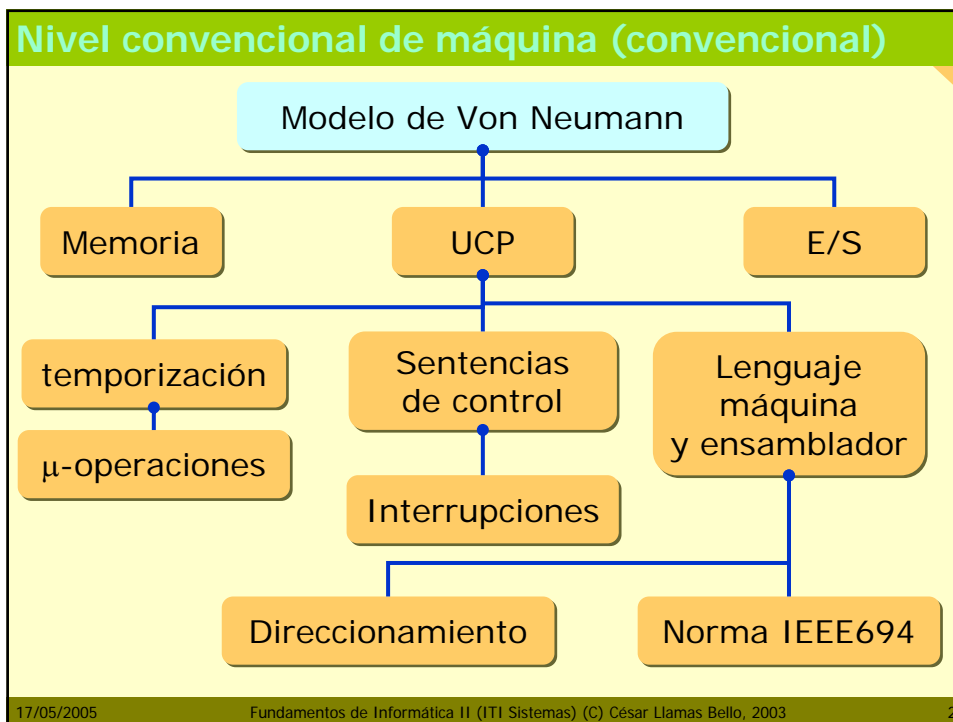


### 3. Nivel de máquina convencional

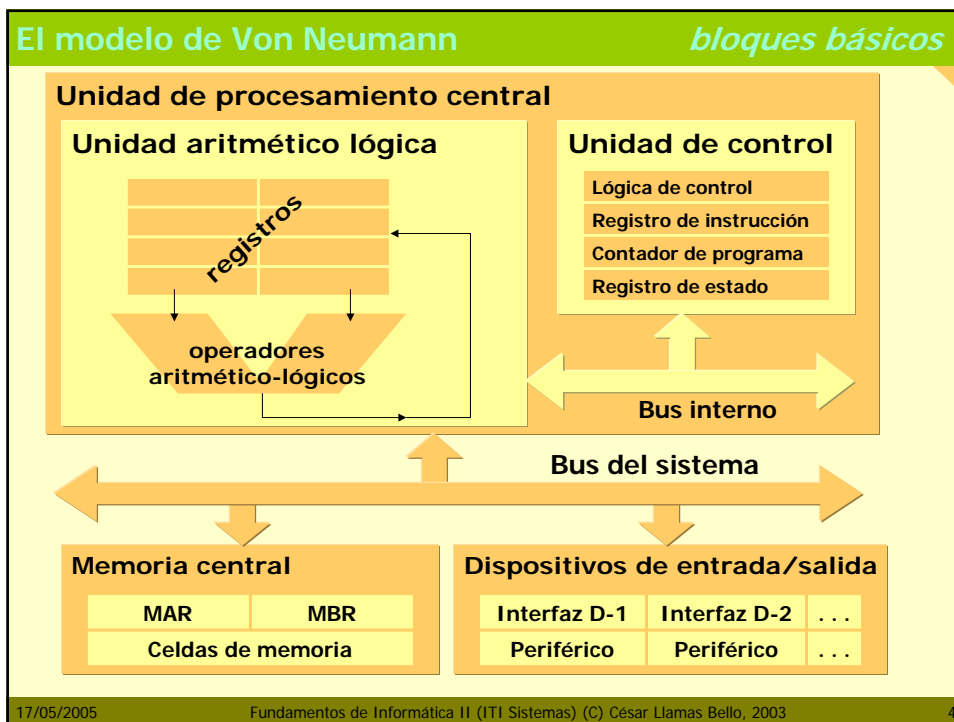
- ❑ El Modelo de von Neumann
- ❑ La Unidad Central de Procesamiento
- ❑ Lenguaje Máquina y Lenguaje Ensamblador

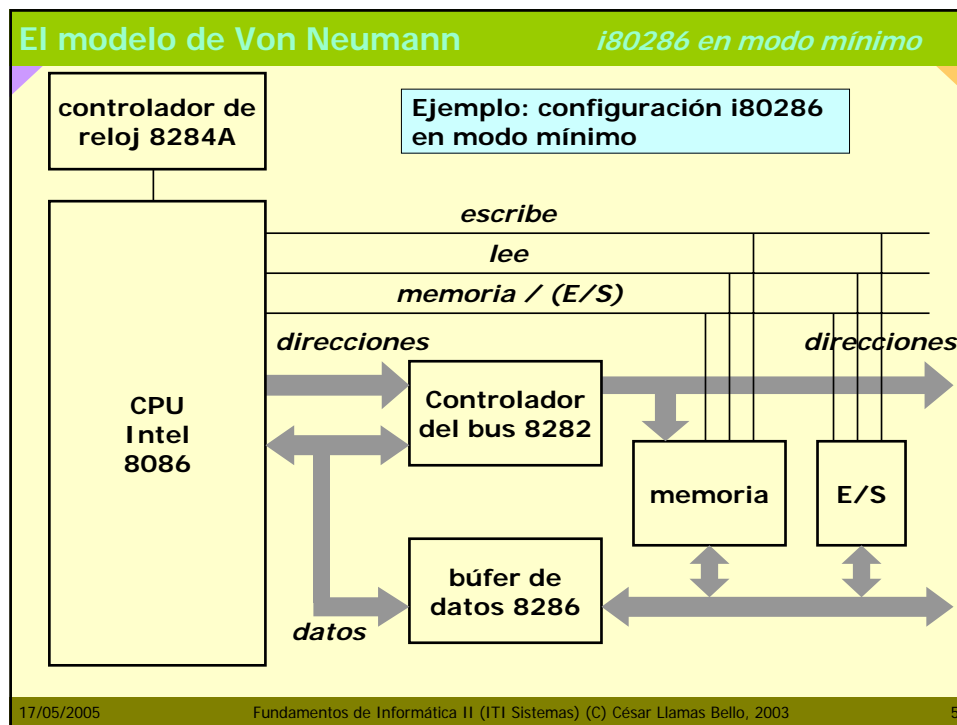


**El modelo de Von Neumann**

- ❑ Las máquinas habituales contienen:
  - Una unidad de procesamiento de datos. Que obedece a un programa almacenado en la unidad de almacenamiento.
  - Una unidad de almacenamiento de datos. Que contiene los datos iniciales, los datos de trabajo, los datos finales y el programa.
  - Unidades de entrada y salida.
  - Articuladas mediante uno o varios buses
- ❑ Idea original: Permitir programas automodificables.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 3





- Unidades funcionales básicas** *memoria*
- ❑ Memoria central (o principal); almacena
    - Las instrucciones del proceso;
    - los datos iniciales, intermedios y finales; y
    - los datos adicionales para el proceso (listas, tablas, ...)
  - ❑ Se organiza en forma de tabla, donde cada elemento tiene la misma capacidad, y cada célula se identifica con un número entero:
    - El tamaño de la palabra de memoria es de 1, 2 o 4 bytes
    - El identificador es la dirección o posición de memoria con posiciones correlativas (aunque puede haber huecos).
- 17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 6

**Unidades funcionales básicas** *memoria*

- ❑ Características importantes
  - Tiempo de acceso (x ns)
  - Organización de la memoria
    - Número de bytes por célula
  - Tamaño de la memoria
    - Se mide en bytes, independientemente de la organización.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 7

**Unidades funcionales básicas** *memoria*

- ❑ Otros elementos:
  - MAR (*memory address register*): registro donde se inserta la posición de la memoria a leer o escribir.
  - MBR (*memory buffer register*): registro donde se aloja el resultado de leer o el dato a insertar
  - Este es un esquema simple con acceso único.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 8

**Unidades funcionales básicas** *UCP*

- ❑ Los diseñadores de la UCP especifican el conjunto de instrucciones a que obedece
- ❑ Opera como un intérprete sobre los programas escritos en el código de la máquina:  
Tareas repetitivas (*read-eval*):
  1. Interna: extrae la instrucción, la decodifica y se predispone a leer la siguiente
  2. Externa: ejecuta la instrucción actual (puede influir en la siguiente tarea)

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 9

**Unidades funcionales básicas** *UCP*

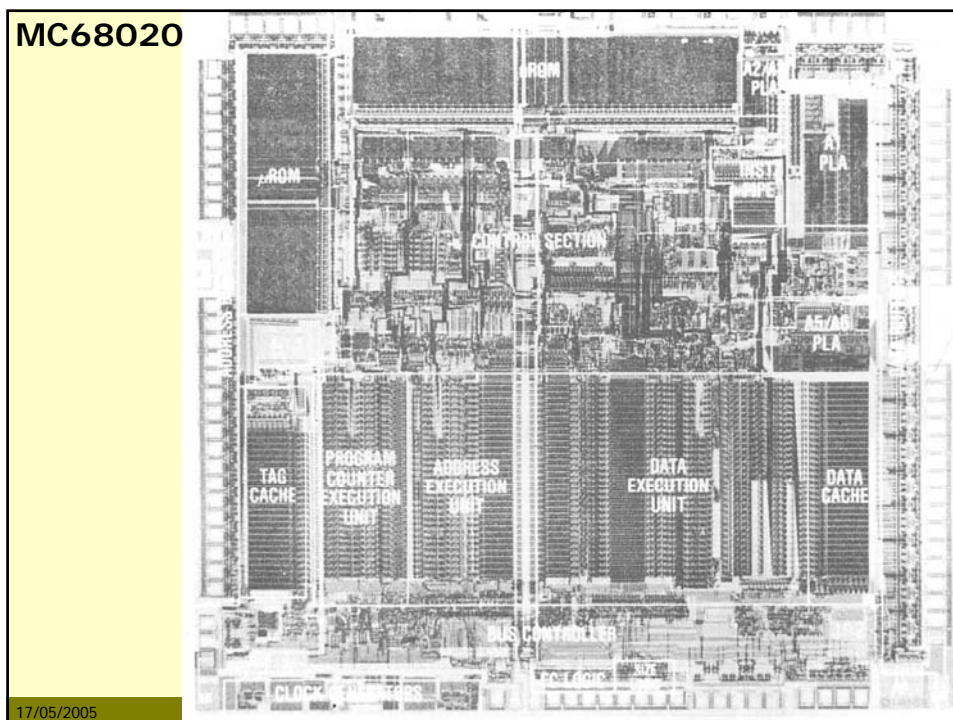
- ❑ Bloques:
  - Unidad aritmético-lógica (UAL) efectúa operaciones sobre los datos de los registros internos.
  - El tamaño de estos registros define el tamaño de la palabra de UCP
  - Puede contener registros especializados
  - Puede haber registros para trabajar con direcciones de memoria

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 10

**Unidades funcionales básicas** **UCP**

- ❑ + bloques:
  - Unidad de Control  
Realiza el ciclo de ejecución secuencial de la UCP, controlando el resto de componentes.
  - Posee varios registros especializados:
    - PC (*Program Counter*) Índice a la posición de memoria de la instrucción actual
    - IR (*Instruction Register*) Contiene la instrucción en curso
    - SW (*Status Word*) Contiene diversos bits que plasman el estado de la ejecución.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 11



**Unidades funcionales básicas** *UCP*

- ❑ Una UCP y una memoria (SISD)
- ❑ Una unidad de control y varias UAL
  - Una instrucción afecta a varios datos (SIMD)  
Máquina para procesar vectores de datos  
"procesador vectorial"
- ❑ Varias UCP (MIMD)
  - Una sola memoria:  
"Multiprocesador de memoria compartida"
  - Memorias separadas  
"Máquina multiprocesador de memoria independiente"

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 13

**Unidades funcionales básicas** *UCP*

- ❑ Una máquina puede estar formada por varios procesadores  
*¿dónde está el nivel de máquina?*
- ❑ Respuesta: El nivel de máquina tiene un nivel inferior que es el *nivel de procesador*.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 14

**Unidades funcionales básicas** *Entrada/Salida*

- ❑ Interactúan con fuentes de datos externas a los bloques anteriores
  - Desde el exterior los datos se presentan en algún código para la representación externa
    - Ej: Código de barras, código de impresora, caracteres alfanuméricos, ...
  - Los dispositivos de E/S pueden hacer una conversión a un código para la representación interna

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 15

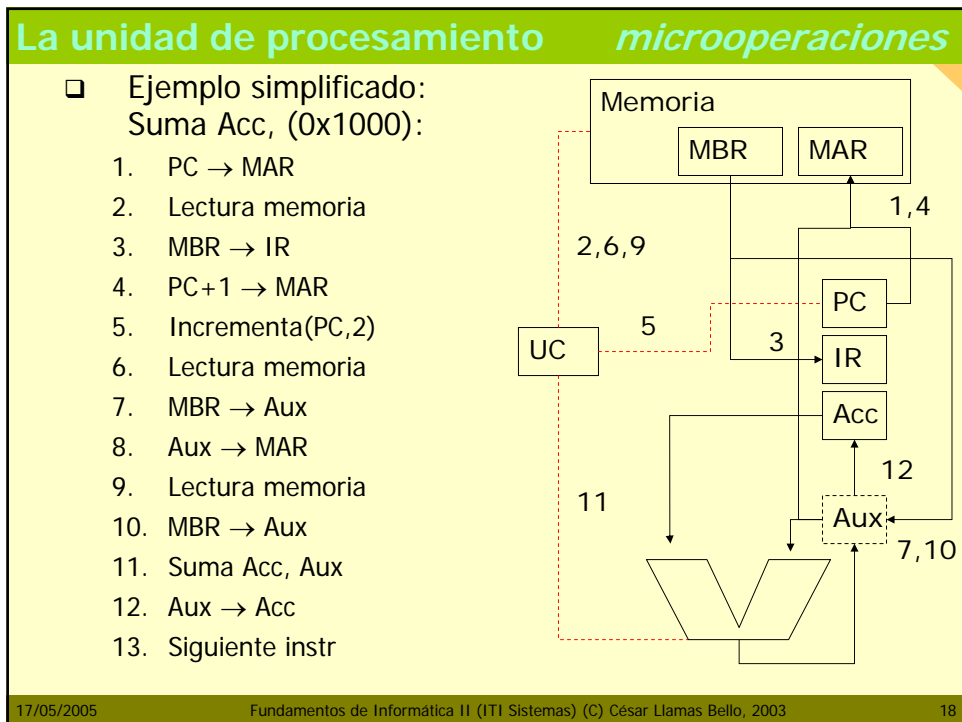
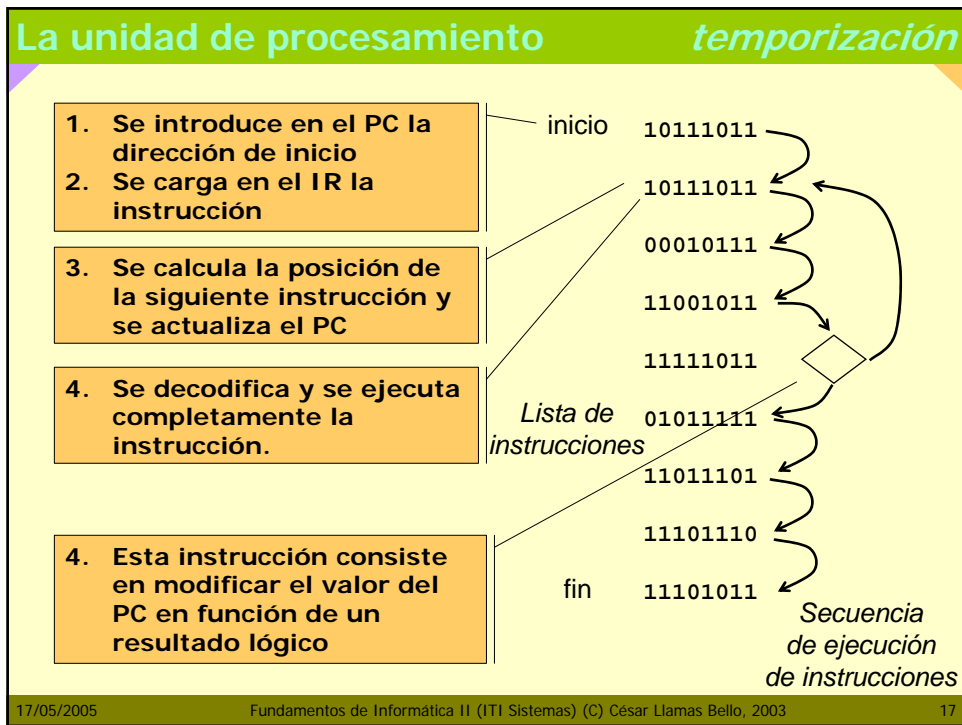
**La unidad de procesamiento** *temporización*

El diagrama muestra un flujo de control para la ejecución de una instrucción. Comienza con un nodo 'inicio' (ovalado) que apunta a un punto de conexión (círculo amarillo). Desde allí, una flecha apunta a un rectángulo 'Leer la instrucción actual' etiquetado como 'Ciclo de lectura'. Una flecha descendente desde este rectángulo apunta a otro rectángulo 'Ejecutar la instrucción' etiquetado como 'Ciclo de ejecución'. Una flecha horizontal desde el rectángulo 'Ejecutar la instrucción' apunta a un nodo 'fin' (ovalado). Una flecha también apunta desde el punto de conexión de inicio hacia el rectángulo 'Ejecutar la instrucción', indicando un salto o un ciclo de retroalimentación.

- ❑ Este es el funcionamiento básico: síncrono con respecto al programa.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 16





### La unidad de procesamiento *microoperaciones*

**Unidad de control de lógica cableada**

**Unidad de control  $\mu$  - programada**

$\mu$ -controlador

Memoria  $\mu$ -programa

$\mu$ -PC

$\mu$ -IR

- ❑ Unidad de control de lógica cableada (*random*)
  - Diseñada mediante secuenciadores para cada instrucción
  - Usual en máquinas RISC
- ❑ Unidad de control microprogramada (M. de Wilkes)
  - Diseñada en torno a un microcontrolador con memoria PC e IR.
  - Usual en máquinas CISC
  - Originalmente se podía actualizar el microprograma

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 19

### La unidad de procesamiento *sentencias de control*

- ❑ Saltos:
  - Condicionales e incondicionales
  - Absolutos en la memoria o relativos a la posición del programa.
    - Los saltos relativos a la posición del programa permiten reubicar el programa sin tener que alterar el programa
    - Los saltos absolutos no son reposicionables en otra ubicación de la memoria diferente en la que fueron *creados* (compilados).

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 20

**La unidad de procesamiento** *sentencias de control*

- ❑ Salto a subrutinas
  - Se usa una estructura llamada PILA de RETORNO
  - Incondicional o condicional
  - Una subrutina tiene este esquema:
    1. Punto de entrada a la subrutina
    2. Salvaguardia de registros
    3. Recogida de valores desde la pila
    4. (código de la subrutina)
    5. Paso de valores del resultado a la pila
    6. Restauración de registros
    7. Retorno de la subrutina

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 21

**La unidad de procesamiento** *sentencias de control*

- ❑ Al hacer la llamada a la subrutina se usa una pila de retorno para almacenar el valor del PC
- ❑ Al volver de una rutina se rescata el valor de la pila de retorno al PC
- ❑ Este mecanismo permite anidar llamadas a subrutinas

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 22

**La unidad de procesamiento** *sentencias de control*

- ❑ Interrupciones:
  - Procesamiento asíncrono a la secuencia del programa
- ❑ Origen:
  - Externo al programa:
    - Hardware: Alimentación, petición de E/S...
    - Software: Despachador del S.O., otro proceso del sistema...
  - Interno al programa.
    - Por petición propia (trap).
    - Por fallo en alguna instrucción: overflow, violación de memoria, ...

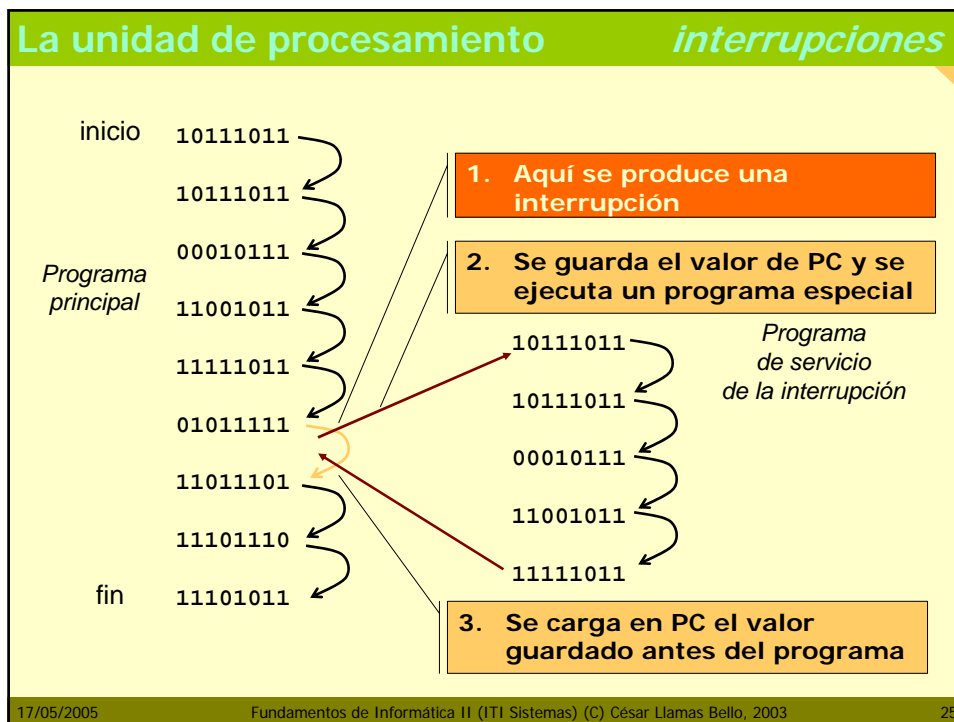
17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 23

**La unidad de procesamiento** *interrupciones*

```
graph TD; inicio([inicio]) --> LIA[Leer la instrucción actual]; LIA --> EII[Ejecutar la instrucción]; EII --> fin([fin]); EII -- "Interrupciones habilitadas" --> TI[Test interrupción: interrupción]; TI --> EII; EII -- "Interrupciones inhabilitadas" --> LIA; subgraph Ciclo_de_ejecucion [Ciclo de ejecución]; LIA --> EII --> fin; end; subgraph Ciclo_de_interrupcion [Ciclo de interrupción]; EII --> TI --> EII; end;
```

- ❑ Ejecución asíncrona: la ejecución síncrona se ve alterada por fenómenos exteriores al programa.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 24



**La unidad de procesamiento** *interrupciones*

❑ En los procesadores evolucionados se distingue entre:

- Procesamiento normal, o de usuario.  
En él se evalúan los programas habituales.
- Procesamiento excepcional, o de supervisor, o privilegiado.  
En él se evalúan las rutinas de servicio, y disfrutan de acceso a **instrucciones privilegiadas** (protegidas).
- Ciertas rutinas de servicio del sistema operativo precisan instrucciones privilegiadas y se ejecutan en modo supervisor.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 26

## Lenguaje máquina y lenguaje ensamblador

- ❑ El lenguaje máquina de un procesador se especifica en un catálogo con todas las instrucciones admisibles por la máquina, y su semántica.
- ❑ El lenguaje ensamblador es una visión de alto nivel del lenguaje máquina y es más tratable por un humano entrenado.

17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

27

## L. Máquina y L. Ensamblador

*ejemplo*

### Ejemplo ensamblador Motorola 68000

```
MOVE.W D4, -(SP) ; D4 -> a la pila
MULS D2, D4      ; D2 x D4 -> D4
BMI DEUDA        ; salta a DEUDA si <0
BEQ NOPAGA       ; salta a NOPAGA si =0
MOVE.L D4, A$2000 ; D4 -> Memoria($00002000)
MOVE.W (SP)+, D4 ; restaura D4 desde la pila
...
BRA FINAL        ; salto incondicional
DEUDA MOVE.W (SP)+, D4 ; restaura D4 desde la pila
...
BRA FINAL        ; salto incondicional
NOPAGA MOVE.W (SP)+, D4 ; restaura D4 desde la pila
...
FINAL
```

17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

28

L. Máquina y L. Ensamblador		<i>ejemplo</i>
MOVE.W D4, -(SP)	<b>0011</b> (rrr=A7=111) <u>100 100</u> (rrr=D4=010)	
	=0011 1000 0010 0111	= x3827
MULS D2, D4	<b>0100110000</b> <u>000</u> (rrr=D2=010)	
	=0100 1100 0000 0010	= x4C02
	<b>0</b> (Ddr=D4=100)(S=1)(Q=1) <u>00000000</u> (Hdr=010)	
	=0100 1100 0000 0010	= x4C02
BMI DEUDA (40D)	<b>0110</b> (MI=1101) (Dis=00100010)	
	=0110 1101 0010 0010	= x6D22
BEQ NOPAGA (64D)	<b>0110</b> (EQ=0111) (Dis=01000000)	
	=0110 0111 0100 0000	= x6740
MOVE.L D4, \$A2000	=0010 (abs.L=001 111) 100 (rrr=D4=010) ...	
	0000 0000 0000 0000...	
	0010 0000 0000 0000.....	

**xxx** = opcode  
**yy** = info adicional

Resultado:  
... 38 27 4C 02 4C 02 6D 22 67 40 02 03 E2 00 ...

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 29

L. Máquina y L. Ensamblador		<i>aspecto</i>
<p>❑ El lenguaje ensamblador tiene tres tipos de primitivas:</p> <ul style="list-style-type: none"> <li>• Mnemónicos (nemónicos): Cada mnemónico alude a un juego de instrucciones similares del lenguaje máquina.</li> <li>• Pseudoinstrucciones: Con instrucciones auxiliares para (por ejemplo): <ul style="list-style-type: none"> <li>▪ reserva de memoria, carga de datos en memoria,</li> <li>▪ indicar posiciones de memoria,</li> <li>▪ macroinstrucciones, ...</li> </ul> </li> <li>• Pragmas: para dar indicaciones al compilador sobre la compilación del programa (en tiempo de ejecución <i>del compilador</i>).</li> </ul>		

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 30

**L. Máquina y L. Ensamblador** *ejemplo*

DEUDA	... BRA MOVE.W	FINAL (SP)+, D4 ...	; salto incondicional ; restaura D4 desde la pila
-------	----------------------	---------------------------	--

Zona de comentarios

Zona de operandos

Zona de instrucciones

Zona de etiquetas de direcciones de memoria simbólicas

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 31

**L. Máquina y L. Ensamblador** *direccionamiento*

- ❑ Cada Mnemónico se refiere a un conjunto de instrucciones que realizan una acción pero que se diferencian en la forma de referirse a los datos.
- ❑ Modo de direccionamiento:  
Modo de referirse a los operandos

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 32



**L. Máquina y L. Ensamblador** *direccionamiento*

- ❑ Implícito: dato incluido en la instrucción
  - Return
- ❑ Inmediato o literal:
  - Move reg, #0
- ❑ Directo por registro
  - Add acum, #1

The diagram illustrates three addressing modes:

- Implicit:** An orange box labeled 'instrucción' contains a blue box labeled 'dato'.
- Inmediato o literal:** An orange box labeled 'instrucción' contains a blue box labeled 'dato'.
- Directo por registro:** An orange box labeled 'instrucción' contains a white box labeled 'nombre registro'. An orange arrow points from 'nombre registro' to a white box labeled 'registro', which contains a blue box labeled 'dato'.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 33

**L. Máquina y L. Ensamblador** *direccionamiento*

- ❑ Direccionamiento directo (absoluto):
  - Jump x0300
- ❑ Indirecto:
  - A memoria Move D, (x0200)
  - Por registro Move (D), #45

The diagram illustrates indirect addressing modes:

- Direccionamiento directo (absoluto):** An orange box labeled 'instrucción' contains a white box labeled 'dirección memoria'. An orange arrow points from 'dirección memoria' to a blue box labeled 'dato' in a vertical 'memoria' structure.
- Indirecto:**
  - A memoria:** An orange box labeled 'instrucción' contains a white box labeled 'dirección memoria'. An orange arrow points from 'dirección memoria' to a white box labeled 'dirección' in a 'memoria' structure. Another orange arrow points from 'dirección' to a blue box labeled 'dato' in the 'memoria' structure.
  - Por registro:** An orange box labeled 'instrucción' contains a white box labeled 'nombre registro'. An orange arrow points from 'nombre registro' to a white box labeled 'dirección' in a 'registro' structure. Another orange arrow points from 'dirección' to a blue box labeled 'dato' in a 'memoria' structure.

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 34

**L. Máquina y L. Ensamblador** *direccionamiento*

❑ Relativo

- `Jmpr 50`

❑ Por base y desplazamiento

- `Move D1, 5(A2)`

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 35

**L. Máquina y L. Ensamblador** *direccionamiento*

❑ Indexado por registro

- `Cmp D2, A2(A5)`

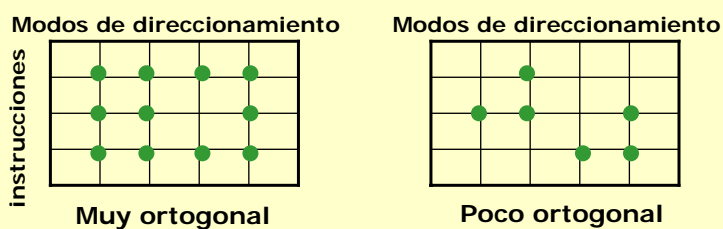
❑ Indirecto automodificado

- `Clear ++(A1)`

17/05/2005 Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003 36

### (Pistas) Ortogonalidad

- ❑ No todos los modos de direccionamiento están disponibles para todas las instrucciones
  - Grado de ortogonalidad: medida en que los modos de direccionamiento están disponibles para el conjunto de instrucciones.



17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

37

### Utilidad de modos de direccionamiento

- ❑ Indirecto -> trabajo con variables
  - Permite separar el espacio de direcciones del programa del de los datos.
- ❑ Relativo
  - Permite reubicar programas.
  - Es menos costoso en uso de la memoria.
- ❑ Por base y desplazamiento -> estructuras
  - Permite referirse a elementos de estructuras (*record*)
- ❑ Indexado -> como punteros a estructuras y vectores
- ❑ Auto-modificado -> en bucles.

17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

38

## Mnemónicos de la norma IEEE 694

### INSTRUCCIONES DE TRANSFERENCIA DE DATOS

LD	Cargar desde la memoria
ST	Memorizar
MOVE	Copiar de fuente a destino
FILL	Memorizar en varias posiciones
PUSH	Apilar el operando
POP	Desapilar el operando
XCH	Intercambiar contenidos
IN	Lectura desde puerto a registro o memoria
OUT	Escritura a puerto desde registro o memoria

17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

39

## Mnemónicos de la norma IEEE 694

### INSTRUCCIONES ARITMÉTICAS

ADD-	Suma -U, -S, -C, -R
SUB-	Resta -U, -S, -C, -R
MUL-	Multiplicación -U, -S
DIV-	División -U, -S
TEST	Actualizar SW según el operando
TESTSET	Lee y actualiza una posición de memoria
NEG	Cambia el signo del operando
ADJ	Ajusta a BCD
CVT	Convierte el tipo o formato del operando

17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

40

## Mnemónicos de la norma IEEE 694

### INSTRUCCIONES LÓGICAS Y DE DESPLAZAMIENTO

AND, OR	Producto, suma lógica
XOR	OR exclusivo
NOT	Complemento lógico --C, --V
SET, CLR	Establecer a 1, a 0, los operandos --C, --V
SHL, SHR	Desplazamiento de bit a izquierda, derecha
SHLA	Desplazamiento aritmético a izquierda
SHRA	Desplazamiento aritmético a derecha
ROL, ROR	Rotación a izquierda, derecha, sin acarreo
ROLC, RORC	Rotación a izquierda, derecha, con acarreo

17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

41

## Mnemónicos de la norma IEEE 694

### INSTRUCCIONES DE CONTROL DE FLUJO DE EJECUCIÓN

BR	Salto incondicional
-B-	Salto condicional (*)
CALL-	Invocación a procedimiento (*)
RET-	Retorno de procedimiento (*)
RETI	Retorno de servicio de interrupción
RETSKIP	Retorno de procedimiento incrementado
SKIP-	Salto hacia delante (*)

(\*) Prefijos: D-, I-

Sufijos: -Z, -NZ, -E, -NE, -GT, -GE, -LT, -LE, -H, -NH, -L,  
-NL, -C, -P, -N, -V, -PE, -NC, -NP, -N, -NV, -PO, -T, -F.

17/05/2005

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

42

## Mnemónicos de la norma IEEE 694

### INSTRUCCIONES DE CONTROL Y MISCELÁNEA

BRK	Iniciar una interrupción
CC-	El operador destino toma el valor de la condición
EI, DI	Habilita o inhabilita interrupciones
ENTER	Suministro de protocolos de entrada (ENTER)
EXIT	y salida (EXIT) de procedimientos
HALT	Parar la UCP
INS	Inserción de campo de bits
NOP	Avanzar a la instrucción siguiente
TR	Remplazar operador-dirección por contenido
WAIT	Parar hasta interrupción