el contador de programa (PC) controlan la secuencia de ejecución de un programa. Son un caso especial de transferencia, donde el operando destino es PC
- Clasificación:
- Saltos Incondicionales: **JMP** etiqueta ⇨ IP ← etiqueta

  Condicionales: **J{condición}** etiqueta ⇨ Si condición, IP ← etiqueta. Si no, IP ← sig.I.
- Saltos incondicionales: siempre se produce el salto
- Saltos condicionales: se realiza el salto si se da la condición sobre los *flags*

Instrucciones de transferencia de control (II)

[N]

- **Z** - Cero
- **E** – Igual
- **C** - Acarreo
- **S** - Signo
- **O** - *Overflow* (Desbordamiento)

[N] [E]

- **G** - Mayor que
- **L** - Menor que
- **A** - Superior*
- **B** - Inferior*

- **P** - Paridad
- **PE** - Paridad par
- **CXZ** - CX=0
- **PO** - Paridad impar

*Se refieren a operandos sin signo

Instrucciones de transferencia de control (III)

- **Nombre:** **LOOP**
- **Formato:** **LOOP desplazamiento**
- **Descripción:**

  Si CX es diferente de cero, entonces IP = IP + desplazamiento. Si CX es cero entonces ejecuta la instrucción siguiente. El desplazamiento debe estar comprendido entre -128 y 127

  Mediante esta instrucción es posible implementar bucles. También son factibles los bucles anidados pero debemos hacer uso de la pila

- **Ejemplos:**

```
MOV CX, contador        ; CX = nº. de veces que ejecuta el bucle
BUCLE:
        Instrucciones dentro del bucle
LOOP BUCLE      ; CX = CX -1. Si CX ≠ 0 saltar a Bucle
```

Instrucciones aritméticas: comparación

- **Nombre:** **CMP**
- **Formato:** **CMP destino, origen**
- **Descripción:**

  Compara los operandos origen y destino modificando el registro de flags

  Realmente lo que hace es realizar la resta de ambos operandos, pero no almacena el resultado. Así, si el resultado es negativo, origen será mayor que destino. Si el resultado es positivo origen será menor que destino, y si cero serán iguales

- **Ejemplos:**

```
CMP AX, DX      ; Compara los operandos AX y DX
CMP CL, 'A'     ; Compara CL con el código ASCII del carácter A
CMP DL, [BX]    ; Compara DL con el contenido de la posición de memoria
                ; apuntada por BX
```


Ejemplo de acceso a una cadena mediante direccionamiento relativo a base

```
DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
numeros  DB 1,2,3,4,5,6 ; Números de la cadena.
.CODE
Inicio:
MOV AX, @DATA       ; Indicamos donde se
MOV DS, AX          ; encuentran los datos

LEA BX, numeros ; En DS:BX dirección cadena
MOV CX, 6       ; Indicamos el número de dígitos
Bucle:
MOV DL, [BX]        ; Llevamos a DL el valor
                    ; del número
ADD DL, 7           ; Le sumamos 7
MOV [BX], DL        ; Dejamos el número
                    ;actualizado
INC BX              ; Incrementamos BX en 1
LOOP Bucle

MOV AH, 4Ch ;Solicitamos al sistema operativo
INT 21h     ;la terminación de nuestro programa
END Inicio
```

Ejemplo de acceso a una cadena mediante direccionamiento relativo mediante índice

```
DOSSEG
.MODEL SMALL
.STACK 100h
.DATA
numeros  DB  1,2,3,4,5,6; Números de la cadena.
.CODE
Inicio:
MOV AX, @DATA        ; Indicamos donde se
MOV DS, AX           ; encuentran los datos

MOV  SI, 0       ; Ponemos el registro SI a 0
MOV  CX, 6       ; Indicamos el número de dígitos
 Bucle:
   ADD numeros[SI], 7
      INC SI
      LOOP Bucle
MOV AH, 4Ch; Solicitamos al sistema operativo
INT 21h      ; la terminación de nuestro programa
END Inicio
```