



Entrada y Salida (E/S)

Rafael Ignacio Zurita
rafa@fi.uncoma.edu.ar

Rodolfo del Castillo
rdc@fi.uncoma.edu.ar

Apunte de cátedra ^[a]

Resumen

En las unidades anteriores hemos examinado la estructura interna y operación de una unidad de procesamiento central (CPU). De cualquier manera, para que una computadora tenga un valor real para las personas, debe poder comunicarse de alguna manera con sus usuarios. Después de todo, no tiene ningún sentido en diseñar y crear una super computadora del tamaño de un clip, y que pueda resolver la última duda existencial de la vida, el universo, y todo lo demás, si no puede, de alguna manera, decirnos la respuesta. En este apunte se presentan los caminos y mecanismos por los cuales la información entra y sale de una computadora, y los dispositivos o periféricos conectados a la computadora para llevar a cabo este propósito (por ejemplo, una impresora y el monitor led).

2 Subsistema de Entrada y Salida

La organización básica de una computadora consiste de un procesador, la memoria, y el subsistema de entrada y salida (E/S). El **subsistema de entrada y salida** está compuesto de dispositivos de E/S (o periféricos) y controladores de dispositivos, interconectados al procesador y memoria mediante buses del sistema.

entrada y salida

La **entrada y salida** son señales o datos recibidos y enviados por el sistema, y permite a la computadora comunicarse con el mundo exterior, tal vez con una persona (usuario) o con otro sistema (procesador). Por ejemplo, mediante la entrada y salida la computadora puede cargar programas y datos en la memoria (por ejemplo desde un medio de almacenamiento como el disco rígido), presentar datos a sus usuarios (en una pantalla), recibir entrada (desde un teclado y mouse), o enviar datos a otros sistemas mediante la red.

Existe una amplia variedad de periféricos que pueden ser parte de un subsistema de entrada y salida. Los dispositivos más comunes en una PC son los discos rígidos, teclado, mouse, monitor, impresora, escaner, cámara, microfono, parlantes, leds, placas de red, lectoras de CD/DVD/SD, gamepads, etc. Un otro ejemplo son los smartphones, en donde existen también muchos periféricos, como lo son la pantalla, el táctil sobre la pantalla, botones, sensores de huellas digitales, luz del flash, leds, cámaras, gps, acelerómetros, modem de comunicaciones, parlante, microfono, radio wireless, radio FM, etc.

Debido a que todos los dispositivos son lentos comparados con la velocidad de ejecución de la CPU los sistemas están diseñados para solapar el procesamiento con

[a] Obra derivada con permiso escrito (detalles de las obras originales y permisos en la última sección) de artículos de Alan Clements, para las materias de arquitecturas de computadoras, de la universidad de Teesside, Inglaterra; y de Computer Programming and Architecture the VAX-11, Henry Levy, Digital Press (1980).

la E/S. Esto significa, por ejemplo, que si el procesador inicia una operación de E/S en un dispositivo puede seguidamente continuar ejecutando instrucciones de un programa. Luego, el procesador verificará (en un momento conveniente) si la operación se completó; o alternativamente el dispositivo puede enviar una señal al procesador cuando se haya completado la operación.

Las diferentes velocidades entre procesador, memoria y periféricos, influyen, también, la organización de estos componentes dentro de la computadora, por lo que es usual que varios buses sean utilizados para interconectarlos. Antiguas mini-computadoras tuvieron dos o más buses, uno para conectar la memoria a la CPU, y otro para conectar los dispositivos de E/S a la CPU. Un posterior refinamiento, por ejemplo en la PDP-11 Unibus, utilizó un unico bus para todas las transferencias entre periféricos de E/S, memoria y la CPU. Las computadoras personales PC de arquitectura x86 tuvieron también distintas organizaciones de conexión. En la Figura 1 se presenta un diagrama de bloques de una PC típica de los años 2000.

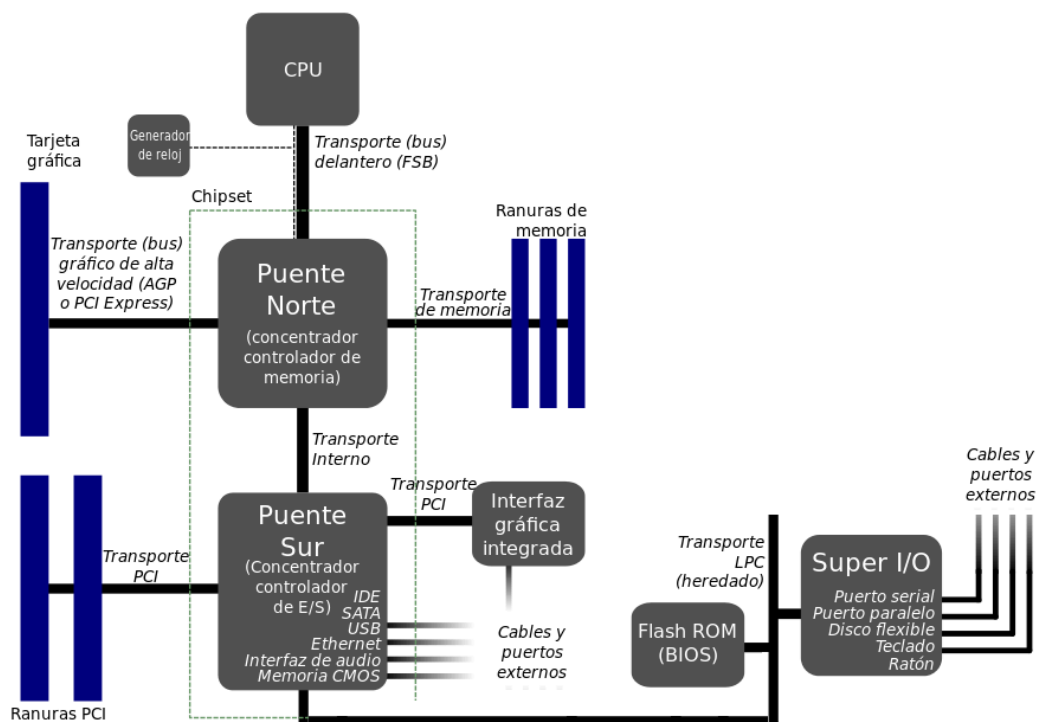


Figure 1: Estructura de CPU, memoria, chipsets y buses de una PC típica en el año 2000.

bus del sistema

Haya uno o mas buses en el sistema el propósito es siempre el mismo: acarrear señales de control, datos y direcciones entre los componentes conectados al bus. En la Figura 2 puede observarse un esquema general de este objetivo. Las señales de dirección permiten a un programa seleccionar entre diferentes dispositivos de E/S conectados al sistema, mientras que las líneas de datos accarean la información actual que se está transfiriendo. Las señales de control especifican qué tipo de operación se desea realizar (leer, escribir, señalar un interrupción, etc). Además, no es necesario que todos los buses sean del mismo tamaño si el sistema tiene mas de uno. Por ejemplo, el bus de memoria del antiguo procesador 8086 tiene 20 líneas de dirección y 16 líneas de datos, mientras que el bus de E/S tiene 16 líneas de dirección

y 8 o 16 líneas de datos. Sea cual sea la organización de interconexión es en el bus del sistema en donde la información de la computadora fluye. Cualquier componente conectado puede colocar información en el bus, y cualquier otro puede tomar información. Por lo que la utilización de un bus en las computadoras establece una organización sencilla para el pasaje de información entre las diferentes unidades funcionales inter conectadas.

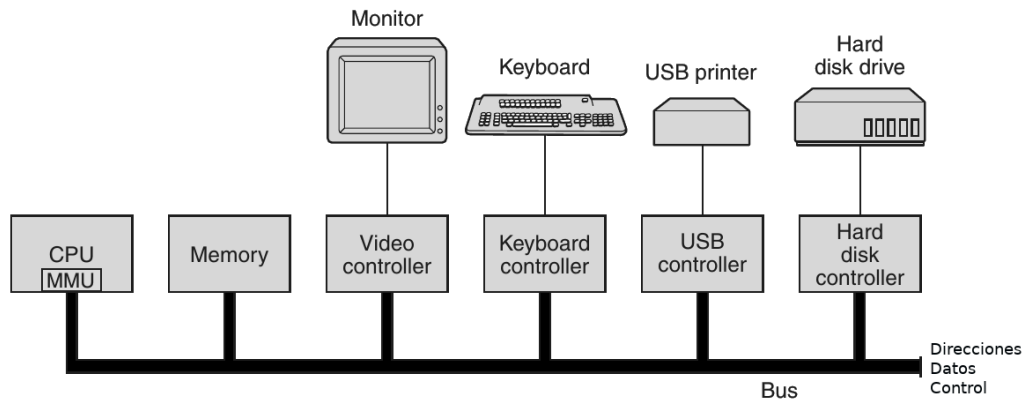


Figure 2: Componentes de una computadoras interconectados por un bus.

3 Periféricos, controladores e Interfaces

periférico Un **dispositivo de E/S (o periférico de E/S)** es un equipamiento hardware que realiza alguna función de comunicación con el mundo exterior para la computadora.

controlador Un **controlador de E/S** dirige la operación del periférico de acuerdo a comandos recibidos desde el procesador. También, convierte los datos recibidos por el sistema en el formato requerido por el dispositivo, y viceversa.

Usualmente el periférico en sí se encuentra fuera de la parte visible de la CPU (chip/motherboard), mientras que casi siempre, el controlador se encuentra junto con el procesador y la memoria. Muchos periféricos diferentes tienen el objetivo de convertir datos del mundo exterior de la computadora a datos que esta pueda procesar, estos son por ej.: teclados, escáneres, joysticks, sensores, etc; otros periféricos convierten datos de la computadora dentro de formas que se utilicen en el mundo exterior: monitores, impresoras, etc. El propósito de algunos es simplemente almacenar gran cantidad de datos para utilizarlos mas adelante: discos rígidos, pen drives, memorias flash, cintas magnéticas, etc.

¿Es el controlador parte del periférico? Algunas veces la línea divisoria entre el periférico y su controlador es confusa. Como ejemplo, observe el teclado y su controlador en la Figura 3. El circuito decodificador del teclado (keyboard scan) convierte una depresión de una tecla en un número de 7bits. Además, la interfaz (controlador) conectada al bus ubica este número en el bus de datos si el procesador lo requiere. Pareciera que el periférico consiste del bloque decodificador y de los bloques de interfaz con el bus (controlador). En un sistema típico, el decodificador se encuentra empaquetado con el teclado mismo, y la mayoría de los diseñadores de computadoras dicen que el bloque

decodificador es parte del teclado y el controlador consiste sólo del bloque de interfaz conectado al bus (dentro de la computadora). Afortunadamente la línea divisoria no tiene importancia en programas de E/S que controlen un teclado, más importante es el modelo de programación de E/S que un programa utiliza.

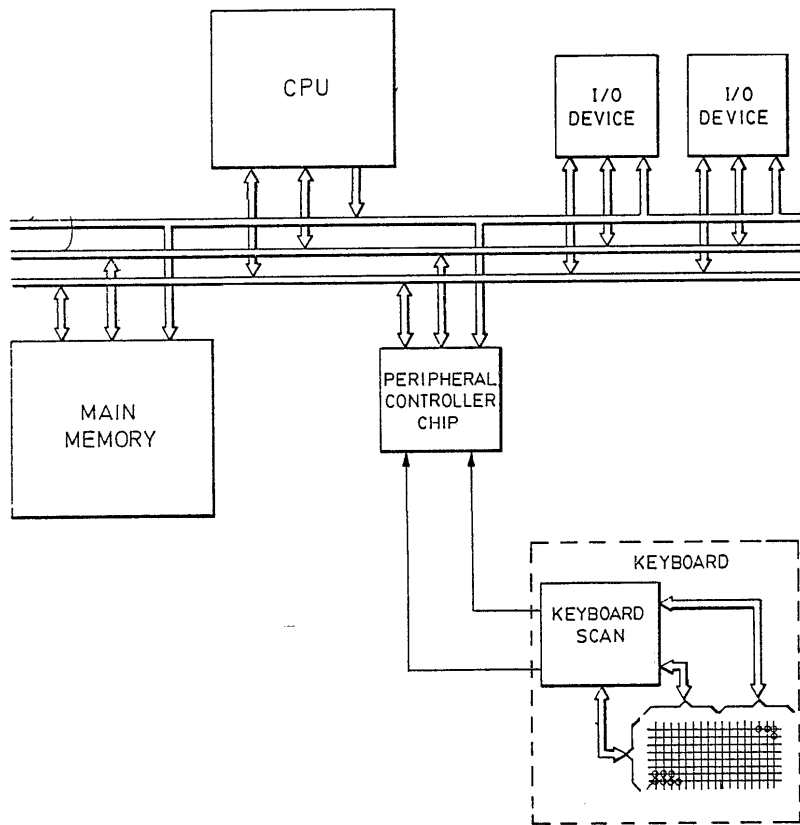


Figure 3: Diagrama de bloques de un teclado y su controlador.

puertos de E/S

La **interfaz de E/S (o puertos de E/S)** es una parte del controlador del dispositivo (ver Figura 4). Está compuesto por registros de estado, control y datos; y son accedidos (generalmente) por el procesador durante operaciones de entrada y salida. Los registros de estado proveen información del estado del dispositivo de E/S a la CPU. Estos registros son frecuentemente de sólo-lectura, lo que significa que la CPU puede leerlos, pero no cambiar su contenido. Los registros de control son utilizados por la CPU para configurar y controlar el dispositivo. Los bits en estos registros podrían llegar a ser de sólo-escritura, lo que significa que la CPU los puede alterar, pero no leerlos. Igualmente, en muchos sistemas y dispositivos estos pueden ser también de lectura y escritura. Los registros de datos son utilizados para leer o enviar datos desde y hacia el dispositivo de E/S.

En algunas circunstancias, un registro podría tener más de una de las categorías anteriores, por ejemplo, un registro podría tener ciertos bits para configurar el dispositivo, y otros bits para obtener información del estado. Dispositivos sencillos como el teclado y ratón pueden ser representados con sólo unos pocos registros, mientras que dispositivos más complejos como los discos y adaptadores gráficos podrían tener docenas de registros. El modelo de programación de E/S del dispositivo (su

documentacion, hoja de datos, etc) describe todos los registros asociados con el dispositivo, su contenido, y la forma de utilizarlos.

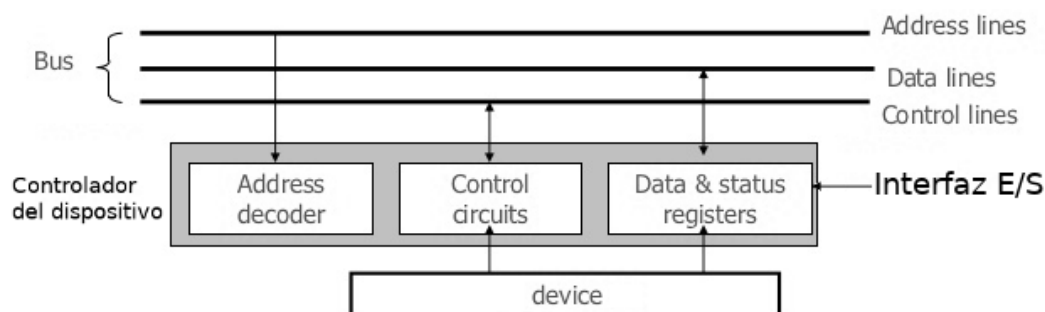


Figure 4: Componentes del controlador del dispositivo.

Por ejemplo, el dispositivo UART conectado al CBUS de la placa de desarrollo Malta (cpu MIPS) contiene 8 registros de 8 bits cada uno. Si conectado a este periférico se encuentra una terminal (pantalla y teclado) el registro RXTX del UART contendrá el código ASCII de cada tecla presionada en la terminal. Para leer un dato desde el teclado, un programa debe ejecutar alguna instrucción que transfiera el contenido de RXTX dentro de un registro del procesador. Una vez que el dato se encuentra en el procesador este puede ser manipulado con otros datos.

Aunque si bien el controlador del UART escribe datos provenientes del teclado de la terminal en RXTX, el procesador puede transferir también a ese registro del UART un código ASCII. Si lo hace, la interfaz del dispositivo enviará el dato a la terminal, que una vez recepcionado, lo mostrará en pantalla. Por lo tanto, el UART es un dispositivo de entrada y salida.

Desafortunadamente, el mecanismo anterior no es tan sencillo de programar debido principalmente a la gran diferencia de velocidad entre la CPU y el dispositivo UART. Si el procesador necesita enviar un dato a pantalla a través del registro RXTX, debe verificar primero (al menos) el estado del dispositivo, para conocer si el periférico está en condiciones de recibir nuevos datos (o si está ocupado, por ejemplo, procesando algún dato anterior). Para esto, un programa debería ejecutar alguna instrucción que transfiera el contenido del registro LSSTAT (Line Status Register) a un registro del procesador, para procesar su contenido. Si el contenido de este registro indica que el periférico está en condiciones de aceptar datos, el programa puede entonces transferir el código ASCII al registro RXTX, para mostrarlo en la pantalla.

4 Programación de E/S

En la programación de la E/S se debe utilizar alguna técnica soportada por el hardware para transferir datos hacia (y desde) los dispositivos (hacia los registros de la interfaz del controlador del dispositivo). Los datos pueden ir desde el dispositivo hacia el procesador, o hacia la memoria; o si el dispositivo es de salida los datos serán transferidos desde la CPU o la memoria hacia la interfaz del controlador del dispositivo (registros).

Las tres estrategias (técnicas) utilizadas en la actualidad son: **E/S programada (polling)**, **E/S mediante interrupciones**, y **E/S mediante acceso directo a memoria (DMA)**. La estrategia utilizada debe estar soportada por los modelos de programación de la com-

putadora y los dispositivos en particular.

E/S mapeada en memoria

Para comenzar una operación de "entrada" el programa debe iniciar una transferencia de datos desde el dispositivo (desde su interfaz). Por ejemplo, un programador podría escribir, en una máquina miniMIPS, la instrucción `lb $t3, 0xffff0004($zero)` la cual lee el código ASCII desde el **registro de datos** de la interfaz del teclado, y lo coloca en el registro de la CPU `t3`. En este caso, el programador utilizó una instrucción "cargar byte", la cual es la misma instrucción que se utiliza en MIPS para cargar en registros de la CPU datos que estén en la memoria RAM. Además, en vez de utilizar una dirección de memoria RAM, el programador utilizó la dirección efectiva `0xffff0004`, la cual es la dirección del registro de datos del teclado en miniMIPS. A este tipo de acceso se le denomina **E/S mapeada en memoria**, ya que el procesador trata a la **interfaz del controlador de un dispositivo** (registros del controlador) como una extensión de la memoria. Esta estructura es utilizada en la mayoría de los procesadores modernos.

La Figura 5 (b) muestra la organización del hardware de una computadora con E/S mapeada en memoria. Parte del espacio de memoria normal de la CPU está dedicado a los dispositivos de E/S, por lo que cada registro de E/S (puerto) tiene una dirección de memoria en el espacio direccionable del procesador. Además, las arquitecturas que utilizan E/S mapeada en memoria pueden utilizar las mismas instrucciones de acceso a memoria, por lo que no se necesitan instrucciones especiales (ni se utilizan código de operaciones específicos en el campo de operación de una instrucción). Como desventaja se suele mencionar que parte del espacio direccionable es utilizado, por lo que se reduce el tamaño de la memoria física que el sistema puede tener. Esto, aunque si bien es correcto, no es tan importante en las arquitecturas modernas con espacios direccionables de 32 y 64 bits.

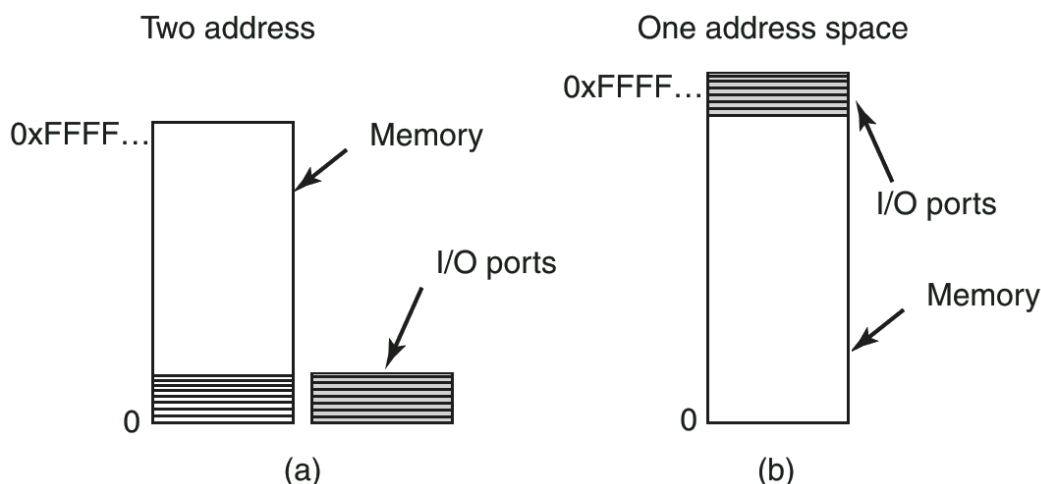


Figure 5: (a) E/S aislada. (b) E/S mapeada en memoria.

E/S aislada

Algunos microprocesadores tienen en cambio otro mecanismo, llamado comúnmente **E/S aislada (isolated I/O)**, las cuales utiliza instrucciones especiales para realizar entrada y salida. Por ejemplo en los procesadores intel x86 y en los antiguos procesadores z8000 el conjunto de instrucciones de cada arquitectura presentaban dos instrucciones especiales para acceder a los registros de cada dispositivo: `in` y `out`.

en x86 e inb y outb en el z8000. En estas arquitecturas, cuando el procesador ejecuta la instrucción `OUT 0xFF01` (ejemplo hipotético), el contenido del acumulador (o registro especial de la CPU) es colocado en el bus de datos. Al mismo tiempo, el número `0xFF01` es ubicado en los bits menos significativos del bus de direcciones, y un pulso (señal de control) es generado en la línea de escritura de E/S (I/O write). Cada interfaz de cada dispositivo en el sistema monitorea las líneas del bus de dirección. Cuando una interfaz de E/S observa su propia dirección junto con una señal de escritura o lectura (de registro), la interfaz actúa en consecuencia y realiza la ejecución de la transferencia de dato de E/S.

En la E/S aislada, la memoria principal y los dispositivos se encuentran en diferentes buses, por lo que el espacio direccionable por referencias a memoria e instrucciones de E/S son diferentes, aunque ellos pudieran reconocer la misma dirección numérica. Es decir, por ejemplo, la dirección `0xFF01` podría ser una dirección de memoria principal, y a la vez ser una dirección de un registro de un dispositivo. El tipo de instrucción es la que determina si es uno u otro. Debido a esto es que se suele denominar E/S aislada (espacio de direcciones diferentes para memoria y E/S).

5 Licencia, obras originales, permisos y bibliografía

Licencia de uso

Se permite copiar, distribuir y modificar este apunte; únicamente para fines académicos. Se permite copiar y distribuir copias modificadas con el mismo fin. Se solicita mantener la información de los autores de este apunte y de las obras originales.

Obras originales y permisos

Para ver el listado de obras originales y permisos escritos de los autores visite: <http://se.fi.uncoma.edu.ar/ayodc1/apuntes/permisos.txt>

Bibliografía

- Artículo

6 Indice

E/S aislada	6
E/S mapeada en memoria	6
bus del sistema	2
controlador	3
entrada y salida.....	1
periférico	3
puertos de E/S	4