

UNIDADES FUNCIONALES DEL COMPUTADOR

Tema 5

Principios de Computadoras

Temas

Introducción

1. Bus
 2. Memoria
 3. Entrada/Salida
 4. Unidad aritmética-lógica
 5. Unidad de Control
-

3. ENTRADA/SALIDA

Tema 5

Principios de Computadoras

3. Entrada/Salida

- Introducción
 - Entrada/salida programada
 - Entrada/salida manejada por interrupciones
 - Entrada/salida de acceso directo a memoria (DMA)
 - Entrada/salida usando canales de datos
-

Introducción

- Los dispositivos E/S permiten el intercambio de información entre la CPU y la memoria y el mundo exterior.
 - Los sistemas E/S incluyen: el dispositivo E/S (periférico), la unidad de control de este dispositivo, el software diseñado para realizar las operaciones E/S.
 - Las instrucciones de E/S son las que más varían entre una máquina y otra.
-

Introducción

- Los sistemas E/S se diferencian por la necesidad de utilizar la CPU en la ejecución de las operaciones E/S:
 - 1) Si la CPU ejecuta el programa para iniciar, dirigir y terminar la operación E/S: **E/S PROGRAMADA**.
 - 2) Si el dispositivo E/S puede pedir servicio a la CPU y liberar a la CPU de comprobar periódicamente el estado de los periféricos: **INTERRUPCIÓN**.
 - 3) Si la CPU es responsable sólo de iniciar la operación E/S y el dispositivo E/S es capaz de realizar transferencias con la memoria sin intervención de la CPU: **ACCESO DIRECTO A MEMORIA (DMA)**.
-

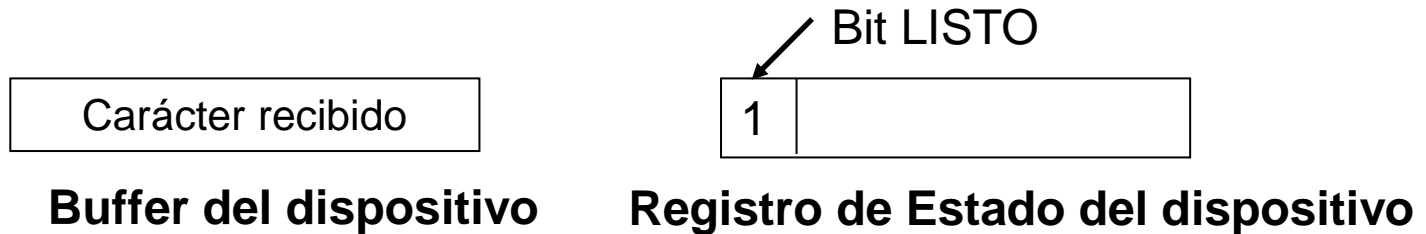
1. E/S programada

- “*Transferencia programada de datos*” o “*control directo por programa*”. Es el método más sencillo pero también el más ineficiente.
- Tiene una sola instrucción de entrada y una sola de salida. Cada una de estas instrucciones selecciona uno de los dispositivos de E/S y se transfiere un solo carácter entre un registro fijo del procesador (acumulador) y un registro buffer (registro de datos) conectado al dispositivo E/S seleccionado. El dispositivo no tiene acceso directo a la memoria principal.
- El procesador debe ejecutar una instrucción explícita por cada carácter que se lee o se escribe. Para transferir un conjunto de datos será preciso ejecutar un número de instrucciones igual o múltiplo al del número de datos.

1. E/S programada

Ejemplo: teclado

- Se introduce un carácter por teclado, se guarda en el buffer del teclado (registro) y se pone el bit LISTO a 1.



- Cuando se usa E/S programada el procesador entra en una iteración, leyendo constantemente el registro de estado del teclado, esperando a que se ponga a 1 LISTO. Cuando esto sucede, lee el registro del buffer para obtener el carácter y se pone LISTO a 0.

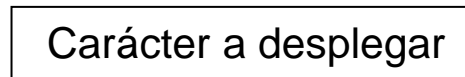
1. E/S programada

Ejemplo: pantalla

- Se quiere sacar un carácter por pantalla.



Registro de Estado del dispositivo



Buffer del dispositivo

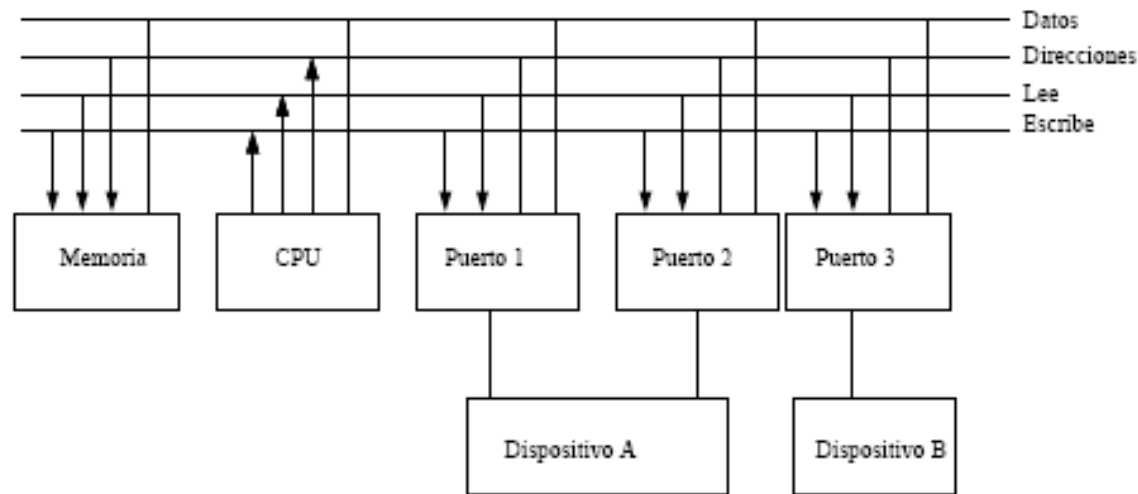
- El procesador lee el estado de despliegue para ver si LISTO está a 1. Si no es así realiza una iteración hasta que lo esté. Escribe el carácter a sacar por pantalla en el buffer de despliegue, se transfiere a pantalla y se pone el bit LISTO a 0. Cuando el terminal ha terminado vuelve a poner LISTO a 1.

1. E/S programada – Direccionamiento E/S

- Los dispositivos E/S, la memoria principal y la CPU se comunican normalmente por un solo bus (bus del sistema).
- Las líneas de direcciones del bus se usan para las direcciones de la memoria y para seleccionar dispositivos de E/S. Cada unión entre bus y dispositivo de E/S se llama **PUERTO DE E/S** y se le asigna una **dirección única**. El puerto E/S incluye un registro buffer de datos.

1. E/S programada – Direccionamiento E/S

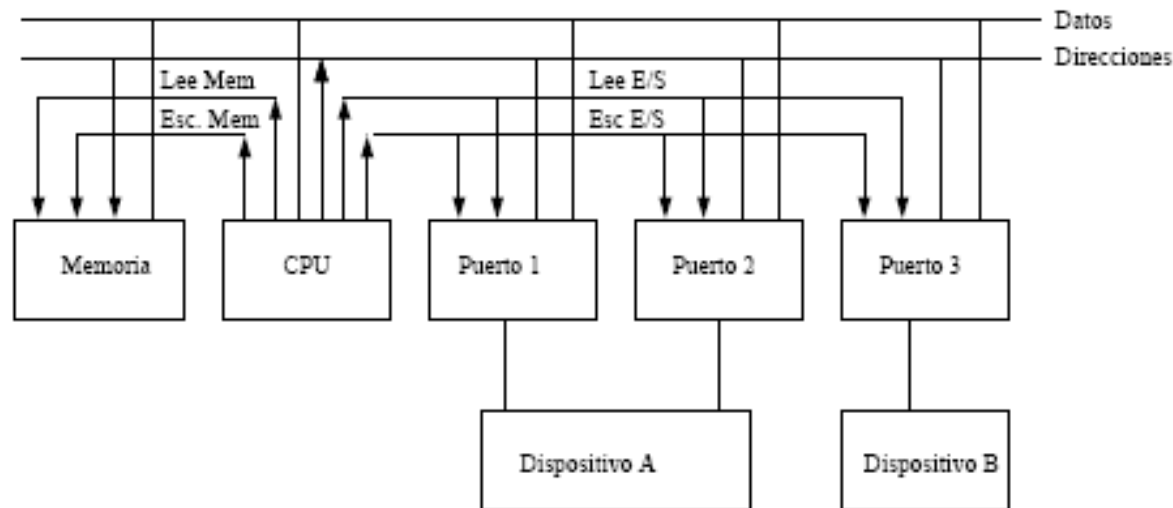
- Una estrategia es asignar parte de las direcciones de memoria principal a los puertos E/S: **E/S MAPEADA EN MEMORIA**.
- Una instrucción de referencia a memoria se convierte en una instrucción de E/S si la dirección corresponde a un puerto E/S. No se necesitan instrucciones especiales para E/S (INSTRUCCIONES E/S IMPLÍCITAS).



Las líneas "Lee" y "Escribe" se usan para iniciar un ciclo de memoria y para hacer transferencias E/S. Ej: Motorola 68000.

1. E/S programada – Direccionamiento E/S

- Otra estrategia es mantener separados los espacios de direcciones E/S y de memoria. Una instrucción de referencia a memoria activa las líneas READ M o WRITE M. Una instrucción de E/S activa las líneas READ IO o WRITE IO.
- Un dispositivo E/S y una posición de memoria pueden tener la misma dirección. Ej: 8085, 8086.



1. E/S programada – Direccionamiento E/S

- Si el espacio de direcciones de memoria y el de los dispositivos E/S está separado, las instrucciones E/S son INSTRUCCIONES E/S EXPLÍCITAS.
- Puede que el bus de E/S esté separado del de memoria o que ambos compartan el mismo bus.
- Instrucciones E/S explícitas imprescindibles:
 - Entrada: IN X
Una palabra se transmite desde la puerta E/S X al acumulador
 - Salida: OUT X
Una palabra se transmite desde el acumulador a la puerta E/S X

1. E/S programada – Circuitos de Interface

- Conexión de dispositivo E/S a un ordenador: con circuitos estándar que se llaman **INTERFACES E/S**. Estos circuitos permiten conectar un gran número de dispositivos de diferentes características a un bus del sistema común evitando usar hardware de propósito especial.

1) **Buffer**: circuito más simple. Es un registro de una palabra que actúa como puerto E/S. Tiene una dirección única y se accede a él como si fuera una dirección en memoria. Útil en la comunicación en paralelo.

2) **UART**: permiten conexión del ordenador con un dispositivo E/S que emplee comunicación serie (ej: modem). Es un registro de desplazamiento programable que transforma cadenas de datos en serie en datos en paralelo y viceversa.

3) **PPI (Programmable Peripheral Interface)**: pueden ser modificados por el programa de control para adaptarse a las características de distintos dispositivos E/S.

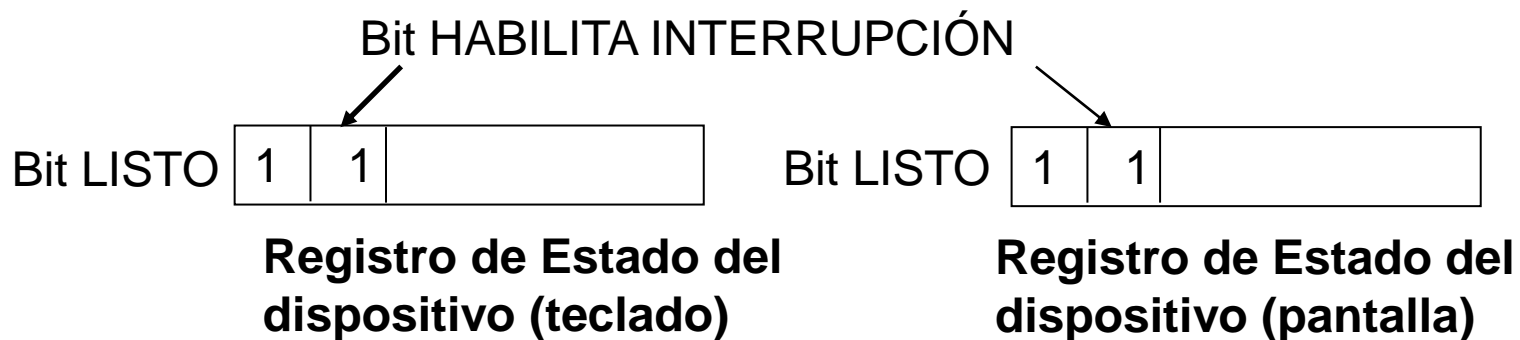
1. E/S programada - Problemas

Desventaja: esquema ineficiente

- Los ritmos de transferencia están limitados por la velocidad a la cual puede la CPU comprobar y servir un dispositivo E/S.
- El tiempo que la CPU dedica a estas tareas podría ser empleado en otras tareas.
- Pueden ocurrir retardos cuando un periférico está esperando por servicios de la CPU. Si hay varios sistemas E/S, cada periférico puede tardar bastante en ser comprobado.
- La E/S programada transmite los datos a través de la CPU, en lugar de permitir que pasen directamente del dispositivo a la memoria.

2. E/S manejada por interrupciones

- El procesador inicializa el dispositivo de E/S y le indica que genere una interrupción cuando haya terminado su operación o cuando necesite algo. Mientras tanto, se dedica a hacer otras tareas
- Se utiliza otro bit: bit HABILITA INTERRUPCIÓN. Se pone a 1 para indicar que el dispositivo E/S requiere algo o ha terminado algo.



2. E/S manejada por interrupciones

- La capacidad de interrupción libera a la CPU de la tarea de comprobar periódicamente el estado de los periféricos.

- **Desventaja:**

Este esquema requiere una interrupción para cada carácter transmitido. Procesar interrupciones resulta caro.

Una interrupción hace que la CPU guarde el estado previo del programa y transfiera el control a un nuevo programa. Cuando la interrupción ha sido servida, la CPU puede reanudar la ejecución del programa interrumpido.

2. E/S manejada por interrupciones

- Una interrupción llega a la CPU, una vez que se finaliza la instrucción en curso almacena la dirección de la siguiente instrucción, busca el programa de tratamiento de la interrupción, lo ejecuta y vuelve al programa que estaba anteriormente en curso.
- Una excepción es un evento que ocurre durante la ejecución de un programa y que requiere la ejecución de un fragmento de código situado fuera del flujo normal de ejecución. El sistema operativo maneja las excepciones.
- Las excepciones son generadas por el hardware o el software:
 - Hardware: división por cero, ejecución de instrucciones ilegales.
 - Software: detectadas por el sistema operativo o el propio proceso.

2. E/S manejada por interrupciones

- Cuando ocurre una excepción, tanto hardware como software, el control es transferido al Sistema Operativo, que ejecuta la rutina de tratamiento de excepción correspondiente
- Existen 256 interrupciones: algunas están ya predefinidas, y otras las define el usuario (**INTERRUPCIONES ENMMASCARABLES**). El manejo de excepciones necesita el soporte del lenguaje de programación para que el programador pueda especificar la rutina a ejecutar cuando se produzca una excepción.
- Si existe una rutina para la excepción generada, el sistema operativo transfiere el control a dicha rutina, en caso contrario aborta la ejecución del proceso.
- Cada rutina se identifica con un número, el **VECTOR**.

2. E/S manejada por interrupciones

- La **TABLA DE PUNTEROS o DE VECTORES DE INTERRUPCIÓN** es el enlace entre el tipo de interrupción y el procedimiento diseñado para servirla. Puede haber hasta 256 entradas en la tabla. Cada entrada es un puntero de doble palabra que contiene la dirección de la rutina de servicio.
-

2. E/S

- La **TABLA** es el encargado de diseñar cada una de las direcciones

Tipo	Excepción	Tipo Ex.
0	Error división	Fallo
1	Paso a paso	Trampa
2	NMI	Interrupción
3	Breakpoint	Trampa
4	INTO (overflow)	Trampa
5	BOUND	Fallo
6	Opcode inválido	Fallo
7	Copro no disponible	Fallo
8	Doble fallo	Aborto
9	Violación de página en copro.	Fallo
10	Estado del segmento de tarea inválido	Fallo
11	Segmento no presente (Mem. Virtual)	Fallo
12	Fallo de pila	Fallo
13	Copro (9)	Fallo
14	Fallo de página (Mem. Virtual)	Fallo
15	Reservada. No usar.	
16	Error coprocesador	Fallo
17	Mala alineación de datos en memoria.	Fallo
18	Comprobación de máquina	Aborto
19-31	Reservadas	
32-255	Interrupciones de usuario	Interrupciones

S

INTERRUPCIÓN
to
en la tabla.
ne la

2. E/S manejada por interrupciones

- Las **interrupciones** y **excepciones** son acontecimientos externos o internos al procesador, que provocan una desviación en el flujo de control de la CPU. Estas anomalías son asíncronas, ya que no se saben cuando van a suceder.
- Las interrupciones se originan por acontecimientos externos (dispositivos E/S). Las excepciones se generan automáticamente como consecuencia de algún acontecimiento anormal, producido y detectado en el desarrollo del programa en curso de ejecución.

2. E/S manejada por interrupciones

TIPOS DE INTERRUPCIONES: externas e internas

1) Interrupciones **externas**:

- Interrupciones activadas por componentes hardware externos, detectadas por el **Controlador de Interrupciones Programable** Avanzado local (APIC) que dispone el procesador.
- Las líneas que provocan las interrupciones externas son la INTR y NMI.

2. E/S manejada por interrupciones

TIPOS DE INTERRUPCIONES:

1) Interrupciones **externas**:

- **NMI**: interrupción **NO ENMASCARABLE**, es siempre atendida por la CPU. Esta interrupción es el resultado de un problema hardware serio (error en memoria o en bus).

Esta interrupción se activa por flanco ascendente, y tras su activación, se atiende mediante la entrada numero 2 de la tabla de punteros de interrupción.

- **INTR**: es una interrupción que se origina por parte de un componente externo al procesador. Se trata de una interrupción **ENMASCARABLE** y su aceptación y puesta en marcha depende del estado del señalizador IF del registro E-FLAGS. Si IF=1, se atiende la interrupción enmascarable, pero si IF=0, no se tiene en cuenta la petición.

2. E/S manejada por interrupciones

TIPOS DE INTERRUPCIONES:

2) Interrupciones **internas**:

- Interrupciones originadas por la ejecución de alguna instrucción especial (interrupciones provocadas por software). Su desarrollo funcional es igual al que corresponde a una excepción.
- Las instrucciones que pueden provocar este tipo de interrupciones son **INT n** y **INTO**.
- INT n es una interrupción NO ENMASCARABLE, generada por software. Siempre que se ejecuta esta instrucción, se salta a la rutina de la interrupción que indique el valor 'n' que apunta a la tabla de punteros de interrupción.
- INTO es una intrusión que salta al vector 4 de la tabla, siempre que el valor del bit OF del registro E-FLAGS valga 1.

2. E/S manejada por interrupciones

TIPOS DE EXCEPCIONES: fallos, trampas y abortos

- Las excepciones son provocadas automáticamente por el procesador al detectar alguna anomalía en el flujo de control. Tipos:
 - 1) **Fallos**: excepciones que se encargan de corregir un error al intentar ejecutar una instrucción, retornando al lugar donde la CPU la dejó, tras la finalización de la excepción. De esta forma la instrucción que la había provocado, se puede realizar. Ej: la CPU ejecuta una operación matemática y aun no tiene todos los operandos que están involucrados en la instrucción.
 - 2) **Trampa**: excepciones que se generan tras la finalización de la instrucción. Ej: interrupciones definidas por el usuario e incluidas en el programa.
 - 3) **Aborto**: excepciones que indican errores muy graves, fallos irrecuperables.

2. E/S manejada por interrupciones

LECTURAS RECOMENDADAS:

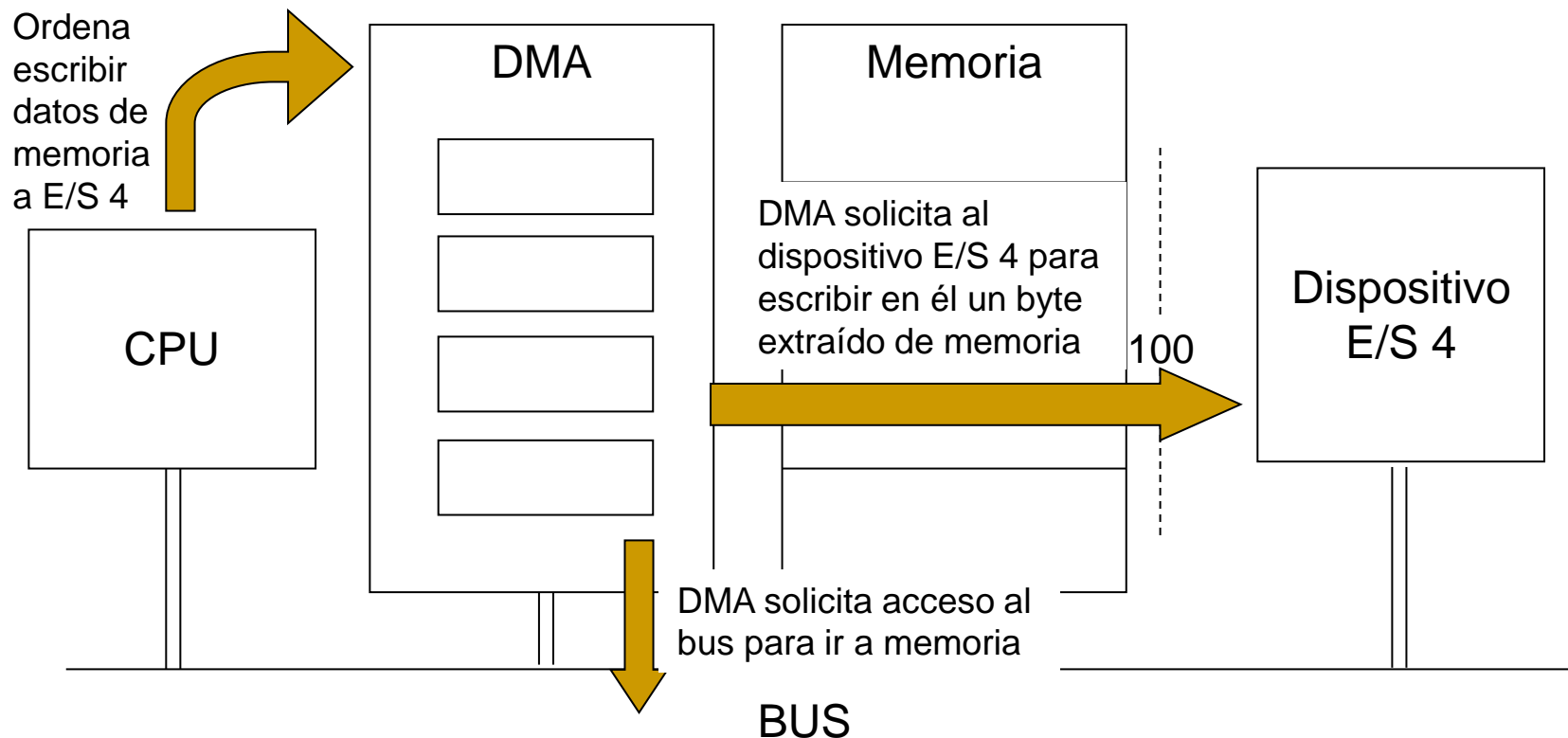
- 1. Ejemplo: interrupciones del 8086/8088:** En los apuntes de la asignatura (pdf)
 - 2. Ejemplo: interrupciones del Pentium y sucesores:**
<http://misnotas.co.cc/Angulo/Capitulo14.html>
-

3. E/S de acceso directo a memoria (DMA)

- Para evitar las interrupciones, se vuelve al primer esquema (E/S programada) pero haciendo que otro dispositivo distinto al procesador controle los bits LISTO y las transferencias de información. Este dispositivo es el CONTROLADOR DE ACCESO DIRECTO A MEMORIA (DMA) y tiene acceso directo al bus.
- El DMA tiene, por lo menos, 4 registros en su interior:
 - Reg. 1: contiene la dirección de memoria para leer/escribir.
 - Reg. 2: número de bytes a transferir (palabras).
 - Reg. 3: número del dispositivo E/S con el que comunicarse.
 - Reg. 4: especifica si los datos serán leídos/escritos en el dispositivo E/S.

3. E/S de acceso directo a memoria (DMA)

Ejemplo: escribir un bloque de 32 bytes de la dirección de memoria 100 al terminal de E/S número 4.



DMA incrementa en 1 (100) su dirección y decrementa en 1 su contador (32). Repite esta operación hasta que no quedan bytes que escribir en E/S 4

3. E/S de acceso directo a memoria (DMA)

- Cuando DMA termina envía una interrupción a la CPU.
- La CPU sólo tiene que inicializar los 4 registros del DMA y luego queda libre para realizar otras tareas, hasta que la transferencia termina y le llega la interrupción de DMA.
- La DMA libera mucho a la CPU de la carga que representan las operaciones de E/S.
- Desventaja: si el DMA se comunica con un dispositivo E/S de alta velocidad se requerirá de muchos ciclos de bus para la transferencia de memoria a los dispositivos. Durante estos ciclos la CPU debe esperar para usar el bus (DMA tiene prioridad sobre CPU). El DMA quita ciclos de bus a la CPU (ROBO DE CICLOS). Aún así, DMA es menos costoso que los otros dos métodos anteriores y se usa mucho en PCs y microcomputadoras.

4. E/S usando canales de datos

- En macrocomputadoras el robo de ciclos sí es muy costoso porque existen muchas operaciones de E/S. El robo de ciclos saturaría al bus.
- Solución: incorporar a estas arquitecturas procesadores especiales de E/S llamados CANALES DE DATOS.
- Un canal es realmente una computadora especializada. Se le puede dar un programa a ejecutar y lo puede hacer sin ninguna ayuda de la CPU principal.

4. E/S usando canales de datos

