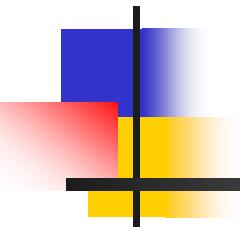
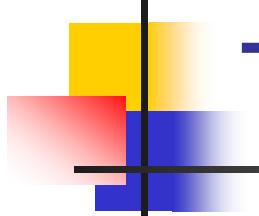


Organización de Computadoras

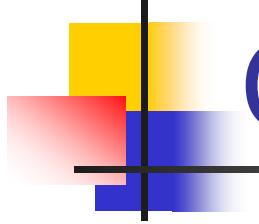


Clase 8



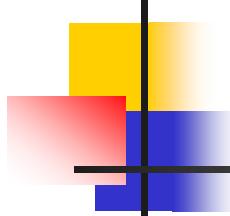
Temas de Clase

- Organización de Registros
- Instrucciones



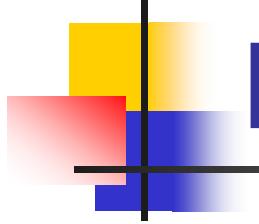
Organización de registros

- **Registros visibles al usuario:** son utilizados por el programador.
- **Registros de control y estado:** son utilizados por la UC para controlar la operación de la CPU (no son visibles por el programador).



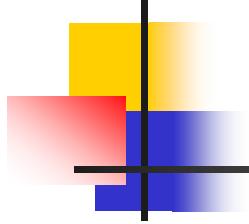
Registros visibles al usuario

- Propósito general
- Datos
- Dirección
- Códigos de condición



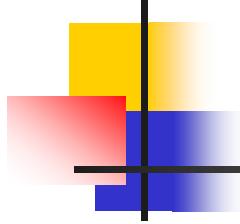
Registros visibles al usuario(2)

- Pueden ser asignados a una variedad de funciones:
 - ✓ cualquier registro de propósito general puede contener el operando para cualquier código de operación (verdadero propósito)
 - ✓ pueden existir restricciones (ej. registros dedicados a operaciones en PF)



Registros visibles al usuario(3)

- ✓ se pueden utilizar para direccionamiento (ej. indirecto de registro)
- ✓ sólo para datos ó sólo para direcciones
- ✓ los registros de dirección pueden ser asignados para un mdd (ej. reg índice para direccionamiento autoindexado)

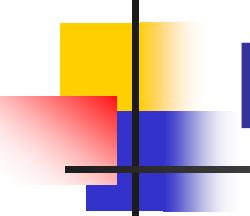


Discusión

❖ ¿Todos los registros de propósito general ó especializar su uso?

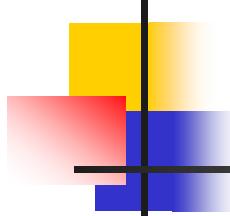
- ❖ Todos de propósito general: afecta al tamaño de las instrucciones.
- ❖ Especializados: puede estar implícito en el código de operación a qué registro se refiere (ej. Acumulador). Se ahorran bits. Limitan la flexibilidad del programador.

❖ No hay una receta.



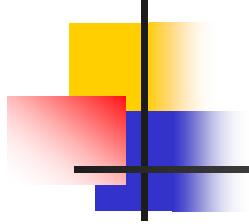
Número de registros

- ❖ Afecta al tamaño de la instrucción.
- ❖ Mayor Nº de registros, más bits para especificarlos en la instrucción.
- ❖ Pocos registros: más referencias a memoria
- ❖ Nº óptimo: entre 8 y 32 reg. Más, no hay gran mejora (aumenta tamaño de la instrucción).
- ❖ 2^{do} cuatrimestre: discutimos RISC.



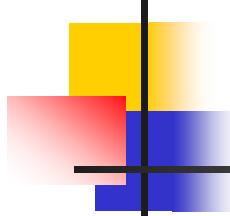
Longitud de los registros

- De direcciones: deben ser capaces de almacenar la dirección más grande.
- De datos: deben estar habilitados para almacenar la mayoría de los tipos de datos.
- Algunas máquinas permiten 2 registros contiguos utilizados como un solo registro para almacenar valores de doble longitud.



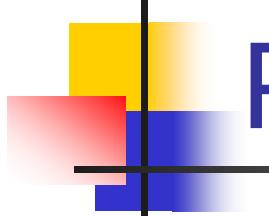
Bits de condición (banderas)

- Bits establecidos por la CPU como resultado de operaciones.
- Pueden ser utilizados por las instrucciones de bifurcación condicional.
- Generalmente no son alterados por el programador.



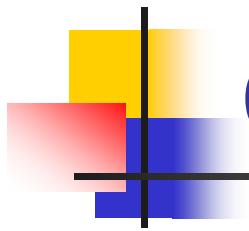
Registros de control y estado

- Empleados para controlar la operación de la CPU. En la mayoría de las máquinas no son visibles al usuario.
- Los 4 esenciales para la ejecución de instrucciones:
 - Contador de programa (PC)
 - Registro de instrucción (IR)
 - Registro de dirección de memoria (MAR)
 - Registro buffer de memoria (MBR)



Reg. de control y estado (2)

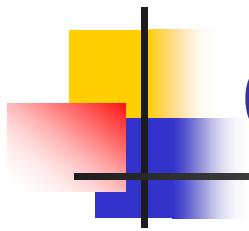
- Los 4 reg recién mencionados se emplean para el movimiento de datos entre la cpu y memoria.
- Dentro de la CPU los datos se deben presentar a la ALU para procesamiento, ésta puede acceder al MBR y a los reg visibles por el usuario. Puede haber también reg temporales adicionales para intercambiar datos con el MBR y demás reg visibles.



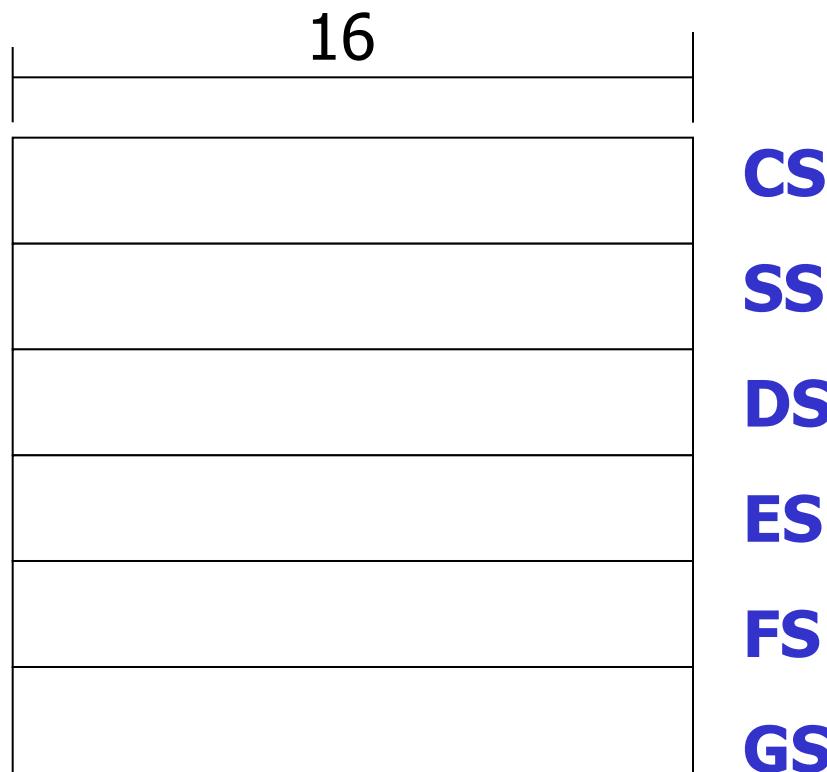
Organización de registros CPU PII Intel (principales)

16	8	8	
	AH A X	AL	EAX
	BH B X	BL	EBX
	CH C X	CL	ECX
	DH D X	DL	EDX

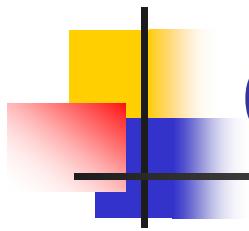
De uso general



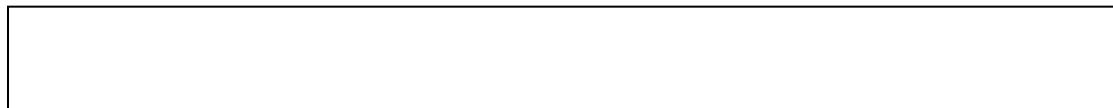
Organización de registros CPU PII Intel (principales)(2)



Segmentos



Organización de registros CPU PII Intel (principales)(3)

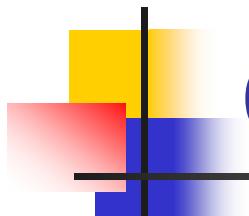


EIP



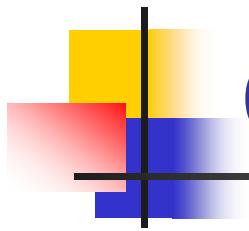
EFLAGS

PC y banderas



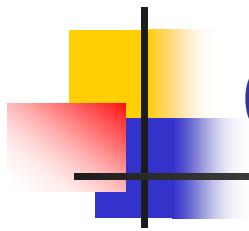
Organización de registros CPU PII Intel (principales)(4)

- **AX** : acumulador, es el principal en las operaciones aritméticas
- **BX** : puntero base (dir de memoria)
- **CX** : contador, interviene en instrucciones de ciclo
- **DX** : datos, participa en multiplicación y división

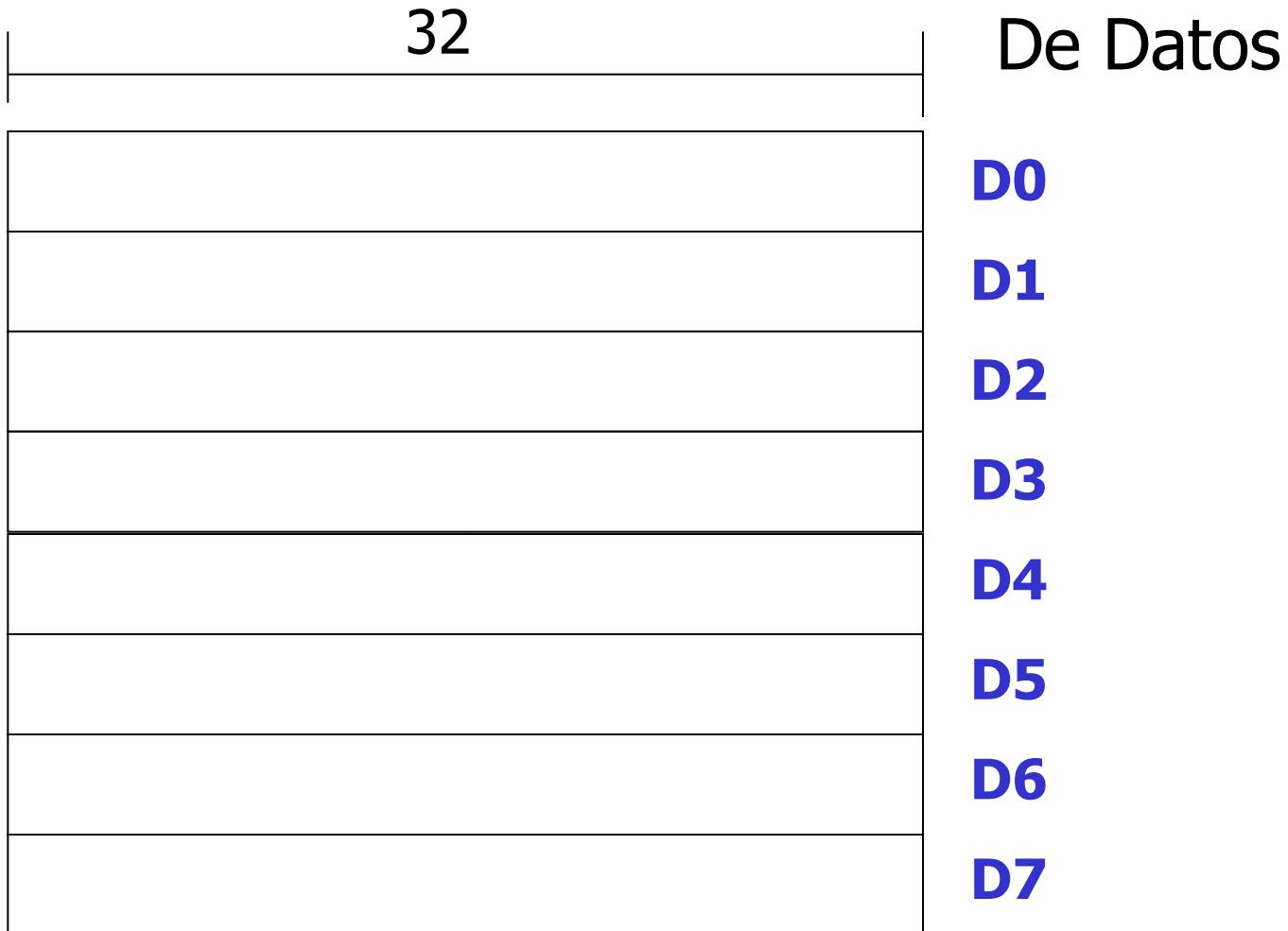


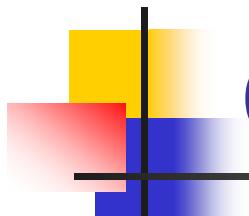
Organización de registros CPU PII Intel (principales)(5)

- **SI y DI** : apuntadores que utilizan las instrucciones que recorren arreglos o tablas
- **BP y SP** : también son apuntadores a memoria, pero a una zona especial: pila ó stack
- **E** : reg de 32 bits



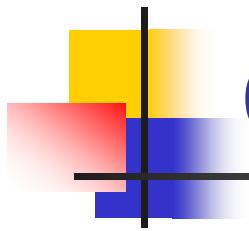
Organización de registros CPU MOTOROLA 68000



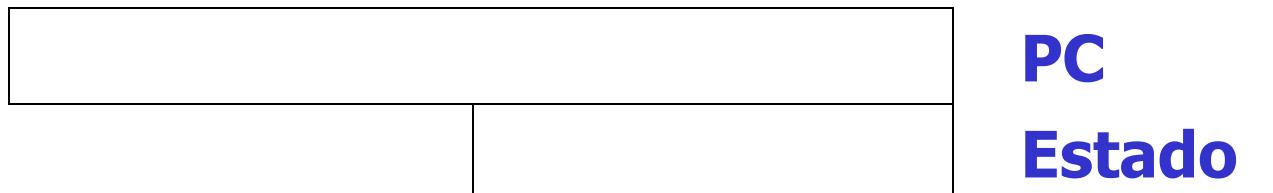


Organización de registros CPU MOTOROLA 68000 (2)

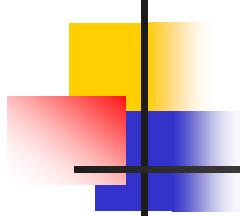
	De
A0	Direcciones
A1	
A2	
A3	
A4	
A5	
A6	
A7	
A7'	
Apuntador del stack usuario	
Apuntador del stack supervisor	



Organización de registros CPU MOTOROLA 68000 (3)



- 8 registros de 32 bits de datos
- 9 registros de direcciones
 - 2 stacks: uno para usuario y otro para S.O.

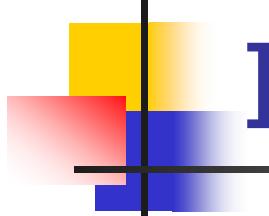


Instrucciones - Intel

➤ Tienen la forma :

instrucción destino,fuente

➤ destino y fuente son 2 operandos,
donde c/u de ellos está especificado por
alguno de los mdd vistos, el otro
operando es un registro de la CPU



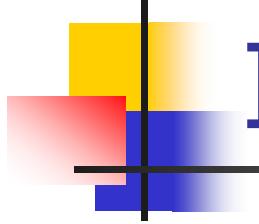
Instrucciones - Intel (2)

❖ Llamando :

- mem = especificación de una dirección de memoria
- reg = registro de la CPU
- imm = dato inmediato

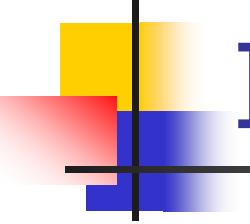


Las instrucciones
tienen la forma



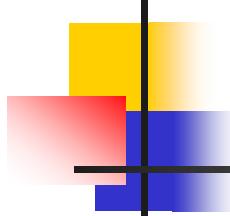
Instrucciones - Intel (3)

- Instrucción mem, reg
- Instrucción reg , mem
- Instrucción reg , reg
- Instrucción reg , inm
- Instrucción mem, inm



Instrucciones - Intel (4)

- El nombre destino y fuente proviene del hecho que si hay un movimiento de datos, es desde la derecha (fuente) hacia la izquierda (destino).
- En una suma hay 2 operandos y el resultado se almacena en el lugar del operando izquierdo (destino).

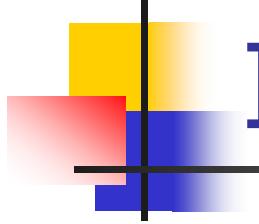


Instrucciones - Intel 8086

Ejemplos:

- ADD AX,BX → $AX=AX+BX$
- ADD AL,AH → $AL=AL+AH$
- MOV AL,CH → $AL=CH$
- SUB AX,BX → $AX=AX - BX$

❖ **Direccionamiento por registro**

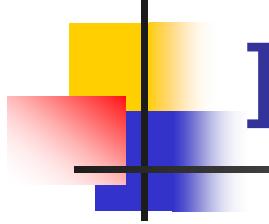


Instrucciones - Intel 8086 (2)

Ejemplos:

- ADD AX,35AFh → AX=AX+35AFh
- ADD AL,15 → AL=AL+15
- MOV AL,3Eh → AL=3Eh
- SUB AX,1234h → AX=AX - 1234h

❖ Direcciónamiento Inmediato



Instrucciones - Intel 8086 (3)

Ejemplos:

- ADD AX, [35AFh]

→ AX = AX + contenido direcc. 35AFh y 35B0h

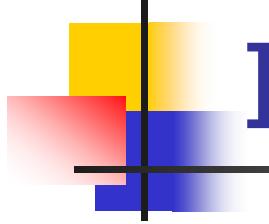
- ADD AL, DATO

→ AL = AL + contenido variable DATO (8 bits)

- MOV CH, NUM1

→ CH = contenido variable NUM1 (8 bits)

❖ **Direccionamiento Directo**



Instrucciones - Intel 8086 (4)

Ejemplos:

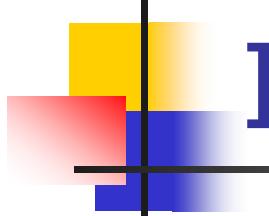
- ADD AX, [BX]

→ AX = AX + dato almacenado en dirección contenida en BX y la que sigue

- MOV [BX], AL

→ dato en la dirección contenida en BX = AL

❖ **Direccionamiento Indirecto por registro**



Instrucciones - Intel 8086 (5)

Ejemplos:

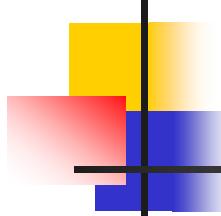
- `MOV CX, [BX+SI]`

→ CX = dato almacenado en la direcc. BX+SI
y la siguiente

- `MOV [BX+DI], AL`

→ dato almacenado en la direcc. BX+DI = AL

❖ **Direccionamiento base + índice**



Instrucciones - Intel 8086 (6)

Ejemplos:

- MOV AL, [BX+2]

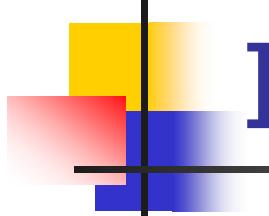
→ AL=dato almacenado en dir BX+2

- MOV [BX+2Ah], AX

→ dato almacenado en dir BX+2Ah y la que sigue = AX (16 bits)



Direccionamiento Relativo por registro



Instrucciones - Intel 8086 (7)

Ejemplos:

- MOV AL, [BX+SI+2]

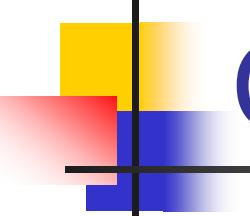
→ AL = dato almacenado en la dir BX+SI+2

- MOV [BX+DI+2Ah], AX

→ dato almacenado en la dir BX+DI+2Ah y la que sigue = AX (16 bits)

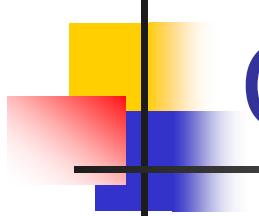


Direccionamiento relativo base+índice



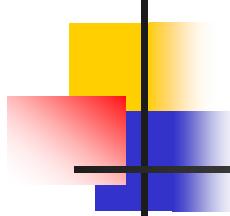
Formatos de instrucción- Criterios de diseño

- ✓ ¿Instrucciones cortas ó largas?
- ✓ N° de bits/seg
 - ✓ ancho de banda de la memoria
- ✓ Velocidad procesador/Velocidad memoria
- ✓ Instrucciones más cortas
 - ✓ el procesador “parece” más rápido.



Formatos de instrucción- Criterios de diseño (2)

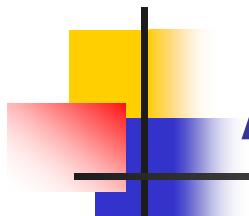
- ✓ Suficientes bits para expresar todas las operaciones deseadas.
- ✓ La experiencia demuestra dejar bits libres para el futuro.
- ✓ Cantidad de bits de datos.



Ejemplo para MSX88

- Editar prueba.asm
 - Usar Editor de textos
 - Ensamblar prueba.asm
 - Usar Asm88
 - Prueba.o y Prueba.lst
 - Enlazar prueba.o
 - Usar Link88
 - Prueba.eje
 - Usar MSX88
 - Cargar prueba.eje y ejecutar
- ```
ORG 2000H
MOV BX,3000H
MOV AX,[BX]
ADD BX, 02H
MOV CX,[BX]
ADD AX,CX
PUSH AX
POP DX
HLT

org 3000h
db 55h, 33h, 44h, 22h
END
```



# Archivo .lst

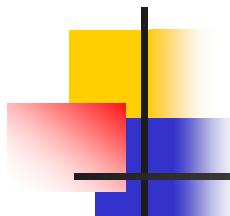
| Dir.             | Código máquina | Línea | Código en lenguaje ensamblador |
|------------------|----------------|-------|--------------------------------|
|                  |                | 1     | ORG 2000H                      |
| 2000 BB 00 30    |                | 2     | MOV BX,3000H                   |
| 2003 8B 07       |                | 3     | MOV AX,[BX]                    |
| 2005 81 C3 02 00 |                | 4     | ADD BX, 02H                    |
| 2009 8B 0F       |                | 5     | MOV CX,[BX]                    |
| 200B 03 C1       |                | 6     | ADD AX,CX                      |
| 200D 50          |                | 7     | PUSH AX                        |
| 200E 5A          |                | 8     | POP DX                         |
| 200F F4          |                | 9     | HLT                            |
|                  |                | 10    |                                |
|                  |                | 11    | org 3000h                      |
| 3000 55 33 44 22 |                | 12    | db 55h, 33h, 44h, 22h          |
|                  |                | 13    | END                            |

S I M B O L O S:

Nombre:

Tipo:

Valor:



# mas información ...

- Organización de los registros
  - Capítulo 11 apartado 11.2. Stallings, W., 5º Ed.
- Formatos de instrucciones
  - Capítulo 10 apartado 10.3.y 10.4 Stallings, W., 5º Ed.
- Links de interés
  - [http://www.intel.com/museum/online/hist\\_micro/hof/index.htm](http://www.intel.com/museum/online/hist_micro/hof/index.htm)
- Simulador MSX88
  - En Descargas de página web de cátedra