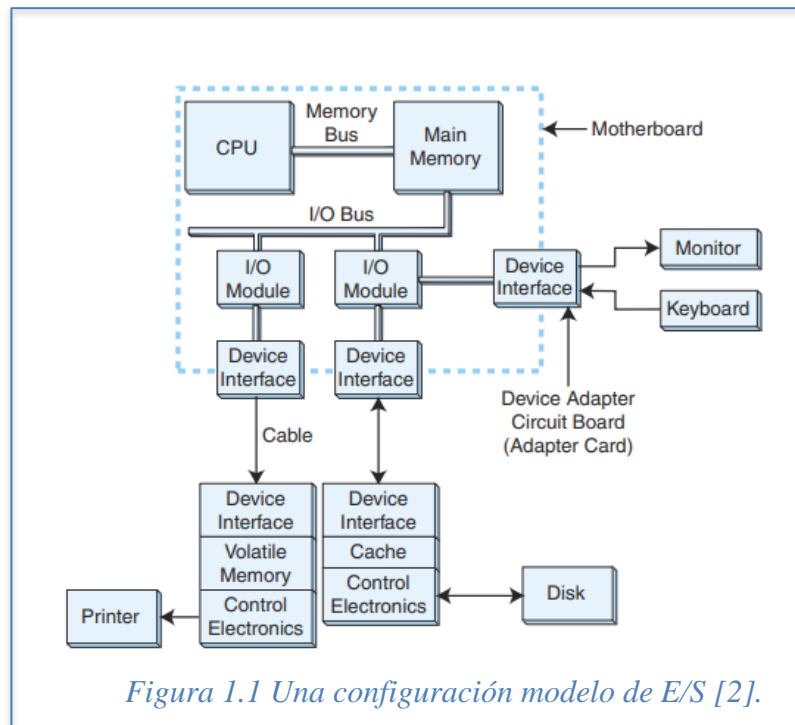


La entrada/salida (E/S) es la parte del computador que permite que la CPU hable con cosas externas: teclados, discos, pantallas, impresoras, redes, etc. Estas interacciones se resuelven mediante módulos o controladores que “traducen” entre el lenguaje interno del procesador (buses, direcciones, registros) y los detalles físicos del periférico (señales eléctricas, tiempos, formatos). En la práctica estos módulos ocultan la complejidad del dispositivo y presentan a la CPU operaciones sencillas de leer/escribir; eso facilita que programas y sistemas operativos trabajen sin conocer cada detalle eléctrico del periférico [1].



Los dispositivos externos varían mucho: unos son lentos (teclados, impresoras), otros muy rápidos (discos, tarjetas de red) y cada uno tiene su propia lógica de control y a menudo una memoria intermedia (buffer). Es decir, el periférico no suele conectarse directamente al bus del sistema; en su lugar va a través de una interfaz que maneja el posicionamiento, la temporización y errores, y que asegura que la información llegue completa y en el formato correcto. Esta heterogeneidad es la razón principal por la que existen los módulos de E/S [2].

Un módulo de E/S (o controlador) es básicamente una pequeña “caja” con registros de datos, registros de estado y lógica de control que se conecta al bus y al dispositivo. El procesador le manda órdenes por el bus de control, el módulo usa su registro de datos para intercambiar bytes y ofrece señales de estado para decir si está listo, ocupado o con error. Además, los módulos suelen tener buffers para compensar la diferencia de velocidad entre la CPU/memoria y el periférico, evitando que la memoria principal quede ocupada todo el tiempo [1].

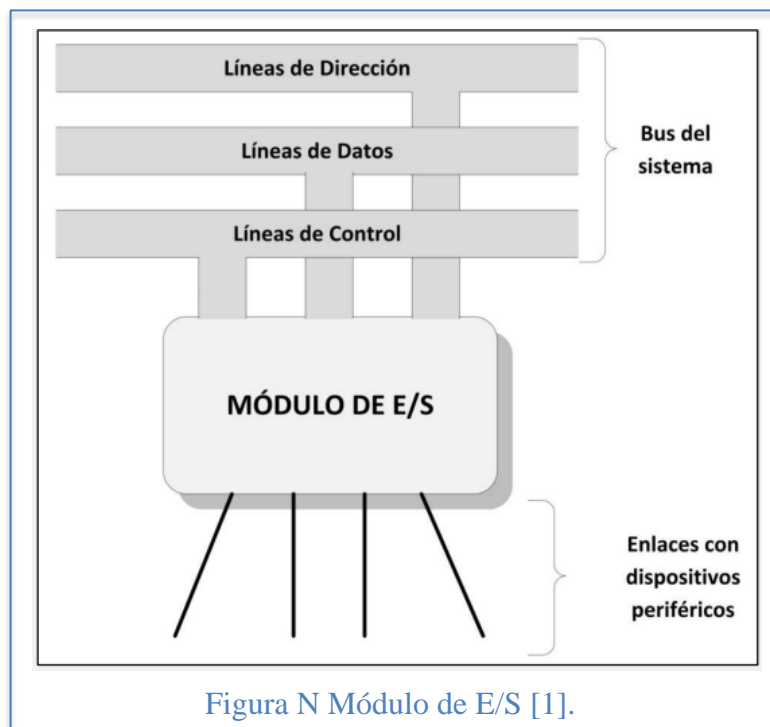


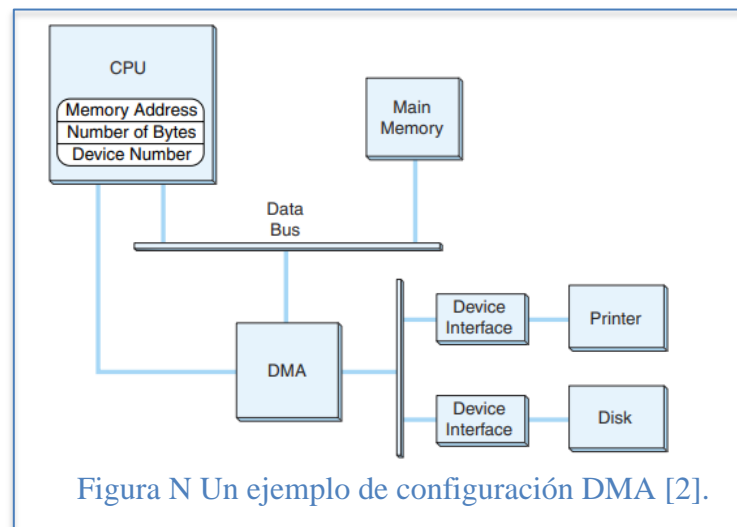
Figura N Módulo de E/S [1].

La E/S programada (o polled I/O) consiste en que la CPU controla todo: envía la orden y se queda “preguntando” al módulo si hay datos o si terminó la operación. Es fácil de implementar y de entender, pero ineficiente, porque la CPU entra en un bucle de espera y no hace trabajo útil mientras tanto. Por eso se usa cuando el sistema es muy simple o cuando el número de periféricos es pequeño y el coste de esperar es tolerable [2].

La E/S mediante interrupciones mejora esto: el procesador inicia la operación y sigue con otras tareas; cuando el módulo termina avisa con una interrupción y la CPU salta a una rutina de servicio para completar la transferencia. Este método necesita soporte de hardware y de software (vectores de interrupción, guardar/restaurar contexto), pero

reduce el tiempo de CPU desperdiciado y es mucho más práctico para sistemas multitarea [2].

El acceso directo a memoria (DMA) es la técnica que más libera a la CPU: un controlador DMA se encarga de mover bloques de memoria entre el periférico y la RAM sin que la CPU copie cada byte. La CPU solo configura la transferencia (dirección, tamaño) y el DMA “roba” ciclos de memoria para transferir los datos (cycle stealing) o toma el control del bus por ráfagas. Esto es ideal para discos o redes donde se manejan bloques grandes; en sistemas grandes incluso se usan procesadores de E/S (IOP) o canales que ejecutan programas de E/S independiente [2][3].



Por dentro, la interfaz interna de un módulo muestra claramente las líneas que todos conocemos: bus de datos (bidireccional), líneas de dirección, y líneas de control/estado. Un ejemplo clásico es el chip programable 82C55A (un ejemplo didáctico) que ofrece puertos de 8 bits, registros de control y señales que permiten configurar modos y sincronizar teclado/pantalla. En resumen: la interfaz interna coordina la selección del puerto, el formato de los datos y las señales de control/handshaking para que la comunicación sea fiable [1].

En la práctica conviene recordar lo siguiente (idea de estudiante): para tareas pequeñas o sistemas embebidos sencillos la E/S programada es más fácil de programar; para sistemas

interactivos y multitarea las interrupciones son la norma; y para transferencias grandes (discos, tarjetas de red) siempre que sea posible se usa DMA o canales para no “secuestrar” la CPU. Además, diseñar o seleccionar un módulo de E/S implica pensar en buffers, priorización de interrupciones y en cómo se mapean las direcciones de E/S en el sistema [3].

- [1] Daniel M. Argüello, Santiago C. Pérez, and Higinio A. Facchini, “Arquitectura de Computadoras,” 2022.
- [2] R. M. . Merrill, *Statistical methods in epidemiologic research*. Jones & Bartlett Learning, 2016.
- [3] S. Harris, “Digital Design and Computer Architecture,” 2007.