

2024

Universidad Tecnológica Nacional [Frba-DISI]



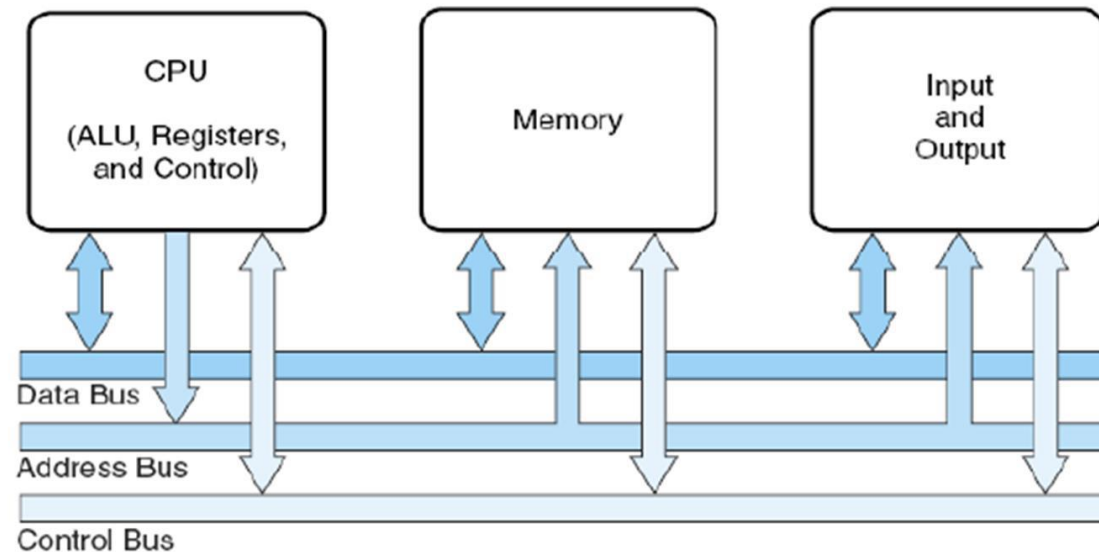
ENTRADA/SALIDA

Unidad XI

Introducción

- En esta unidad vamos a conocer las estructuras básicas utilizadas para la interconexión de componentes de computadoras. Recordemos el esquema básico planteado al inicio de la materia.

Modelo de von Neumann Bus del Sistema



Introducción

- Debido a la necesidad de conectividad de los diversos componentes para formar un sistema, donde se encuentran distintos periféricos que tienen varios métodos de operación (diferentes formatos, diferentes velocidades, etc), se establece una jerarquía por niveles de comunicación presentada en orden consecutivo, desde las **transferencias elementales** que ocurren en la CPU entre buses internos al chip hasta las **transferencias completas** de bloques de información que ocurren en los buses de entrada/salida.
- Bajo el nombre de dispositivos de entrada/salida se clasifican a los componentes hardware que se involucran en la interconexión, como puede ser, un adaptador de red, una controladora de disco, un puerto paralelo o puerto USB, etc.

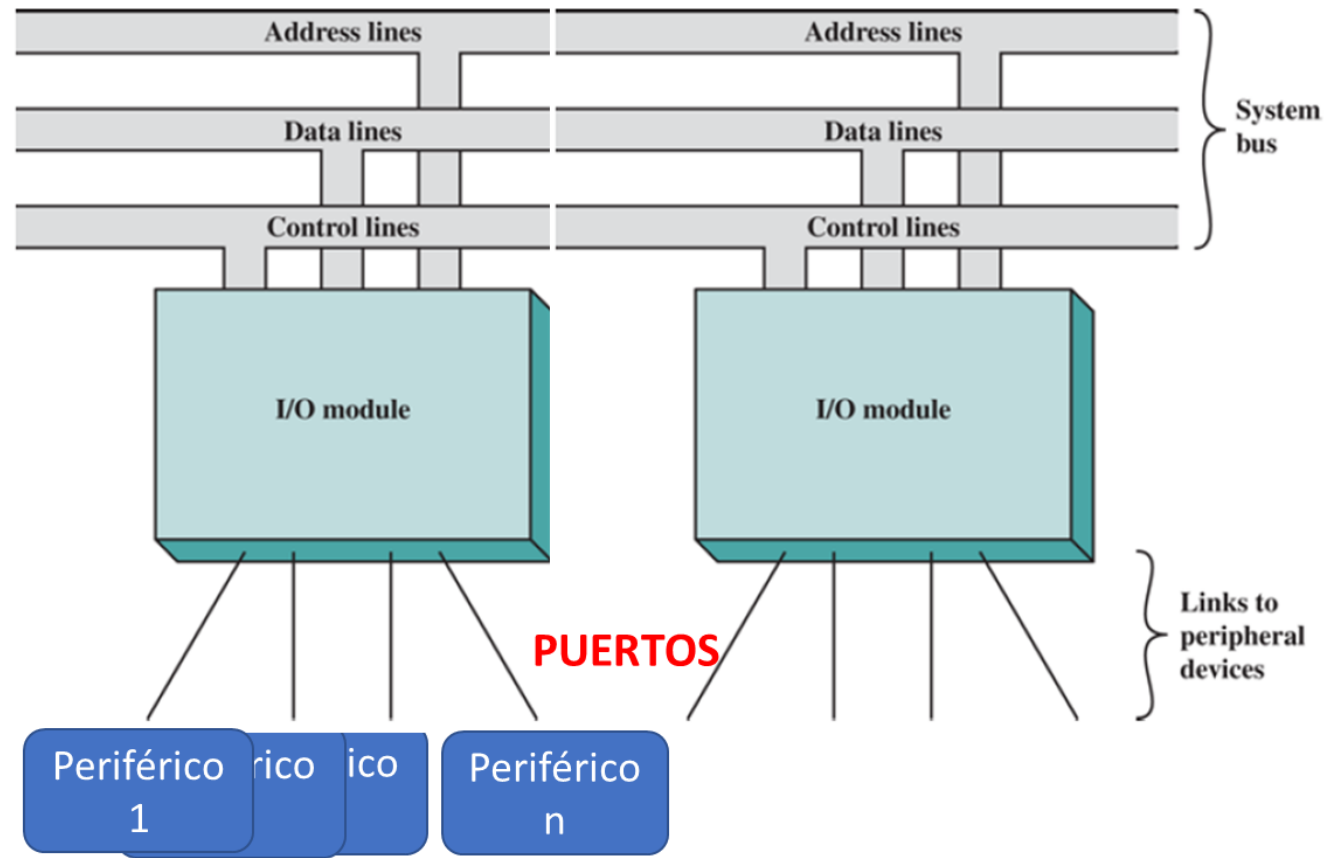
Componentes involucrados en E/S

- Los componentes involucrados son:
 - Los periféricos (interactúan con el medio externo como ser las personas u otros dispositivos).
 - Módulos de Entrada/Salida (Son los que gestionan la comunicación entre el procesador y el periférico).
 - Los buses, que conectan los módulos de Entrada/Salida con la CPU y Memoria.
- Dentro de los buses están el bus de direccionamiento, el bus de datos y el bus de control los cuales transmiten los unos y ceros de la información.
- Estas transferencias se pueden realizar con el control o no por el reloj del sistema, siendo sincrónicas o asincrónicas respectivamente.

Dispositivos/Operaciones de entrada/salida

- Se denominan dispositivos de E/S tanto las unidades periféricas en sí como aquellas “intermediarias”, que se encargan de efectivizar una transferencia (Operación) entre la memoria interna y la memoria externa en los periféricos. Podemos mencionar en este caso a controladores, adaptadores, puertos e interfaces. Como en toda transferencia se utilizan señales de control, dato y dirección, en este caso serán utilizadas sobre los **buses en E/S**.
- Las señales de control y tiempo se utilizan para regular la transferencia elemental e indicarnos las siguientes funciones:
 - Comunicarse con el periférico y el sistema CPU-memoria.
 - Controlar la temporización durante la transferencia.
 - Almacenar temporalmente bits para “paliar” la diferencia de velocidad entre emisor y receptor.
 - Detectar si se produjeron errores durante la transferencia.

E/S Esquema General

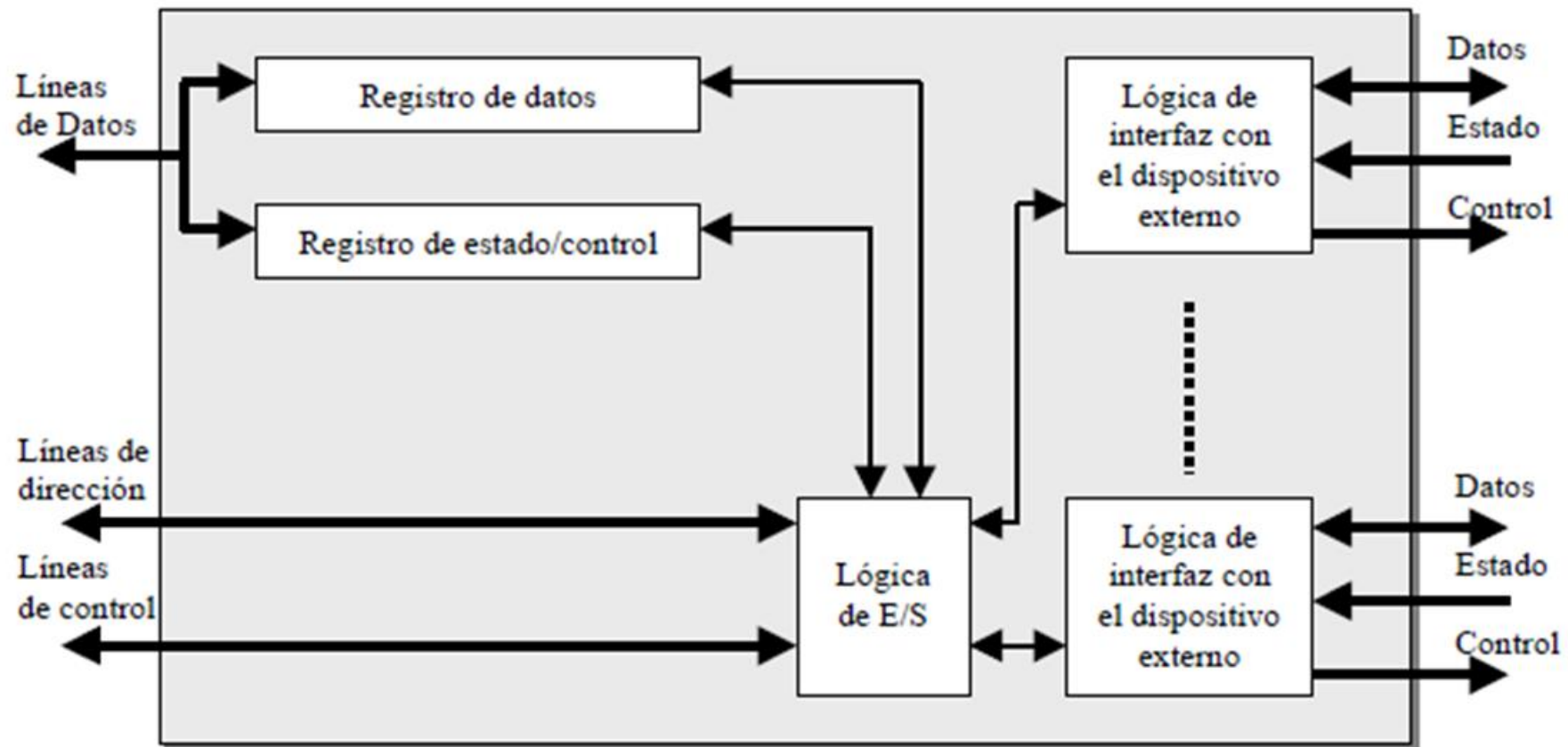


Consideraciones:

- Un módulo puede controlar más de un periférico.
- Cada una de las interfaces de un periférico la referimos como un PUERTO.

Módulo I/O

- A continuación, podemos observar cómo resulta un Módulo de I/O



Módulo I/O

- Funciones de un módulo. Las principales funciones y requisitos de un módulo de I/O se encuentran dentro de las siguientes categorías:
 - Control y temporización
 - Comunicación con el procesador
 - Comunicación con los dispositivos
 - Almacenamiento temporal de datos
 - Detección de errores

Módulo I/O - Interface

- **Interfaces** → Una interfaz es un hardware que actúa de nexo entre un periférico o un adaptador y el bus. Sirve, en primer término, para adecuar las señales y preparar la transferencia elemental basada en un protocolo. No tiene capacidad suficiente para tomar la responsabilidad de la transferencia completa (bloque), son la CPU o el canal los que asumen el control de una transferencia completa como se verá más adelante.
- Resumiendo, las interfaces son las encargadas de informar a la CPU del estado del periférico y de ser la puerta (PORT) de entrada y salida correspondiente a la transmisión de los datos de entrada y salida que llegan o se dirigen a la memoria interna del procesador.

CPU-memoria

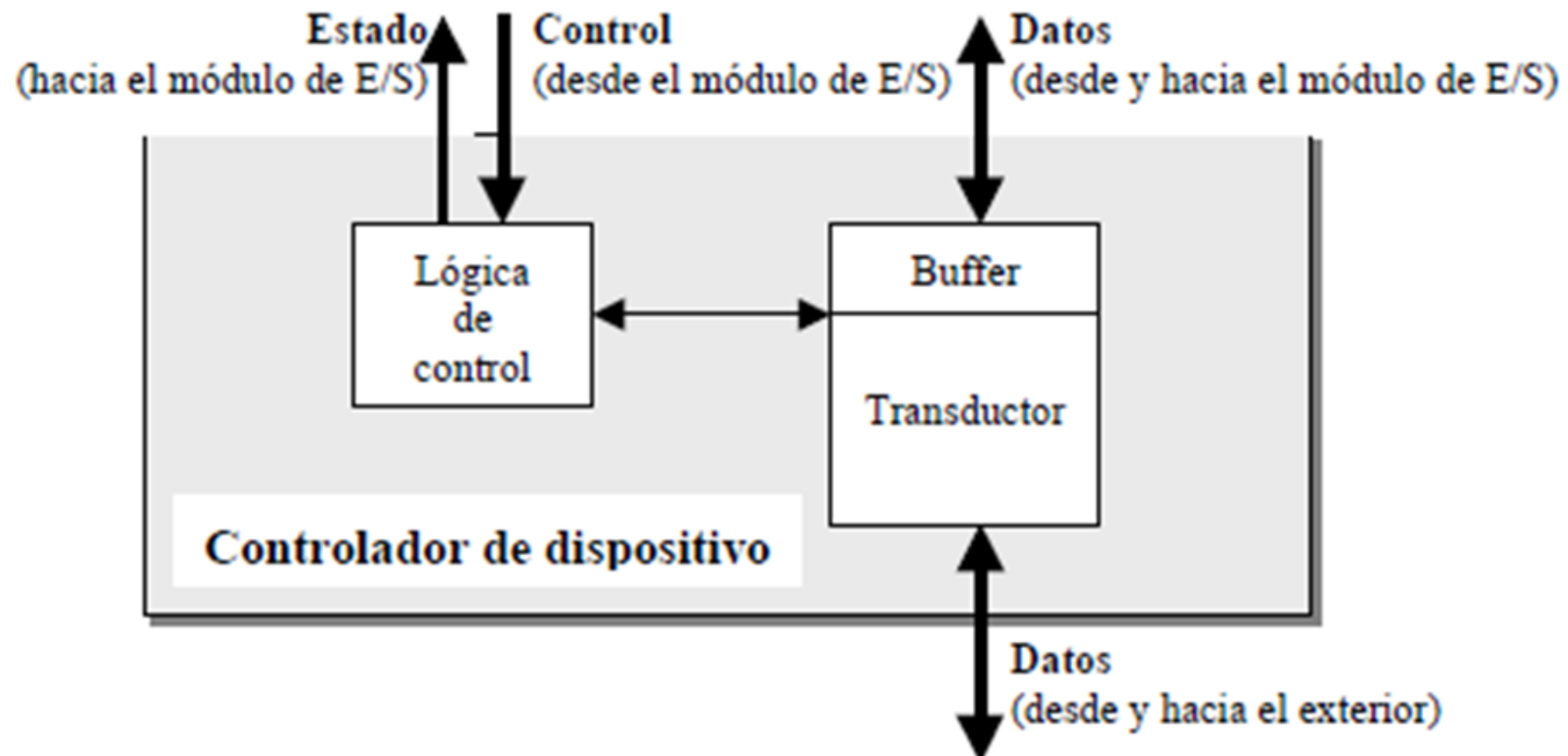


Controlador - periférico

Módulo I/O - Interface

- Las interfaces, básicamente van a estar constituidas por una serie de registros. Dentro de los mismos y dependiendo del modo de realización de la operación de Entrada/Salida, figuran:
 - 1) REP (Registro Estado del Periférico): Como su nombre lo indica, contendrá el estado en que se encuentra el periférico, básicamente si está o no disponible para recibir o enviar datos.
 - 2) RB (Registro Buffer): Es la puerta (PORT) de entrada ó salida por la cual pasarán los datos transferidos desde la memoria interna al periférico o viceversa.
 - 3) RDI (Registro de direcciones de la Interface): A través del mismo y si tiene la tecnología adecuada (DMA – Direct Access Memory), podrá direccionar a través del mismo posiciones de la memoria interna.
 - 4) Otros registros que pueden existir: Registro de Errores, Registro Contador de Palabras, etc

I/O Controladora

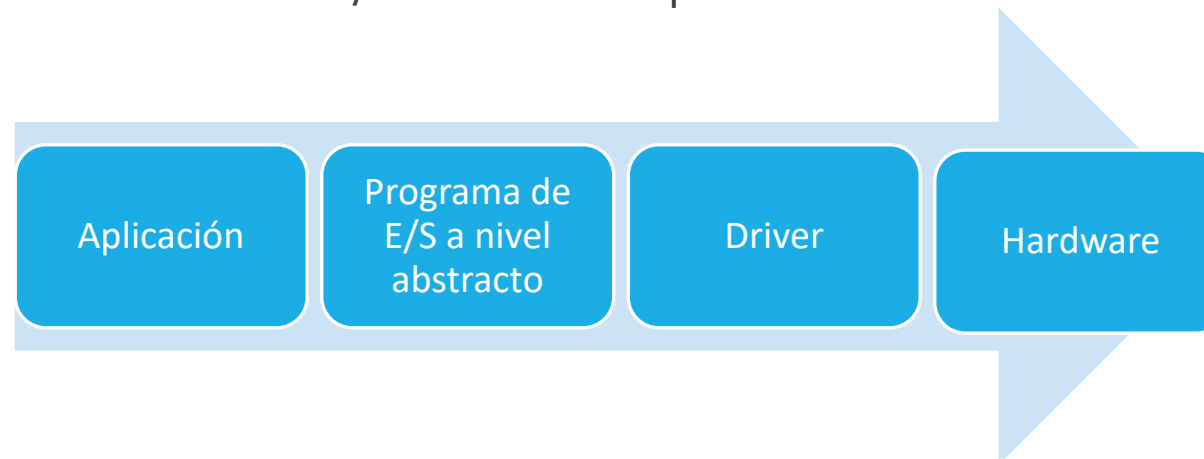


I/O Controladora

- **Controladores** → El término controlador se utiliza en gran medida para definir cualquier unidad de hardware “que gobierne” a otra. O sea, un **controlador de periférico** es un dispositivo asociado en forma directa al periférico, que puede estar físicamente integrado a él, o bien separado de éste, y está constituido por:
 - Un buffer interno (memoria RAM), que permite el almacenamiento de la información que “viaja” desde o hacia el soporte.
 - Una lógica de control, que interpreta comandos de periférico, genera señales para su ejecución y gobierna así la unidad.
 - Va a realizar todos los movimientos de gobierno de un periférico como por ejemplo, en una impresora de chorro a tinta, del movimiento mecánico del cabezal, de la hoja y de la inyección de tinta para imprimir un determinado carácter.

Modalidades de entrada/salida

- El sistema operativo cuenta con programas que gestionan las transferencias de entrada/salida, pero esta vez en un nivel superior o, si lo queremos ver de otro modo, un nivel más abstracto. Estos programas realizan funciones comunes a todos los dispositivos periféricos, más allá de las particularidades de cada uno de ellos.
- Los programas de aplicación llaman a los programas de E/S del sistema operativo mediante una “llamada al sistema”, que puede expresarse de distintas maneras según el sistema operativo que se utilice.
- La relación entre el programa de aplicación en estado de ejecución como peticionario del servicio de transferencia y el hardware de entrada/salida como su proveedor se muestra en el siguiente ciclo:



Modalidades de entrada/salida

- Como se indicó anteriormente una transferencia elemental es un mensaje que migra de un emisor a un receptor y para realizar una transferencia completa se necesitan varias transferencias elementales. Para poder llevar a cabo dicha transferencia se pueden identificar diferentes modos o métodos:
 - Transferencia controlada por programa.
 - Transferencia iniciada por interrupción.
 - Transferencia con acceso directo a memoria.
 - Transferencia a través de un procesador IOP o modo canal.

Transferencia controlada por programa

- El procesador interroga al módulo de E/S para comprobar el estado del dispositivo conectado al mismo (Lo hace a través del registro REP) .
- El módulo de E/S devuelve el estado del dispositivo.
- Si el dispositivo está operativo y preparado para transmitir, el procesador solicita la transferencia del dato mediante una orden al módulo de E/S.
- El módulo de E/S obtiene un dato del dispositivo externo o de la memoria interna según sea la transferencia. Los datos se transfieren desde el módulo de E/S al procesador o viceversa (a través del registro RB).
- Si el sistema utiliza un bus, entonces cada una de las interacciones entre el procesador y el módulo de E/S implica uno o más arbitrajes del bus
- Este proceso tiene la desventaja de la gran utilización de la CPU impidiendo la ejecución de otras instrucciones.

Transferencia controlada por programa

- Con la E/S *programada*, los datos se intercambian entre el procesador y el módulo de E/S.
- El procesador ejecuta un programa que controla directamente la operación de E/S, incluyendo la comprobación del estado del dispositivo, el envío de una orden de lectura o escritura y la transferencia del dato.
- Cuando el procesador envía una orden al módulo de E/S, debe esperar hasta que la operación de E/S concluya.
- Si el procesador es más rápido que el módulo de E/S, el procesador desperdicia este tiempo.

Transferencia iniciada por interrupción

- Esto mejora los tiempos de desperdicio de la modalidad anterior, donde ahora el procesador proporciona la orden de E/S a través de permitirle al módulo de I/O que lo interrumpa cuando esté disponible.
- Por lo tanto, permite que el procesador continúe ejecutando otras instrucciones, y pueda ser interrumpido por el módulo de E/S cuando éste ha terminado su trabajo.
- Tanto con E/S programada como con interrupciones, el procesador es responsable de extraer los datos de la memoria principal en una salida, y de almacenar los datos en la memoria principal en una entrada.

Transferencia iniciada por interrupción

- En que consiste el llamado por Interrupción:
 - Cuando la interfaz a través de su REP detecta que el dispositivo está listo, el módulo de I/O genera una interrupción a través de una señal llamada IRQ #(Requerimiento de interrupción), donde el # será el número de interrupción correspondiente al dispositivo que interrumpe.
 - La CPU al final de cada ejecución de una instrucción comprueba si existe alguna petición de interrupción pendiente y en caso de haberla, interrumpe la tarea en curso y ejecuta una subrutina llamada SAI (Subrutina de Atención de la Interrupción), que atiende dicho requerimiento.
 - Una vez atendido el requerimiento, la CPU reanuda la tarea donde la había dejado.

Transferencia con acceso directo a memoria

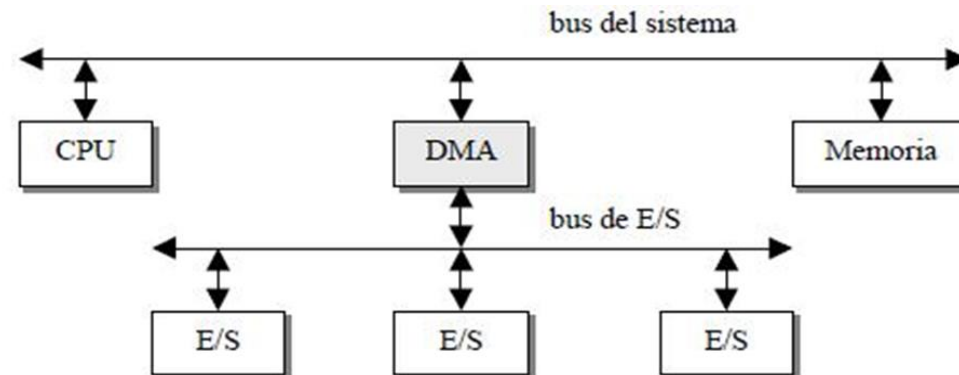
- Tanto con E/S programada como con interrupciones, el procesador es responsable de extraer los datos de la memoria principal en una salida, y de almacenar los datos en la memoria principal en una entrada.
- En este caso, en Acceso directo a memoria (DMA), el módulo de E/S y la memoria principal intercambian datos directamente, sin la intervención del procesador.
- Dicho controlador permite gobernar actividades del sistema, liberando a la CPU de ciertas tareas que pueden llevarse a cabo de manera independiente.
- O sea, posibilita una transferencia entre un dispositivo de entrada/ salida y la memoria interna, sin intervención de la CPU, sólo necesita el permiso de la misma.
- A estos dispositivos se los denomina maestros, ya que tienen el control del bus en un momento determinado.

Transferencia con acceso directo a memoria

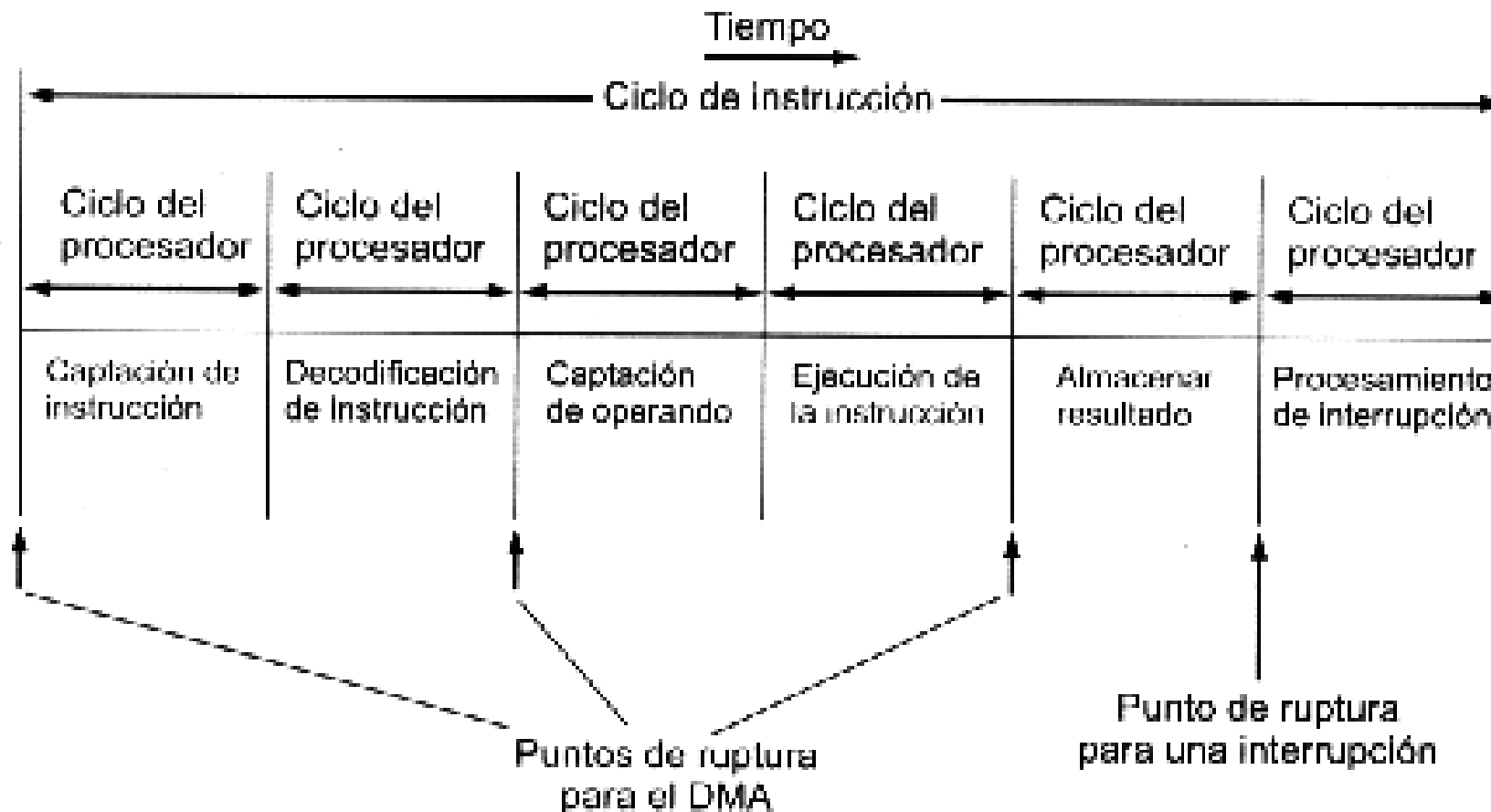
- Funcionamiento del DMA. La E/S es útil en SO multitarea o monotarea si mueve los datos más rápidamente que el procesador, los pasos son:
 1. Programación del controlador DMA.
Tipo de operación a realizar (L/E), número de datos a transferir y direcciones involucradas en la transferencia.
 2. Programación de la interfaz.
Se indica a la interfaz que ha de usar DMA para la transferencia.
 3. Interfaz lista, solicitud de DMA
La interfaz avisa el controlador para iniciar la transmisión
 4. Transmisión
El controlador de DMA va transfiriendo uno a uno los datos en los periodos en los que tiene el control de los buses.
La transmisión finaliza cuando el contador de datos a transmitir = 0.
 5. Fin de la transmisión
Se notifica la CPU mediante una interrupción que se ha completado la transmisión.

Transferencia con acceso directo a memoria

- El DMA requiere de un módulo adicional en el bus del sistema.
- El módulo de DMA es capaz de imitar al procesador y de hecho es capaz de recibir el control del sistema cedido por el procesador.
- Para hacerlo, el módulo de DMA debe utilizar el bus sólo cuando el procesador no lo necesita o debe forzar al procesador a que suspenda temporalmente su funcionamiento.
- Esta última técnica es la más común y se denomina **robo de ciclo**.



DMA por robo de ciclo de bus



Instrucciones y Ejemplos de I/O

- Los siguientes son algunos códigos para utilizar en interrupciones de Entrada/Salida
 - INT 21
 - INT 10
- Se muestran a continuación ejemplos utilizando las mismas y el emulador del 8086

Instrucciones y Ejemplos de I/O

1) Int 21h / AH=9

En este ejemplo con la interrupción 21 y la configuración AH=9; permitirá desplegar una cadena o mensaje que hemos creado previamente en el registro de datos DS:DX. El mensaje o cadena se declara como una variable de tipo dato byte DB y despues el mensaje que va entre comilla y que debe terminar con el signo \$ para indicar el fin de cadena.

```
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 NAME "PROGRAMA 1"; NOMBRE DE MI PROGRAMA
02
03 ORG 100h;CS:IP=0100 QUE ES DONDE COMENZARA EL CODIGO
04
05 MOV AH, 0; PARA CONFIGURAR EL VIDEO
06 MOV AL, 00h; PANTALLA DE 40X25 PÍXELES DE 16 COLORES
07 INT 10h; CONFIGURAR EL MODO DE VIDEO
08
09
10 MOV DX,OFFSET mensaje; ASIGNAMOS EL MENSAJE AL REGISTRO DE DATOS DX
11 MOV AH,9h; CONFIGURACIÓN NECESARIA PARA DESPLEGAR EL MENSAJE
12 INT 21h;INTERRUPCION PARA DESPLEGAR EL MENSAJE EN LA CONSOLA
13
14
15 RET
16 mensaje DB "Mi primer mensaje de E/$";CREACION DEL MENSAJE O CADENA QUE DESEAMOS DESPLEGAR EN PANTALLA
17
18
```


Instrucciones y Ejemplos de I/O

2) Int 21h / AH=1

En este ejemplo con la interrupción 21h y la configuración AH=1; permitirá leer un carácter desde la consola y lo va a guardar en el registro AL. Es decir, con la configuración de AH=1 e interrupción 21h permite que el usuario ingrese un carácter desde la consola. Además, retorna el código ASCII en el registro AL:

```

file  edit  bookmarks  assembler  emulator  math  ascii codes  help
new  open  examples  save  compile  emulate  calculator  convertor  options  help  about

00000100  ORG 100h; CS:IP=0100 QUE ES DONDE COMENZARA EL CODIGO
00000200  MOV AH, 0; PARA CONFIGURAR EL VIDEO
00000300  MOV AL, 00h; PANTALLA DE 40X25 PÍXELES DE 16 COLORES
00000400  INT 10h; CONFIGURAR EL MODO DE VIDEO

00000500  MOV DX, OFFSET mensajeN1; ASIGNAMOS EL MENSAJE AL REGISTRO DE DATOS DX (SOLICITAMOS EL PRIMER NUMERO POR PANTALLA)
00000600  MOV AH, 9h; CONFIGURACION NECESARIA PARA DESPLEGAR EL MENSAJE
00000700  INT 21h; INTERRUPCION PARA DESPLEGAR EL MENSAJE EN LA CONSOLA

00000800  MOV AH, 1; CONFIGURACION PARA LEER EL CARACTER POR CONSOLA
00000900  INT 21h

00000A00  MOV AH, 0Eh; CONFIGURACION PARA UN SOLO CARACTER
00000B00  MOV AL, 10; EL CODIGO 10 CORRESPONDE AL ASCII DE SALTO DE PAGINA
00000C00  INT 10h; INTERRUPCION

00000D00  MOV AH, 0Eh; CONFIGURACION PARA UN SOLO CARACTER
00000E00  MOV AL, 13; EL CODIGO 10 CORRESPONDE AL ASCII DE SALTO DE PAGINA
00000F00  INT 10h; INTERRUPCION

00001000  MOV DX, OFFSET mensajeN2; ASIGNAMOS EL MENSAJE AL REGISTRO DE DATOS DX (SOLICITAMOS EL SEGUNDO NUMERO POR PANTALLA)
00001100  MOV AH, 9h; CONFIGURACION NECESARIA PARA DESPLEGAR EL MENSAJE
00001200  INT 21h;

00001300  MOV AH, 1; CONFIGURACION PARA LEER EL CARACTER POR CONSOLA
00001400  INT 21h

00001500  RET
mensajeN1 DB "Ingrese el Primer numero entero: $"; SE SOLICITA AL USUARIO QUE INGRESE POR CONSOLA UN PRIMER NUMERO ENTERO
mensajeN2 DB "Ingrese el Segundo numero entero: $"; SE SOLICITA AL USUARIO QUE INGRESE POR CONSOLA UN PRIMER NUMERO ENTERO

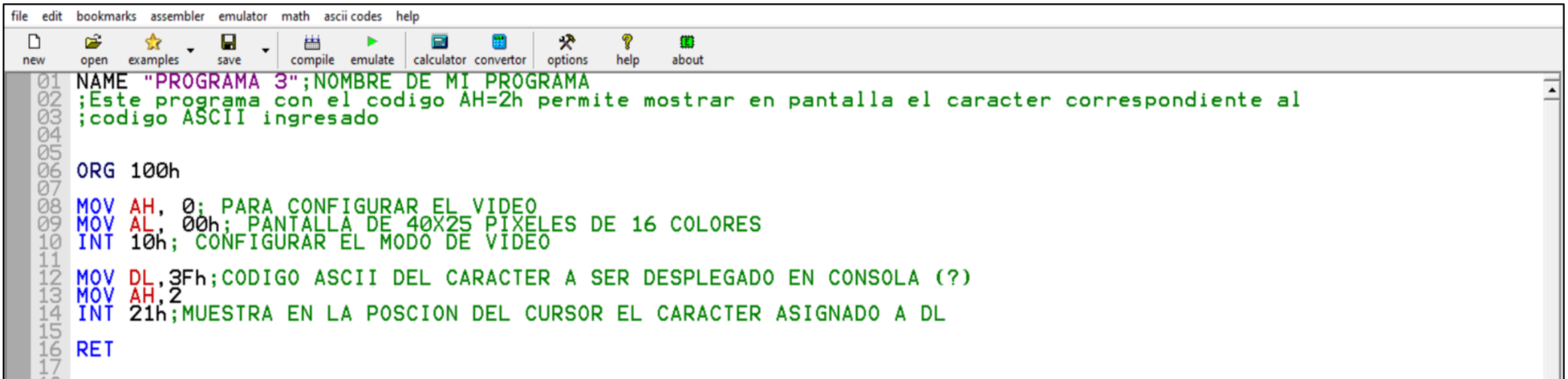
```

Resolver: agregar las líneas de código necesarias a este código para salvaguardar el dato ingresado por el usuario como “Primer número entero” y no sobrescribirlo con el ingreso del “Segundo número entero”.

Instrucciones y Ejemplos de I/O

3) Int 21h / AH=2

En este ejemplo con la interrupción 21h y la configuración del registro AH=2; permitirá mostrar en pantalla o consola el caracter correspondiente al código ASCII ingresado.



```
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 NAME "PROGRAMA 3";NOMBRE DE MI PROGRAMA
02 ;Este programa con el codigo AH=2h permite mostrar en pantalla el caracter correspondiente al
03 ;codigo ASCII ingresado
04
05
06 ORG 100h
07
08 MOV AH, 0; PARA CONFIGURAR EL VIDEO
09 MOV AL, 00h; PANTALLA DE 40X25 PÍXELES DE 16 COLORES
10 INT 10h; CONFIGURAR EL MODO DE VIDEO
11
12 MOV DL, 3Fh; CODIGO ASCII DEL CARACTER A SER DESPLEGADO EN CONSOLA (?)
13 MOV AH, 2
14 INT 21h; MUESTRA EN LA POSICION DEL CURSOR EL CARACTER ASIGNADO A DL
15
16 RET
17
```

Preguntas?

