

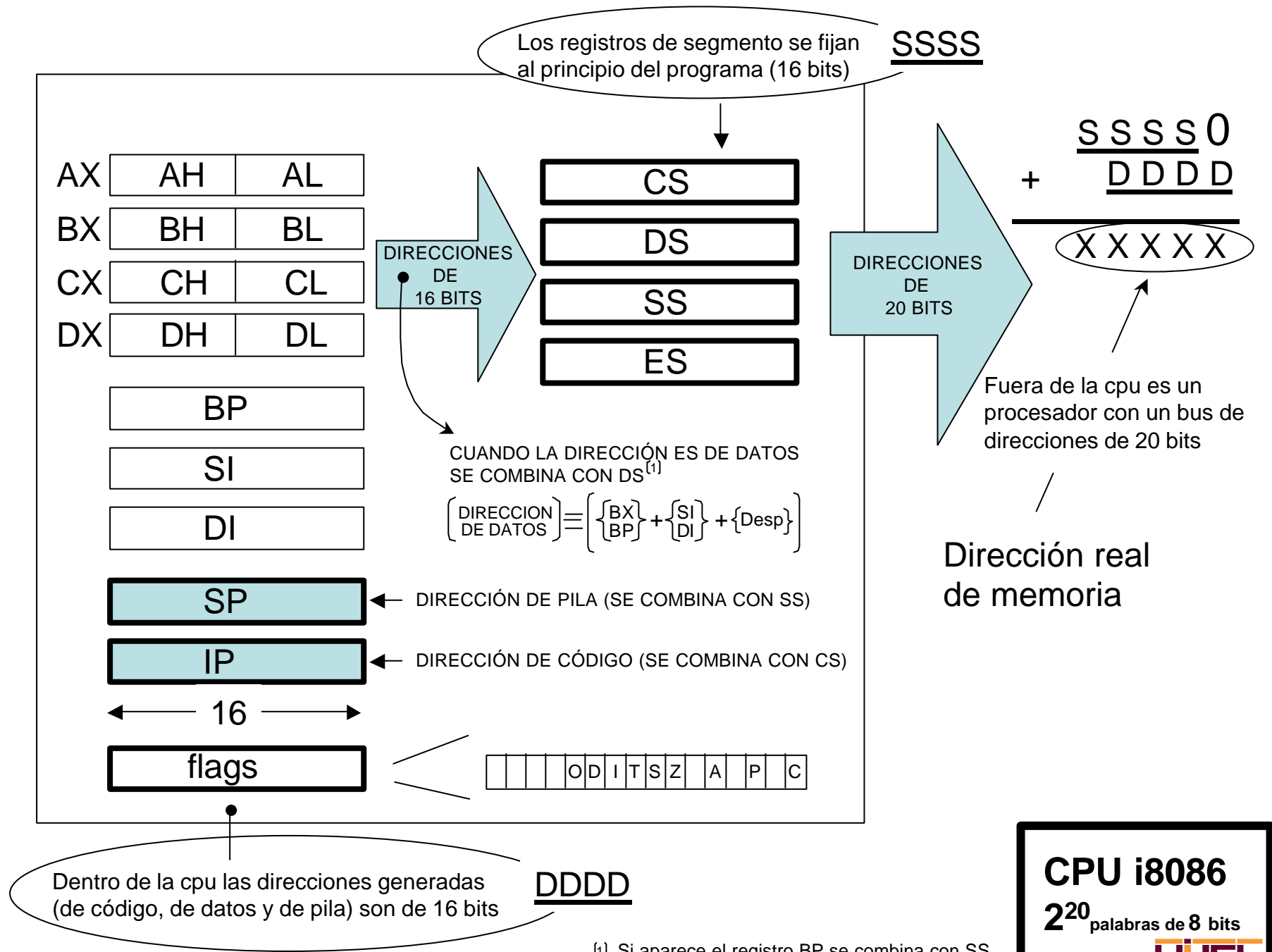
Arquitectura intel 8086

Preámbulo de OSO para alumnos formados en el procesador MIPS

- 1. Introducción**
- 2.- Direccionamiento y Registros de Segmento**
- 3.- La pila**
- 4.- El registro de flags**
- 5.- Instrucciones de salto condicional**
- 6.- Instrucciones de salto incondicional**
- 7.- Instrucciones de llamada y retorno de procedimiento**
- 8.- Instrucciones LDS R16, Mem y LES R16, Mem**
- 9.- Instrucción LEA R16, Mem**
- 10.- Instrucciones de E/S**
- 11.- Memoria de video**
- 12.- Tabla de instrucciones**

Arquitectura intel 8086

1. Introducción

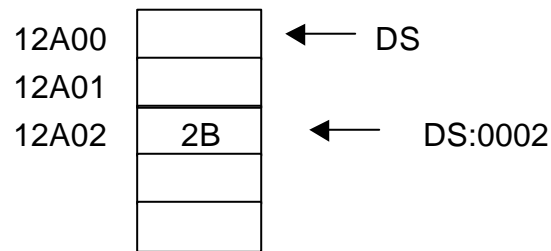


(1) Si aparece el registro BP se combina con SS

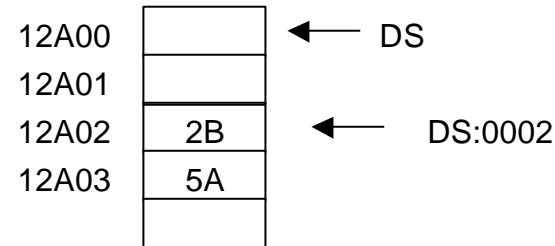
Arquitectura intel 8086

2.- Direccionamiento y Registros de Segmento

- IP se combina con CS *code segment* (apuntador al segmento de código)
- SP se combina con SS *stack segment* (apuntador al segmento de pila)
- Las direcciones de datos (16 bits) se combinan con DS *data segment* (apuntador al segmento de datos)



MOV AL , DS:[0002]



MOV AX , DS:[0002]

Arquitectura intel 8086

2.- Direccionamiento y Registros de Segmento

MOV juan, AX (instrucción *memoria, registro*)

MOV CX, AX (instrucción *registro, registro*)

MOV CX, 012B (instrucción *registro, inmediato*)

Sin embargo en el procesador intel 8086 no existen las instrucciones *memoria, memoria*

~~MOV juan, pedro~~

$$\text{Memoria} = \left[\left\{ \begin{array}{c} \text{BX} \\ \text{BP} \end{array} \right\} + \left\{ \begin{array}{c} \text{SI} \\ \text{DI} \end{array} \right\} + \{\text{Desplazamiento}\} \right]$$

Ojo BP, ver ejemplos

Arquitectura intel 8086

2.- Direccionamiento y Registros de Segmento

- En el caso en que se utilice el direccionamiento *Memoria – inmediato* es necesario indicar la longitud del dato

Ejemplo

```
MOV BYTE PTR [2000],6
```

modificaría la dirección DS:2000

```
MOV WORD PTR [2000],6
```

modificaría las direcciones DS:2000 y DS:2001

- Para introducir un dato inmediato en un registro de segmento previamente hay que introducirlo en un registro R16 (no existe *MOV RegSeg, inmediato*)

Ejemplo

```
MOV AX , 123A
```

```
MOV DS , AX
```

Arquitectura intel 8086

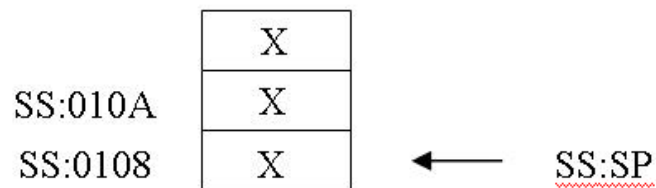
3.- La pila (instrucciones PUSH y POP)

Ejemplo

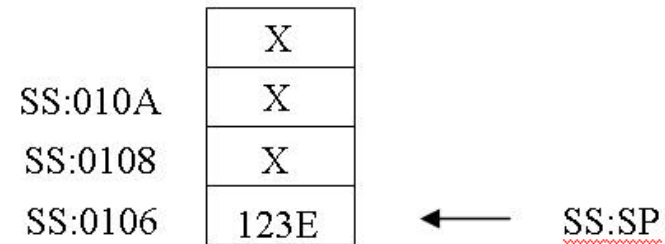
PUSH AX. Almacena en la pila el contenido de AX

Suponemos AX=123E ; SP= 0108

Antes de la ejecución de la instrucción
el estado de la pila es:



Tras la ejecución de PUSH AX:



Con la instrucción **POP BX** retiraríamos de la cima de la pila el dato 123E que quedaría en el registro BX. SP volvería a su antiguo valor 0108.

Arquitectura intel 8086

4.- El registro de flags (sigue)

Flags de estado

- **C:** bit de carry, se activa a 1 si se produce acarreo en una operación aritmética, se activa a 0 si no.
- **P:** bit de paridad, se activa a 1 si el resultado de una operación tiene paridad par, es decir, si el resultado tiene un número par de unos. Se activa a 0 en caso contrario
- **A:** bit carry auxiliar, se activa a 1 si una operación aritmética produce acarreo de peso 16. Se utiliza en las operaciones aritméticas decimales
- **Z:** bit cero, se activa a 1 si una operación produce 0 como resultado. Se activa a 0 en caso contrario
- **S:** bit de signo, se activa a 1 si el bit más significativo de un resultado es 1. Se activa a 0 en caso contrario
- **O:** bit overflow, indica un desbordamiento en una operación aritmética

Arquitectura intel 8086

4.- El registro de flags (sigue)

Flags de control

- **I:** El más importante es el flag I que condiciona a la cpu a atender o no las interrupciones pedidas por los periféricos. El programador pone este bit a 1 mediante la instrucción “sti” y a cero mediante la instrucción “cli”.

Cuando este bit está a 0 la cpu hace caso omiso a las interrupciones (interrupciones inhabilitadas)

Arquitectura intel 8086

5.- Instrucciones de salto condicional

En el procesador intel 8086 las instrucciones de salto condicional evalúan el contenido de los flags y en función del estado realizan o no el salto (éste es un cambio importante frente a MIPS)

Ejemplo

aquí: MOV AX,0001

ADD AX,0002 ; la única de las tres que afecta a los flags

MOV AX,0000

JZ aquí

¿ Qué pasaría con JZ ?

Arquitectura intel 8086

5.- Instrucciones de salto condicional (sigue)

Además de JZ ####, tenemos

JC #### salto si C=1

JO #### salto si O=1

JP #### salto si P=1

JS #### salto si S=1

y las contrarias

JNC #### salto si C=0

JNZ #### salto si Z=0

JNO #### salto si O=0

JNP #### salto si P=0

JNS #### salto si S=0

Es de notar que las estas instrucciones sólo modifican IP (CS quedaría con el mismo valor). En el argot intel esto se conoce como saltos cercanos (NEAR)

Utilizan direccionamiento relativo al IP y para ello gastan un solo byte – salto de -128 a +127 –

también tenemos JA, JL y JG junto con sus contrarias JNA, JNL y JNG

Arquitectura intel 8086

6.- Instrucciones de salto incondicional (JMP)

Salto lejanos (FAR) – modifican CS e IP –

JMP #####:#####

Ejemplo: JMP 123A:569A

JMP FAR Mem

Ejemplo: JMP FAR [2000]

Salto cercanos (NEAR) – modifican IP –

JMP #####

Ejemplo: JMP aqui Utiliza direccionamiento relativo al IP y para ello gasta dos bytes – salto de -32768 a +32767 –

JMP R16

Ejemplo: JMP AX Copia AX en IP

JMP Mem

Ejemplo JMP [BX+ 0200] Copia el contenido de DS:[BX+0200] en IP

Arquitectura intel 8086

7.- Instrucciones de llamada y retorno de procedimiento (CALL y RET)

Llamadas NEAR

CALL #####, CALL R16, CALL Mem antes de saltar almacenan en la pila la dirección de retorno
(sólo IP)

RET retira la dirección de retorno de la pila y la introduce en IP

RET ##### después de hacer ret le suma ##### a SP

Ejemplo

RET 0002 después de RET le sumaría 2 a SP

Llamadas FAR

CALL #####:#####, CALL FAR Mem, antes de saltar almacenan en la pila la dirección de retorno
(apilan primero CS y después IP)

RETF retira la dirección de retorno de la pila (2 elementos pila) y la introduce en IP y CS

RETF ##### después de hacer retf le suma ##### a SP

Ejemplo RETF 0002 después de RETF le sumaría 2 a SP

Arquitectura intel 8086

8.- Instrucciones LDS R16, Mem y LES R16, Mem

Sirven para cargar un puntero FAR en un par de registros

Ejemplo: LDS CX, [2000] modifica los registros DS y CX con los bytes de memoria DS:2000 a DS:2003

LES CX, [2000] modificaría ES y CX

9.- Instrucción LEA R16, Mem

Ejemplo

MOV AX, DS:[0200] carga el contenido de la dirección DS:[0200] en AX

LEA AX, DS:[0200] carga 0200 en AX

Esta instrucción tiene sentido cuando utilizamos en lugar de direcciones de memoria etiquetas

Ejemplo

LEA AX, juan Si juan ocupa las direcciones DS:0200 y DS:0201 la instrucción dejaría el valor 0200 en AX

Arquitectura intel 8086

10.- Instrucciones de E/S

El procesador intel 8086 tiene un bus de E/S (8bits de datos, 16 bits dirección de E/S) y por él puede acceder a 216 direcciones de E/S mediante las instrucciones "IN AL, DX" y "OUT DX,AL". Este espacio de direcciones se reparte entre todos los periféricos conectados por este bus al procesador

Ejemplos

```
MOV DX, 0080
```

```
IN AL, DX      copiaría en el registro AL el contenido del puerto 0080
```

```
MOV DX, 0100
```

```
MOV AL,25
```

```
OUT DX,AL      copiaría en el puerto 0100 el valor 25
```

11.- Memoria de video

El ibm pc consume 4000 bytes de su memoria para manejar el contenido de la pantalla en modo texto (25 líneas de 80 caracteres). Cada carácter gasta 2 bytes

La memoria de video comienza en la dirección B800:0000

Escribir en pantalla es tan sencillo como modificar el primer byte (código ascii) de cada una de las posiciones de pantalla que queremos modificar.