

Maquina elemental

Conceptos preliminares

Registro: Memoria muy rápida que almacena una cierta cantidad d bits

Compuerta: circuitos electrónicos bi-estables unidireccionales (solo permiten dos estados abierto/cerrado)/ Son la base del hardware para las computadores. Para intercambiar información entre dos registros de la misma deben viajar por el bus, y esto se lo logra a través de las compuertas. Cuando las mismas se abren la información está disponible en el bus, cuando se cierra desaparece. No pueden existir dos compuertas abiertas al mismo tiempo.

Arquitectura de Von Neumann

Es del año 1845. Promueve la mecanización del tratamiento digital de la información. Tiene dos conceptos fundamentales:

Principio del programa almacenado: El computador debe tener el programa almacenado en su propia memoria. El uso de la memoria es para el almacenamiento de datos y la instrucción del programa.

Principio de ruptura de secuencia: Las operación de decisión lógica deben ser automáticas, dotando a la máquina de una instrucción llamada “Salto condicional”.

La computadora con arquitectura de Von Neumann tiene cinco partes:

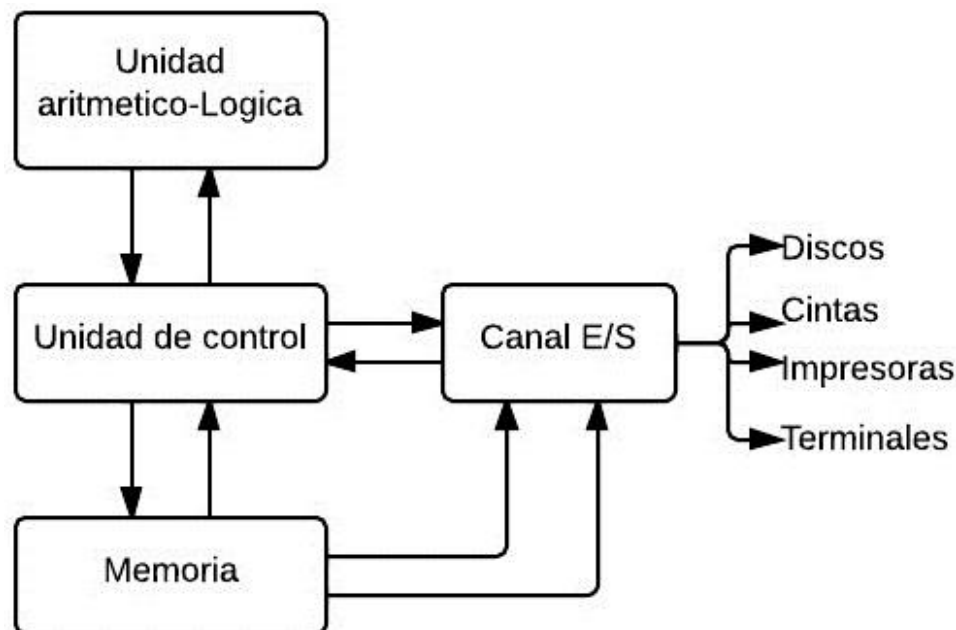
Memoria: Está dividida en celdas cuyo contenido es variable y son identificadas con un número fijo llamado “dirección de memoria”. La capacidad total de la memoria está dada por la cantidad de celdas disponibles. Las celdas tienen datos e instrucciones.

UAL: Unidad encargada de realizar operaciones elementales de tipo aritmético y lógicas.

Dispositivo de E/S: Gestiona la transferencia de información entre los periféricos y la memoria central.

Bus de datos: Sistema digital que transporta datos entre las distintas partes (no la almacena, solo la transmite).

- Bus de datos: Mueve información por los componentes de hardware.
- Bus de direcciones: Ubica los datos en la memoria.
- Bus de control: Marca el estado de una instrucción.



Abacus

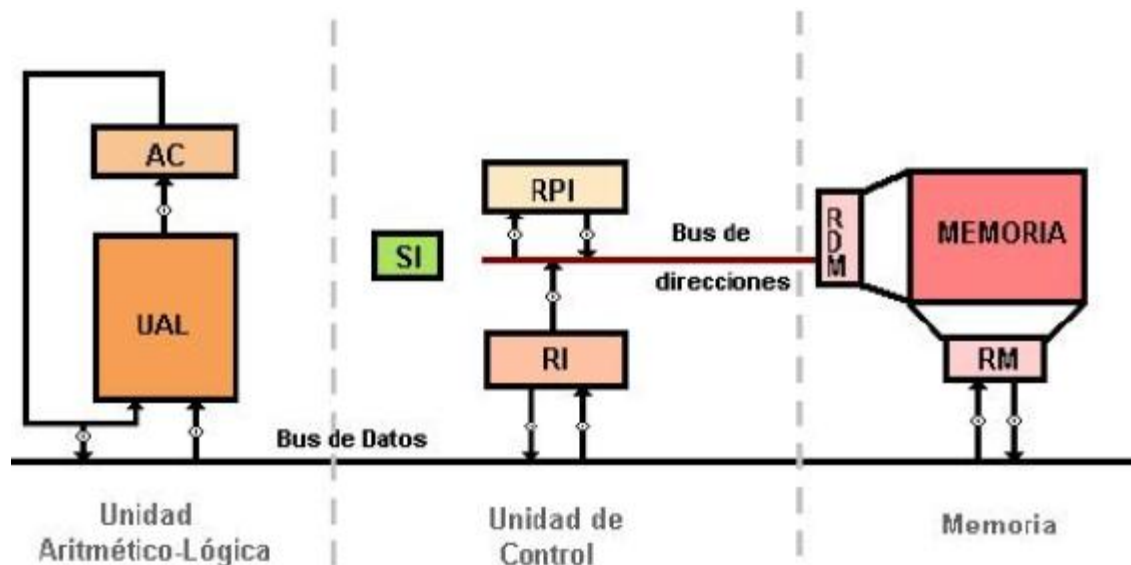
Unidad aritmética-lógica (UAL): Opera con los datos recibidos, según las ordenes de la unidad de control.

Unidad de control (UC): extrae de la memoria la instrucción a ejecutar, analiza dicha instrucción y llama la UAL. Extrae de la memoria, los datos implicados por la instrucción, desencadena el tratamiento de dichos datos en la UAL y almacena el resultado en la memoria.

Unidad central de proceso: UC + UAL

Memoria: conjunto de celdas capaces de almacenar información (datos o instrucciones). Las celdas tienen una dirección, para poder acceder a su contenido.

Secuenciador de instrucciones (SI): analiza el código de operación, distribuye las órdenes al conjunto de unidades del ordenador (UC, UAL, etc.) para hacerles ejecutar las distintas fases de la instrucción.



- Es una máquina de una sola dirección.
- El acumulador es un registro particular que alberga el primer operando y el resultado de las operaciones. Todas las operaciones se realizan “contra” el acumulador. El acumulador realiza las operaciones aritméticas, lógicas y de comparación.
- El ciclo de memoria equivale a dos impulsos de reloj.
- La UC contiene tres registros:
 - RPI: Contiene la dirección de la próxima instrucción a ejecutar.
 - RI: Contiene la instrucción extraída de la memoria.
 - SI: Administra la apertura y cierre de compuertas.
- La memoria tiene dos registros:
 - RDM: Contiene la dirección de la celda de memoria.
 - RM: Contiene el dato de la celda de memoria.
- Tamaño RPI = Tamaño RDM = Tamaño OP = cantidad de celdas direccionables.
- Tamaño AC = Tamaño RI= Tamaño RM = Longitud de instrucción = Longitud de celda.
- Maneja una aritmética de Binario Punto Fijo consigo de 16 bits.

Fases de una instrucción

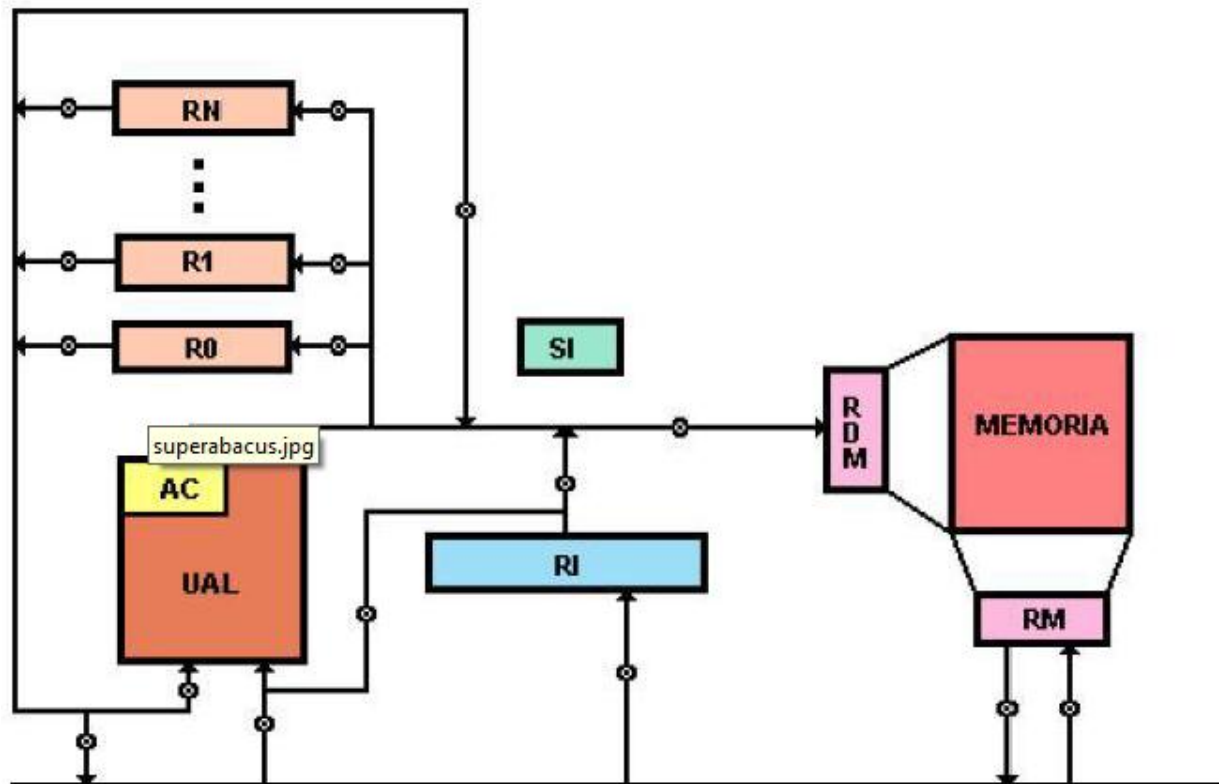
Fase de búsqueda: Consiste en localizar la instrucción que se va a ejecutar, y analizarla. Siempre igual

$RDM < (RPI)$	
$RM < ((RDM))$	
$RI < RM$	
$RPI < (RPI)+1$	

Fase de ejecución: Implica el movimiento de datos. Los pasos se realizan en forma secuencial, la UCP sigue las sedadas por el reloj del sistema. Este paso es distinto para cada operación.

Suma	Carga	Almacenamiento	SALTO
$RDM < (OP)$	$RDM < (OP)$	$RDM < (OP)$	$RPI < (OP)$
$RM < ((RDM))$	$RM < ((RDM))$	$RM < AC$	
$AC < AC + (RM)$	$AC < (RPI)+1$	$(RDM) < (RM)$	

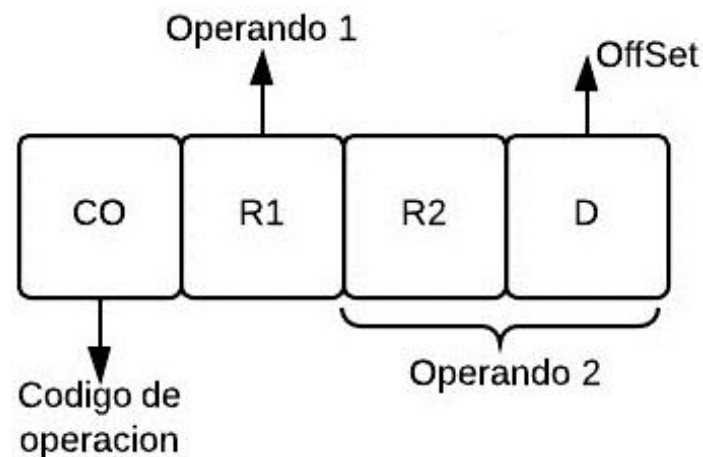
Superabacus



- Es una máquina de tercera generación ya que posee un conjunto de registros banalizados, es decir utilizables tanto como registros aritméticos como registros de índice.
- Es una máquina de dos direcciones (Primer y segundo operando).
- Posee un conjunto de registros generales, utilizables para aritmética o direccionamiento (como registros índice).
- El ciclo de memoria equivale a cuatro impulsos de reloj.
- No tiene RPI, se le asigna esa función al registro 0.
- La UAL se utiliza para calcular direcciones y operar con los datos.
- El ciclo de memoria equivale a cuatro impulsos de reloj, lo que autoriza entre otras cosas a efectuar una modificación de dirección (sumar el contenido de un registro con la dirección indicada en la instrucción) sin retrasar por ello el inicio del ciclo de procesamiento.
- El RI alberga instrucciones de dos operandos.

El R1 y el R2 representan dos registros diferentes. El segundo operando puede contener la dirección de una celda de memoria, se obtiene sumando el OffSet (desplazamiento) al contenido del registro indicado.

$A(\text{celda}) = (R2) + D$.



Multi-programación

Mantenimiento concurrente de dos o más tareas en ejecución (en memoria). Elimina o reduce el tiempo ocioso de la CPU.

- 1) Sin interrupción: un programa posee el control hasta que se termina, o mientras espera una operación de E/S.
- 2) Con interrupción: cuando finaliza una operación de E/S, busca al programa de mayor prioridad (sin E/S pendientes) y le cede el control.
- 3) Tiempo compartido: asigna igual cantidad de tiempo de procesador a cada tarea (o a cada usuario).

Problemas asociados a la multiprogramación:

- Problemas de asignación de memoria (se resuelven utilizando memoria virtual)
- Protección de los programas y los datos en memoria.
- Preservación del estado de la máquina, en los cambios de programa.

Multi-procesamiento: se procesan varias instrucciones simultáneamente, utilizando más de un procesador.

(Procesamiento paralelo)

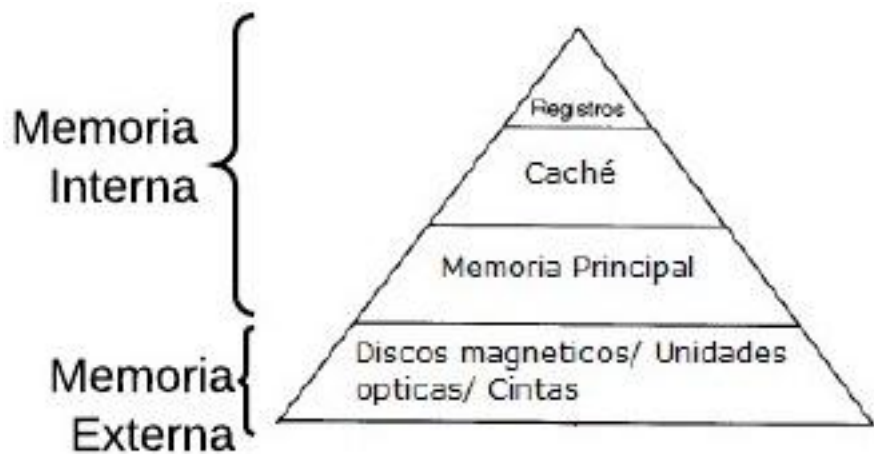
Direccionamiento

Direccionamiento: "función" que recibe un campo operando de una instrucción y devuelve la dirección real del dato en memoria.

- Directo y absoluto: en el campo operando se especifica la dirección real del dato. Los programas no pueden ser trasladados a otra parte de la memoria, sin cambiar todos los operandos. Los operandos deben ser grandes, para poder acceder a toda la memoria. Abacus, IBM, intel.
- Inmediato: en el campo operando se especifica el dato. Abacus, IBM, intel.
- Indirecto: el campo operando contiene la dirección en memoria de una celda X. Esa celda X contiene la dirección Y real del dato en memoria. IBM, intel
- Relativo: la dirección del dato se obtiene sumando una constante al contenido del campo operando. IBM, intel
 - o Por base y desplazamiento: se especifica un registro base (que contiene una dirección) y un desplazamiento desde esa dirección (la constante). IBM
 - o Por base, índice y desplazamiento: se especifica un registro base, un registro índice y el desplazamiento. IBM
 - o Por referencia al programa: la dirección del dato se obtiene sumando el contenido del RPI a un desplazamiento indicado.
 - o Por yuxtaposición (direccionamiento paginado): la memoria se encuentra dividida en páginas (bloques de longitud. Para obtener las direcciones necesitamos:
 - Indicador de página (IP): en un registro específico o de propósito general de la máquina.
 - Dirección de la palabra dentro de la página (DP): en el campo de la instrucción.

Así, concatenando ambas partes obtenemos la dirección completa.

Componentes de un computador Memoria



Mientras más alto vamos en la pirámide la velocidad de acceso aumenta, con ella el costo. En cambio mientras descendemos lo que aumenta es la capacidad de almacenamiento.

Características:

Ubicación, donde esta físicamente la memoria (Interna/Externa).

Capacidad, se mide en:

- Bytes
- Palabras(Especialmente la RAM)

Unidad de transferencia, Palabras (especialmente la RAM) y bloques (para memoria externa).

Método de acceso:

- Secuencial (cintas):
 - Unidades de datos: registros.
 - Acceso lineal en secuencia.
 - Se debe pasar por todos los registros intermedios antes de acceder al registro deseado.
 - Tiempo de acceso variable.
- Directo (discos magnéticos):
 - Dirección única para bloques o registros basada en su posición física.
 - Tiempo de acceso variable.
- Aleatorio(memoria principal, algunas memorias caché):
 - Cada posición direccionable de memoria tiene un mecanismo de direccionamiento cableado físicamente.
 - Tiempo de acceso constante, independiente de la secuencia de accesos anteriores.
- Asociativo (memoria caché):
 - Tipo de acceso aleatorio por comparación de patrón de bits.
 - La palabra se busca por una porción de su contenido en vez de por su dirección.
 - Cada posición de memoria tiene un mecanismo de direccionamiento propio.
 - Tiempo de acceso constante, independiente de las secuencias de accesos anteriores o su ubicación.

Performance:

- Tiempo de acceso:
 - Acceso aleatorio: Tiempo necesario para hacer una operación de lectura o escritura.
 - Sin acceso aleatorio: Tiempo necesario para posicionar el mecanismo de lectura/escritura en la posición deseada.
- Tiempo de ciclo

- Acceso aleatorio: Tiempo de acceso más el tiempo adicional necesario para que una nueva operación pueda comenzar.
- Tasa de transferencia.
 - Tasa con la cual los datos son transferidos dentro o fuera de la unidad de memoria.
 - Acceso aleatorio: $1/\text{tiempo de ciclo de memoria}$.
 - Sin acceso aleatorio: $T_n = T_a + n/R$
 R: Tasa de transferencia (bits/segundos), T_a : Tiempo de acceso promedio.
 T_n : Tiempo promedio para leer/escribir n bits, N: número de bits.

Tipo físico, semi-conductores (RAM, caché), ópticos, magnéticos.

Características físicas:

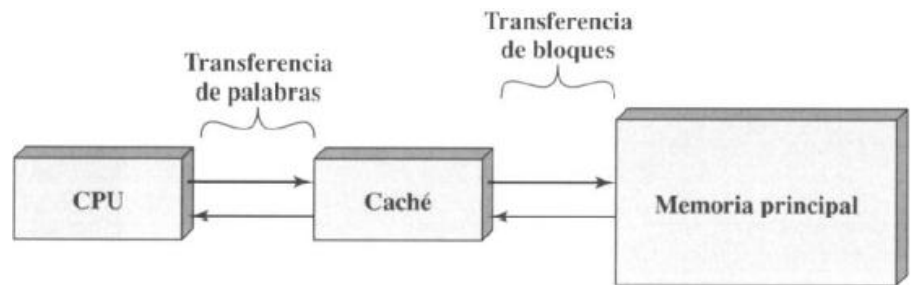
- Volátiles (RAM, caché)/No volátiles (Discos, cintas, etc).
- Borrables (casi todo)/No borrables (ROM).

Organización, como están confeccionados los módulos de la memoria RAM

Memoria caché

El objetivo de la memoria caché es lograr que la velocidad de la memoria sea lo más rápida posible, consiguiendo al mismo tiempo un tamaño grande al precio de memorias semiconductoras menos costosas. Hay memoria principal relativamente grande y más lenta, junto con una memoria caché más pequeña y rápida. La caché contiene una copia de partes de memoria

principal. Cuando el procesador intenta leer una palabra de memoria, se hace una comprobación para determinar si la palabra está en el caché. Si es así, se entrega dicha palabra al procesador. Si no, un bloque de memoria principal, consiste en un cierto número de palabras, se transfiere a la caché y después la palabra es entregada al procesador. Debido al fenómeno de localidad de las referencias, cuando un bloque de datos es capturado por el caché para satisfacer una referencia a memoria simple, es probable que se hagan referencias futuras a la misma posición de memoria o a otras palabras del mismo bloque.



Memorias caché y principal.

El procesador genera una dirección, RA, de una palabra a leer. Si la palabra está en el caché, es entregada al procesador. Sino, el bloque que contiene dicha palabra se carga en la cache, y la palabra después es llevada al procesador.

Numero de caché

Cuando se introdujeron originalmente las cachés, un sistema tenia normalmente solo una caché. Más recientemente, se ha convertido en una norma de uso de múltiples caches. Hay dos aspectos de diseño relacionados con este tema que son el número de niveles de caché y el uso de cache unificada frente al de caché separadas.

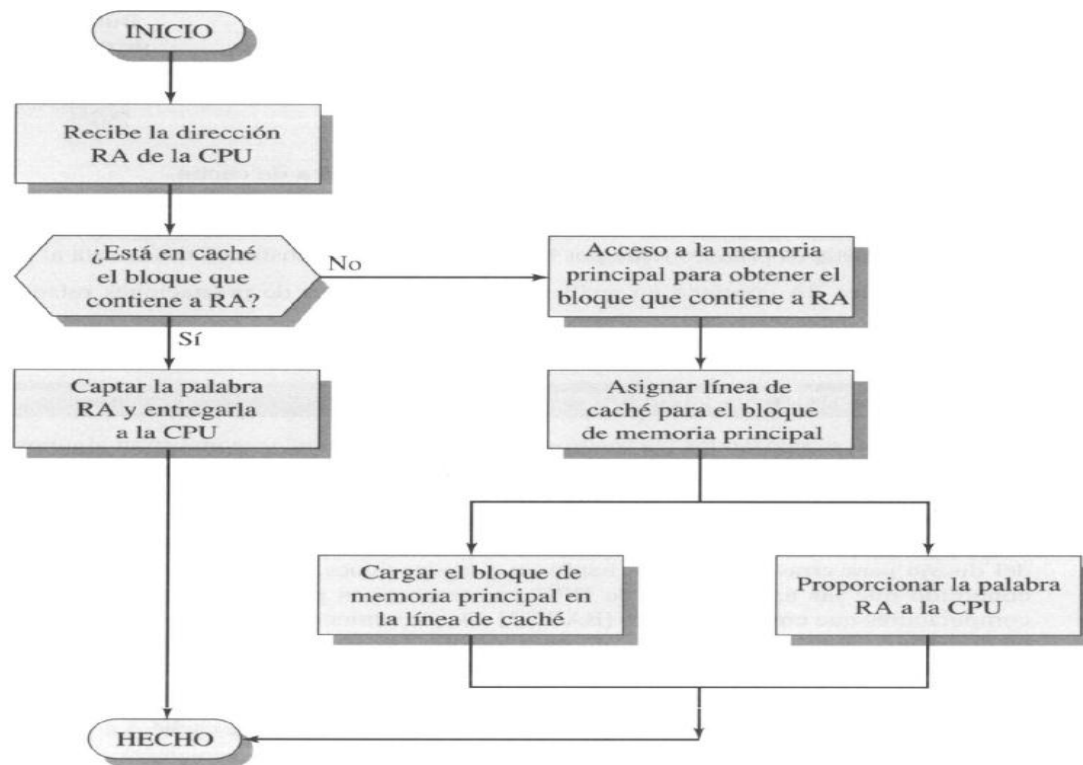
Cachés multinivel

Con el aumento de densidad de integración, ha sido posible tener una caché en el mismo chip del procesador: caché on-chip. Comparada con la accesible a tras de un bus externo, la caché on-chip reduce la actividad del bus externo del procesador y por tanto reduce los tiempos de ejecución e incrementa las prestaciones globales del sistema. Cuando la instrucción o el dato requeridos se encuentran en la caché on-chip, se elimina el acceso al bus.

La inclusión d una caché on-chip deja abierta la cuestión de si es además deseable una caché externa y off-chip. Normalmente la respuesta es afirmativa, y los diseños más actuales incluyen tanto caché on-chip como externa. La estructura más sencilla de este tipo se denomina cache de dos niveles, siendo la caché interna el nivel a (L1) y externa el nivel 2 (L2). La razón por la que se incluye una caché L2 es la siguiente. Si no hay cache L2 y el procesador hace una petición de acceso a una posición de memoria que no está en la caché L1, entonces el procesador debe acceder a la DRAM o a la ROM a través del bus. Debido a la lentitud usual del bus y a los tiempos de acceso de las memorias, se obtienen bajas prestaciones. Por otra parte, si se utiliza una caché L2 SRAM (RAM estática), entonces con frecuencia la información que falta puede recuperarse fácilmente. Si la SRAM es suficientemente rápida para adecuarse a la velocidad del bus, los datos pueden accederse con cero estados de espera, el tipo más rápido de transferencia de bus.

En la actualidad son destacables dos características de diseño de las cachés multinivel. En primer lugar, para el caso de una caché L2 externa, muchos diseños no usan el bus del sistema como camino para la transferencia entre la caché L2 y el procesador, sino que se emplea un camino de datos aparte para reducir el tráfico en el bus del sistema. En segundo lugar, gracias a la continua reducción de dimensiones de los componentes de los procesadores, es fácil encontrar procesadores que incorporan la caché L2 en el propio chip, con la consiguiente mejora de prestaciones.

La mejora potencial del uso de una caché L2 depende de las tasa de aciertos en ambas caches, L1 y L2. Varios estudios han demostrado que, en general, el uso de un segundo nivel de caché mejora las prestaciones.



No obstante, el uso de cachés multinivel complica todos los aspectos de diseño de la caché, incluyendo el tamaño, el algoritmo de sustitución y la política de escritura.

Con la creciente disponibilidad de superficie para caché en el propio chip, en la mayoría de los microprocesadores modernos se ha llevado la caché L2 al chip de procesador, y se añade una caché L3. Inicialmente, la caché L3 era accesible a través del bus externo, pero más recientemente los microprocesadores han incorporado una L3 on-chip. En cualquiera de los casos, añadir un tercer nivel de caché parece suponer mejoras de las prestaciones.

Administración de memoria

Asignación contigua simple

A cada tarea se le asigna la totalidad de la memoria, aunque en realidad ella requiera solo una mínima parte de ese total. La tarea tiene control absoluto de la CPU, hasta su finalización o hasta que ocurra un error.

En la memoria está el sistema operativo y el programa del usuario. El resto, es espacio desperdiciado.

Desventajas:

- Desaprovechamiento de recursos, particularmente en los sistemas en los que el operador debe realizar tareas manuales (carga de cintas o lotes de tarjetas).
- La CPU permanece ociosa mientras los canales de E/S cumplen sus funciones.

Ventajas:

- Es simple, minimiza dificultades de programación.

Asignación particionada

Se hace una partición de modo tal, que cada tarea se aloja en una sección de la memoria. Las particiones pueden tener el mismo tamaño o no. El tamaño es fijo. El programador debe saber en qué partición correrá el programa y cuál es el tamaño de la misma.

La CPU se reparte entre las tareas a modo de fragmentos de tiempo, y en cada uno el sistema operativo hace que el procesador ejecute uno de los programas, según cierta prioridad, en el espacio de memoria que se ha asignado a este (su partición), hasta que se de alguna de las siguientes condiciones:

- 1) Terminación del trabajo
- 2) Detección de un error
- 3) Requerimiento de una operación de E/S
- 4) Terminación de los fragmentos de tiempo

En cualquiera de estos casos, el procesador pasa a ocuparse de la tarea de mayor prioridad entre las restantes. Específicamente, en los dos primeros casos, la tarea es eliminada de la memoria, mientras que en los otros dos restantes, la suspensión es temporaria y puede ser reanudada en el momento que determine el sistema operativo. Es posible quitar las particiones en un momento determinado, sacando los programas que estén en ejecución. Las E/S determinan que proceso se ejecuta. Se ejecuta según la prioridad y si el programa está listo para ejecutarse.

Este sistema es más costoso que el de asignación contigua simple, porque se requiere:

- Mecanismos de protección especiales, para evitar que cualquier tarea interfiera accidentalmente con otra.
- Se necesita más memoria, porque el sistema operativo es más complejo.

Desventajas:

- Cuando se termina la ejecución de los programas chicos, no hay lugar para poner a los mas grandes, porque las particiones son fijas. Se produce **fragmentación** (este problema se resuelve mediante la asignación particionada reasignable y la paginación)

Ventajas:

- Como se basa en la multiprogramación, ofrece los beneficios de esta.

Asignación particionada reasignable

Consiste en la reasignación de las particiones, “compactando” las tareas que están en memoria, eliminando la fragmentación. Requiere ciertos recursos de hardware.

Al reubicar los programas, hay que reubicar a los componentes dependientes de direcciones de memoria, tales como: instrucciones con referencia a la memoria, listas de parámetros y estructuras de datos.

Para solucionar este problema, existe un registro llamado registro de reasignación (o reubicación), cuyo contenido se suma automáticamente a toda dirección utilizada para referirse a la memoria. Es un registro índice adicional “oculto”.

Desventajas:

- Al terminarse una tarea, el sistema debe reubicar todas las demás para compactarlas.
- A pesar de todo, hay todavía desaprovechamiento de memoria.
- Cada vez que se accede a la memoria, hay que realizar el cálculo de la dirección.

Ventaja:

- Esto hace al programa independiente de su ubicación física en la memoria.

Asignación particionada

Se divide el programa en páginas de 1000 bytes y a la memoria en bloques de 1000 bytes. Luego se cargan las páginas del programa en los bloques y se utiliza un mapa de páginas (pueden ser registros físicos o secciones reservadas de la memoria), para indicar en que bloque de la memoria se encuentra cada página del programa.

Si las páginas son muy grandes, se vuelve al método de partición reasignable, mientras que si las páginas son muy chicas, se necesitarán muchos registros de mapas de páginas, lo que incrementa el costo del sistema.

En la práctica, en muchos sistemas se emplean este método de asignación (en 1972!!), el tamaño de las páginas es de alrededor 4000 bytes.

Desventajas:

- El costo asociado con cada transformación de dirección.
- Hay desperdicios de espacio en la última página (ruptura de página). En promedio, por cada tarea se desperdicia media página.
- Este método requiere que todas las páginas del programa (todo el programa) estén en memoria antes de poder ejecutarse.
- Requiere un equipamiento físico cuantioso

Ventajas:

- Este sistema minimiza la fragmentación y evita tener que reordenar (compactar) la memoria, ya que las páginas de cada programa no necesitan estar en forma contigua.

Asignación paginada por demanda (Memoria virtual/Swapping)

La memoria virtual puede en principio ser mayor que la memoria física disponible. Se utiliza una tabla de páginas, que tiene varios campos: número de página, número de bloque (la posición en la memoria), estado (si la página está en memoria o no) y algunos elementos de juicio.

Un algoritmo de asignación ubica inicialmente algunas páginas del programa en la memoria. Por cada requerimiento de acceso a una página se consulta la tabla para saber en qué bloque está. Si la página no está en memoria (según el campo estado), se “demanda” la carga de esa página en memoria. Cuando se detecta que una página no está en memoria, se detiene el procesamiento y se genera automáticamente una interrupción; esta interrupción se conoce como interrupción de página (page fault). Es entonces el sistema operativo el que ejecuta la función de acceder al almacenamiento secundario para adquirir la página pedida y cargar la página, reemplazando con ella alguna de las páginas ya cargadas. La decisión sobre cuál ha de ser la página reemplazada es cuestión que se resuelve con ayuda del campo de juicio. Lo ideal es reemplazar la página que se usa menos frecuentemente. Hay varios algoritmos que permiten decidir que páginas han de conservarse en memoria y cuáles no. Uno de ellos consiste en llevar la cuenta del número de accesos a cada página en el campo de juicio, y reemplazar la página con menor número de accesos.

Desventajas:

- Si la CPU se ocupa de demasiados programas al mismo tiempo, como puede ocurrir cuando se usa paginación por demanda en un sistema con insuficiente memoria física (o algoritmos de juicio deficientes), no es raro que se carguen páginas en la memoria, se las suplante enseguida y luego se las vuelva a cargar y así sucesivamente, con grave deterioro de la producción; en ese caso se dice que el sistema está “trillando” (trashing) programas.
- Muchos accesos a disco (trade off) por cada page fault.
- Muchos accesos a mapa de páginas.

Ventajas:

- Permite ejecutar programas que no entran completamente en la memoria.

Funciones de un módulo de entrada salida

Las principales funciones y requisitos de un módulo de E/S se encuentran dentro de las siguientes categorías:

1. Control y temporización.
2. Comunicación con el procesador.
3. Comunicación con los dispositivos.
4. Almacenamiento temporal de datos.
5. Detección de errores

1. Control y temporización:

Coordina el tráfico entre los recursos internos y los dispositivos externos.

2. Comunicación con el procesador:

Decodificación de órdenes:

El módulo de E/S acepta órdenes del procesador. Estas órdenes generalmente se envían utilizando líneas del bus de control

Datos:

El procesador y el módulo de E/S intercambian datos a través del bus de datos.

Información de Estado:

Puesto que los periféricos son lentos, es importante conocer el estado del módulo de E/S. También puede haber señales para informar ciertas situaciones de error.

Conocimiento de Dirección:

Igual que cada palabra de memoria tiene una dirección, cada dispositivo de E/S tiene otra. Así, un módulo de E/S puede reconocer una única dirección para cada uno de los periféricos que controla.

2. Comunicación con el dispositivo

Esta comunicación implica intercambiar órdenes, información de estado y datos.

3. Almacenamiento temporal de datos

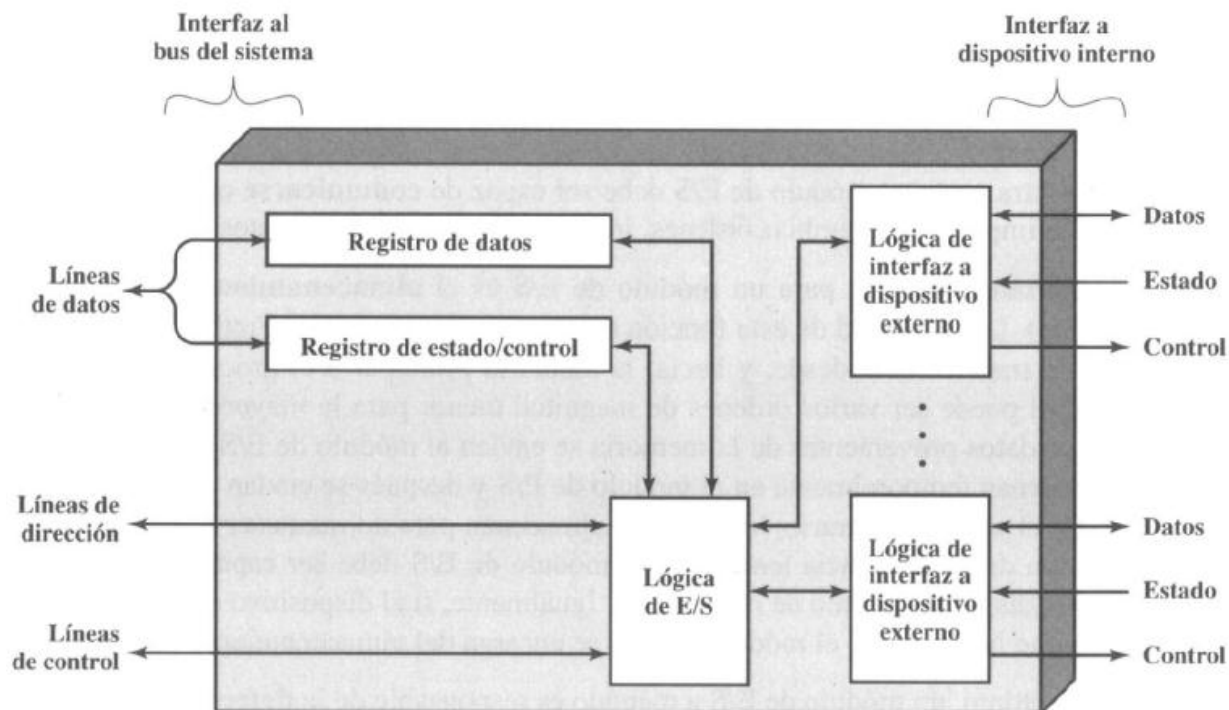
Esta es una tarea esencial para el módulo de E/S. Mientras la velocidad de transferencia desde, y hacia, la memoria principal o el procesador es bastante alta, dicha velocidad puede ser varios órdenes de magnitud menor para la mayoría de los dispositivos periféricos. Los datos provenientes de la memoria se envían al módulo de E/S en ráfagas rápidas. Los datos se almacenan temporalmente en el módulo de E/S y después se envían al periférico a la velocidad de este.

4. Detección de errores

El módulo de E/S es responsable de detectar errores e informar al procesador. Una clase de errores son los defectos mecánicos y eléctricos en funcionamiento del dispositivo. Otra clase está constituida por los cambios accidentales en los bits al transmitirse desde el dispositivo al módulo de E/S.

Estructura de un módulo de E/S

Los datos que se transfieren a, y desde, el módulo se almacenan temporalmente en uno o más registros de datos. Además, puede haber registros de estado dan información del estado presente. Este también puede funcionar como un registro de control, para recibir información del procesador. La lógica que hay en el módulo interactúa con el procesador a través de una serie de líneas de control. Estas son las que utiliza el procesador para proporcionar las ordenes al módulo de E/S. Algunas de las líneas de control pueden ser utilizadas por el módulo de E/S. El módulo también debe ser capaz de reconocer y generar las dirección asociadas a los dispositivos que controla. Cada módulo de E/S tiene una dirección única. Por último, el módulo de E/S posee la lógica específica para la interfaz de cada uno de los dispositivos que controla



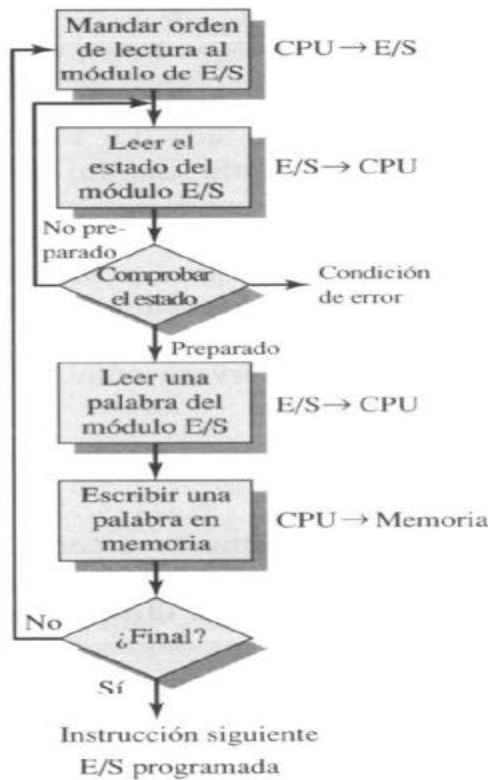
El funcionamiento de un módulo de E/S permite que el procesador vea una amplia gama de dispositivos de una forma simplificada, el módulo debe ocultar los detalles de temporización y electromecánica de dispositivos externos para que el procesador puede funcionar únicamente en termino de órdenes de lectura y escritura.

El módulo de E/S presenta al procesador una interfaz de alto nivel, que se denomina “**canal de E/S**”.

Técnicas para el manejo de E/S

Son posibles tres técnicas para operaciones de E/S. 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 las operaciones de E/S, incluyendo la comprobación del estado del dispositivo, el envío de una orden de lectura o escritura y la trasferencia 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 se desperdicia tiempo. Con la E/S mediante **interrupciones**, continúan ejecutándose otras instrucciones y es interrumpido por el módulo de E/S cuando esta 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 de una entrada. La alternativa se conoce como **acceso directo a memoria (DMA)**. En este caso, el módulo de E/S y la memoria principal intercambian datos directamente, sin intervención del procesador

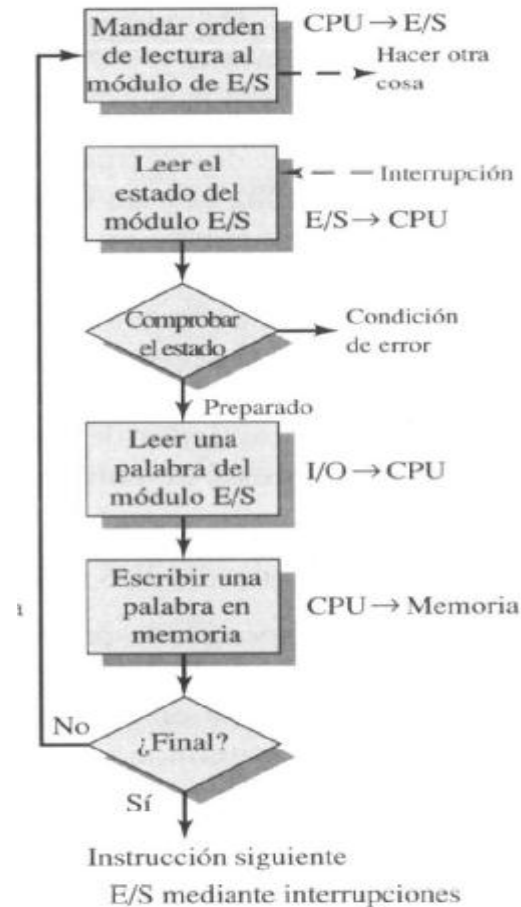


E/S Programada:

Cuando el procesador está ejecutando un programa y encuentra una instrucción relacionada con una E/S, ejecuta dicha instrucción mandando una orden al módulo de E/S apropiado. El módulo de E/S realizara la acción solicitada y después activara los bits apropiados en el registro de estado de E/S. El módulo de E/S no realiza ninguna otra acción para avisar al procesador. En concreto, no interrumpe al procesador. De esta forma, el procesador es responsable de comprobar periódicamente el estado del módulo de E/S hasta que encuentra que la operación ha terminado.

E/S mediante interrupciones:

El problema con la E/S programada es que el procesador tiene que esperar un tiempo considerable a que el módulo de E/S esté preparado para recibir o trasmitir los datos. El procesador, mientras espera, debe comprobar repetidamente el estado del módulo de E/S. Como consecuencia, se degrada el nivel de prestaciones de todo



el sistema.

Una alternativa consiste en que el procesador, tras enviar una orden de E/S a un módulo, continúe realizando algún trabajo útil. Después, el módulo de E/S interrumpirá al procesador para solicitar su servicio cuando esté preparado para intercambiar datos con él. El procesador ejecuta entonces la transferencia de datos, como antes, y después continúa con el procesamiento previo.

Problemas de los anteriores accesos de E/S:

La E/S con interrupciones, aunque más eficiente que la sencilla E/S programada, también requiere intervención activa dl procesador para transferir datos entre la memoria y el módulo de E/S, y cualquier transferencia de datos debe seguir un camino a través del procesador. Por tanto, ambas formas de E/S presentan dos inconvenientes:

- La velocidad de transferencia de E/S está limitada por la velocidad a la cual el procesador puede comprobar y dar servicio a un dispositivo.
- El procesador debe dedicarse a la gestión de las transferencias de E/S; debe ejecutar cierto número de interrupciones por cada transferencia de E/S

Acceso directo a memoria:

El DMA requiere un módulo adicional en el bus del sistema. El módulo o controlador de DMA es capaz de imitar al procesador y, de hecho, de recibir el control del sistema cedido por el procesador. Necesita dicho control para transferir datos a, y desde, memoria a través del bus del sistema. Para hacerlo, el módulo DMA debe utilizar solo 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”, puesto que, en efecto, el módulo DMA roba un ciclo de bus. Cuando el procesador desea leer o escribir un bloque de datos, envía una orden al módulo DMA, incluyendo la siguiente información:

- Si se solicita una lectura o una escritura, utilizando la línea de control de lectura o escritura entre el procesador y el módulo de DMA.
- La dirección del dispositivo de E/S en cuestión.
- La posición inicial de memoria a partir de donde se lee o se escribe.
- El número de palabras a leer o escribir.

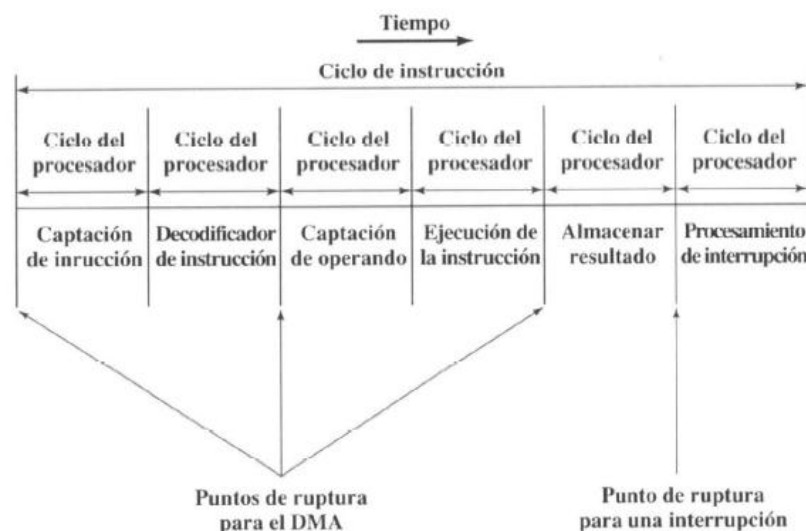
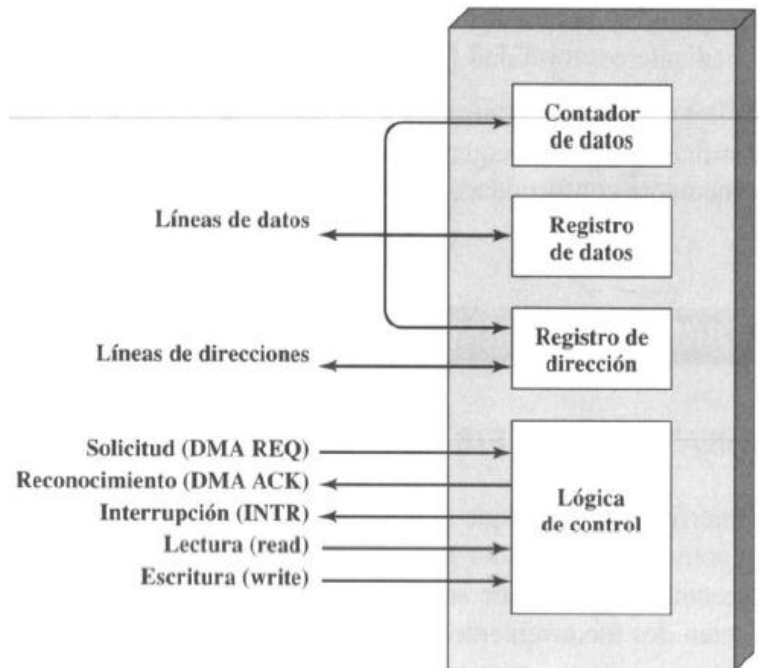
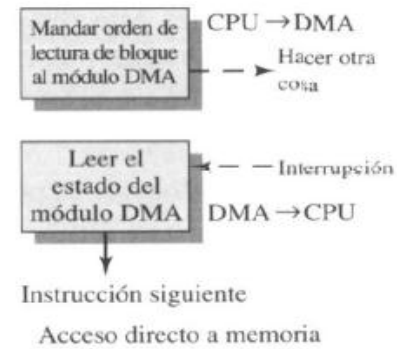
Después, el procesador continúa con otro trabajo. Ha delegado la operación de E/S al módulo de DMA, que se encargara de ella. El módulo de DMA transfiere el bloque completo de datos, palabra a palabra, directamente desde o hacia la memoria, sin que tenga que pasar a través del procesador. Y de esta manera el procesador solo interfiere al principio y al final de la transferencia.

En cada caso, el procesador se detiene justo antes de necesitar el bus. Después, el módulo DMA transfiere una palabra y devuelve el control al procesador.

Obsérvese que no se trata de una interrupción; el procesador no guarda el contexto ni hace nada más.

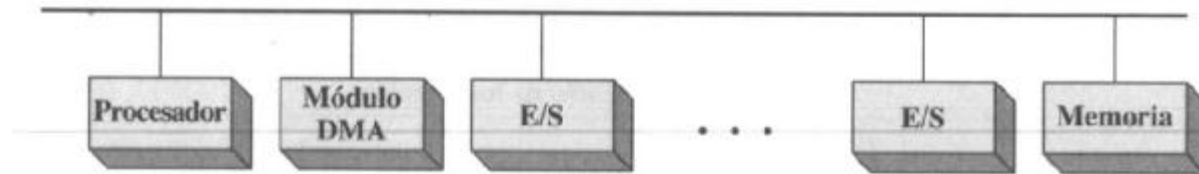
En cambio, el procesador espera durante un ciclo de bus.

El efecto resultante es que el procesador es más lento efectuando los programas. No obstante, para una transferencia de E/S de varias palabras, el DMA es mucho más eficiente que la E/S mediante interrupciones o la E/S programada.

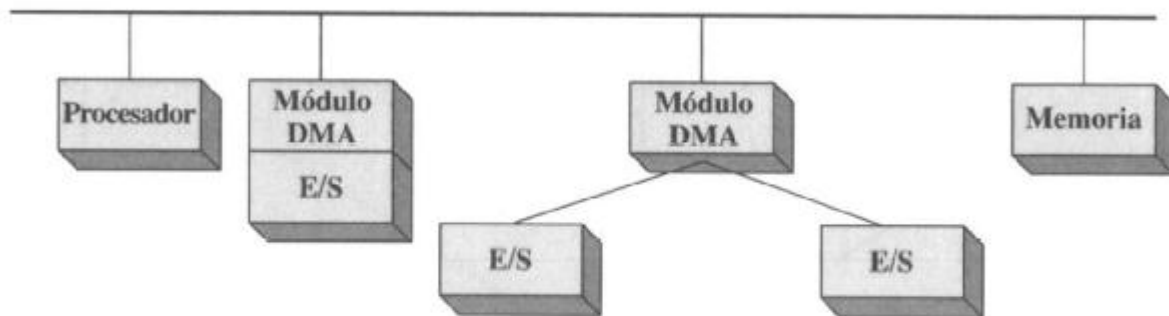


El número de ciclos de bus necesarios puede reducirse sustancialmente si se integran las funciones de DMA y de E/S (B). Esto significa que existe un camino entre el módulo de DMA y uno o más módulos de E/S que no incluye al bus de sistema. La lógica de DMA puede ser parte de un módulo de E/S, o puede ser un módulo separado que controla a uno o más módulos de E/S. Este concepto se puede llevar algo más lejos conectando los módulos de E/S a un módulo de DMA mediante un bus de E/S (C). Esto reduce a uno el número de interfaces de E/S en el módulo de DMA y permite una configuración fácilmente ampliable. En todos estos

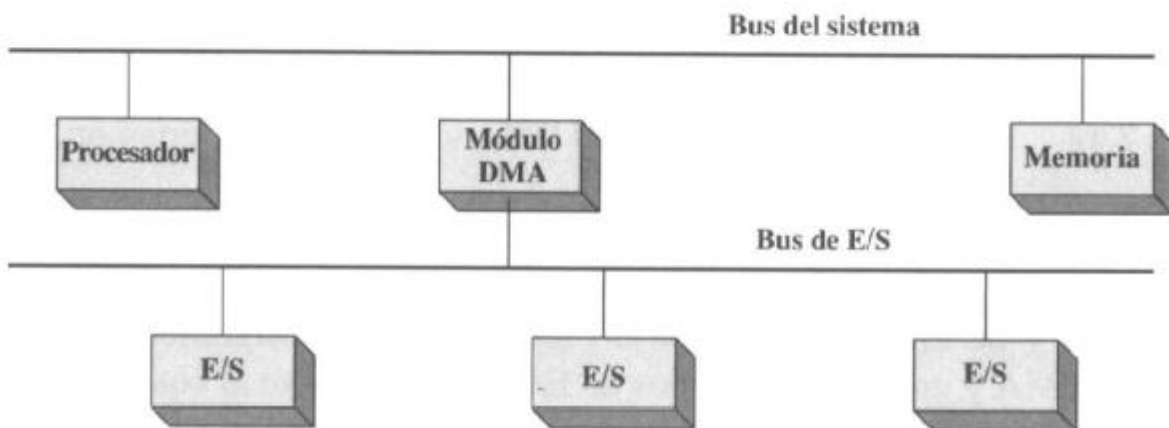
casos el bus de sistemas (B y C), que el módulo de DMA comparte con el procesador y la memoria, es usado como módulo de DMA solo para intercambiar datos con la memoria. El intercambio de datos entre los módulos DMA y E/S se produce fuera del bus del sistema.



(a) Bus único, DMA independiente



(b) Bus único, DMA-E/S integrados



(c) Bus de E/S

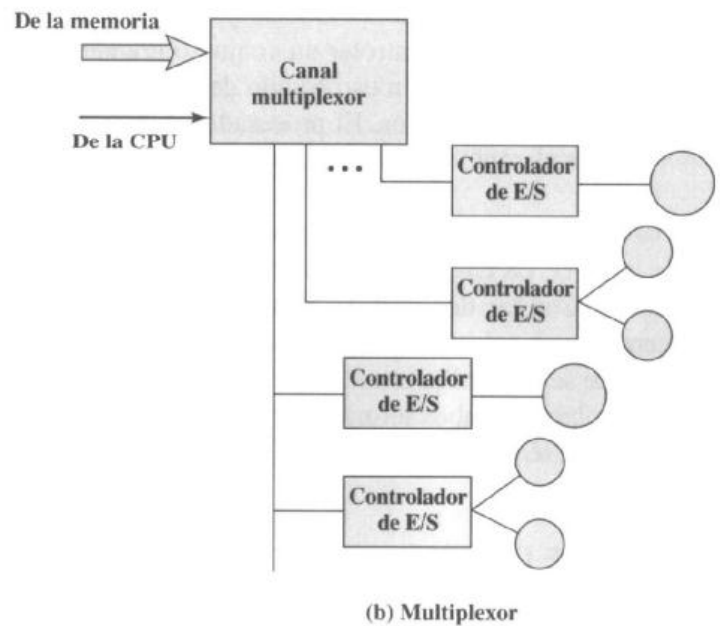
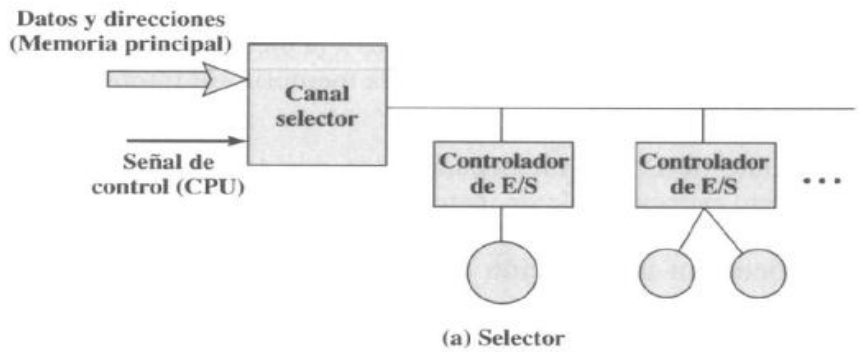
Canales y procesadores de E/S

Canales:

El canal de E/S representa una ampliación del concepto de DMA. Un canal de E/S puede ejecutar instrucciones de E/S (propias), lo que le confiere un control completo sobre las operaciones de E/S. En un computador de tales dispositivos, la CPU no ejecuta instrucciones de E/S. Dichas instrucciones se almacenan en memoria principal para ser ejecutadas por un procesador de uso específico contenido en el propio canal de E/S. De esta forma, la CPU inicia una transferencia de E/S indicando al canal de E/S que debe ejecutar un programa de la memoria.

Son comunes dos tipos de canales de E/S. Un **canal selector** controla varios dispositivos de velocidad elevada y, en un instante dado, se dedica a transferir datos a uno de esos dispositivos. Es decir, el canal de E/S selecciona un dispositivo y efectúa la transferencia de datos. Cada dispositivo, o pequeño grupo de dispositivos, es manejado por un controlador, o módulo de E/S.

Un **canal multiplexor** puede manejar E/S de varios dispositivos al mismo tiempo. Para dispositivos de velocidad reducida, un multiplexor de byte acepta o transmite caracteres tan rápido como es posible a varios dispositivos.



Interrupciones:

Es un mecanismo de hardware (lo realizan los circuitos), asincrónico con la ejecución del programa, mediante el cual el computador deja de ejecutar el programa en curso y pasa a ejecutar otro programa llamado Rutina de Atención a la Interrupción (RAI).

Cuando ocurre una interrupción...:

1. Se termina de ejecutar la instrucción en curso.
 2. Se guarda (por hardware) la PSW (Program Status Word) en un lugar fijo de la memoria, llamado PSW "vieja". Se carga la PSW "nueva" sobre la PSW en curso. La PSW nueva, apunta a la RAI.
 3. Se ejecuta la RAI.
 4. La ultima instrucción de la RAI es LPSW (Load PSW) que restaura la PSW "vieja" y continua con la ejecución del programa interrumpido (el retorno es por software).
- El estado actual de la CPU puede modificarse cargando una nueva PSW o modificando una parte de la PSW en curso.

Máscara de sistema	Clave de Protección	Señales	Código de interrupción	ILC	CC	Máscara de programa	Dirección de instrucción
0	7 8	11 12	15 16	31 32 33	34 35 36	39 40	63

Algunos de los bits de la PSW se prestan para enmascarar ciertas interrupciones. Una vez enmascaradas, las interrupciones de E/S, externas o de máquina deben ser inhibidas temporariamente y quedar pendientes. Dicho enmascaramiento afecta solo a 4 de las 15 interrupciones del programa. Algunas de las instrucciones de conmutación de estado son: LPSW (Load PSW), SPM (Set Program Mask), SSM (Set System Mask), SVC (Supervisor Call), SSK (Set Storage Key) e ISK (Insert Storage Key).

Si mientras se está ejecutando una interrupción aparece otra, puede llamarse a una rutina de cola de interrupciones. Si mientras se está ejecutando una cola de interrupciones ocurre una interrupción más, se volverá a llamar a la rutina de cola, lo que puede ocasionar la pérdida de una interrupción. Para evitar esto, se debe enmascarar todas las interrupciones mientras la CPU procesa una cola.

Tipos de interrupciones:

E/S: tiene lugar cada vez que un canal finaliza una operación de E/S. El canal le cambia la PSW al computador. La RAI revisa la CSW.

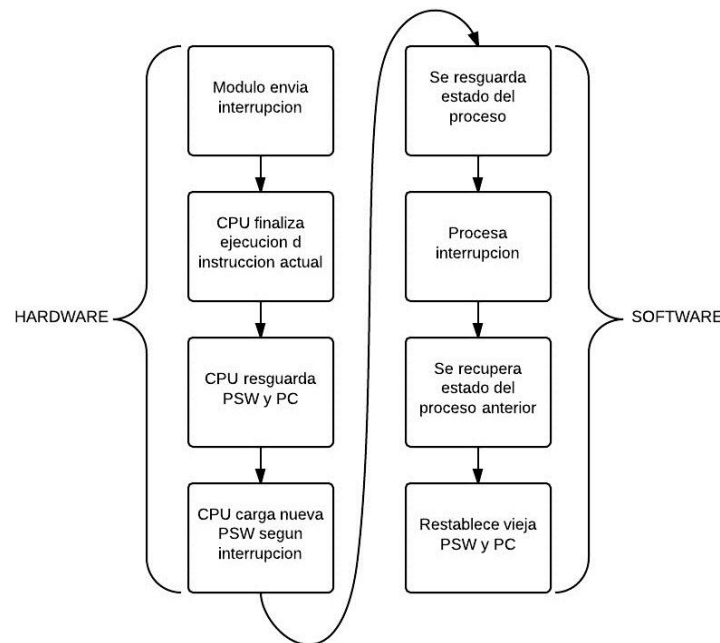
Por programa: provocadas por acontecimientos que suceden durante la ejecución del programa y que no fueron programadas:

1. **Operación:** si se intenta ejecutar una instrucción cuyo CO no existe.
2. Instrucción privilegiada: cuando un usuario intenta ejecutar una instrucción privilegiada.
3. Execute: le dice al sistema que ejecute una instrucción que está en otro lugar de la memoria. La interrupción se produce cuando se hace un execute de un execute.
4. Protección de memoria: cuando un programa quiere acceder a una zona de memoria que no tiene asignada, un área no permitida.
5. **Direccionamiento:** cuando se quiere acceder a un área de memoria no existente (addressing).
6. Especificación: cuando uso una instrucción que precisa entorno a palabra con una dirección sin entornar.
7. Datos (data exception): cuando los datos no respetan el formato.
8. **Overflow:** puede ser de punto fijo, de flotante o de empaquetado. Si no está inhibida la condición de overflow, cancela el programa.
9. **División por cero:** puede ser de punto fijo, flotante o empaquetado.
10. **Underflow de punto flotante:** no cancela, sino que pone 0 y sigue adelante.
11. Significación: cuando en punto flotante se pierden dígitos en la mantisa.

Por llamada a Supervisor: cada vez que se ejecuta una instrucción SVC (Supervisor Call). Esta interrupción no es asincrónica, está escrita en el programa (autointerrupción). Está incluida en las macroinstrucciones, pues estas requieren del uso del SO (OPEN; CLOSE, etc.). Tiene un código que va de 0 a 255, según el cual se sabe que rutina debe ver.

Rutinas del SO: Luego de intercalar las instrucciones de, por ejemplo, GET, el ensamblador se fija en el código de la última instrucción (SVC) que le dice que era una GET y va a las rutinas que se ocupan del bloqueo y desbloqueo. Cada vez que se hace una GET un registro lógico pasa al área de E/S (en memoria). El SO se da cuenta que tiene que ir a buscar un nuevo registro físico por:

- IOCS (Input Output Control System): consta de dos partes:
 - LIOCS (Logical IOCS): contiene las rutinas de bloqueo y desbloqueo. Se fija si hay un nuevo registro lógico, si hay lo pasa y corre el puntero, sino le pasa el control a la PIOCS.



- PIOCS (Physical IOCS): maneja el canal para pasar un nuevo registro físico. Hace la SIO, etc. Cuando el canal termina de trabajar hace una interrupción de E/S.

Externa: puede deberse a tres causas: el operador pulsa la tecla externa (en ese momento el SO interactúa con el operador), reloj de la máquina (timer) (estrategia de asignación del procesador por un tiempo determinado, el timer le avisa al SO entonces hay una interrupción), o comunicación entre computadoras (retorna mediante una interrupción externa).

Por error de máquina: se divide en dos tipos:

- Recuperable: es SO lo atiende, guarda todo el programa en un archivo y luego devuelve su control. Por ejemplo; falla en cinta.
- No recuperable: no retorna nunca.

Prioridad de atención de interrupciones

No se puede atender a más de una interrupción del mismo tipo al mismo tiempo, por eso existen prioridades de ejecución de interrupciones. Las prioridades permiten que una interrupción interrumpa una RAI en proceso

Está inhibida	E/S	Programa	SVC	Externa	Máquina
Se está procesando					
E/S	0	1	1	1	1
Programa	0	0	1	1	1
SVC	0	0	0	1	1
Externa	0	0	0	0	1
Máquina	0	0	0	0	0

0 : inhibida 1 : desinhibida. La más prioritaria es la de error de máquina.

Dos arquitecturas genéricas de procesadore

CISC(Complex instruction set computer (IBM mainframe e intel x86))

- Muchas instrucciones y muy complejas.
- Muchos formatos de instrucciones.
- Instrucciones de distinto tamaño.
- Muchas instrucciones para acceder a operandos en memoria.
- Muchos modos de direccionamiento.
- Muchos tipos de datos.
- Micro-arquitectura software/hardware compleja.
- Pocos registros (normalmente dedicados).
- Compiladores relativamente simples.

Risc (Reduced instruction set computer)

- Pocas instrucciones.
- Instrucciones simples.
- No hay muchas instrucciones para acceder a memoria (load/store).
- Instrucciones de igual tamaño.
- Pocos formatos d instrucción.
- Pocos modos de direccionamiento (y simples).
- Pocos tipos de datos.
- Micro-arquitectura en hardware y simple.
- Muchos registros (de uso general).
- Compiladores complejos.

Los errores de transmisión

Se denomina error de transmisión a:

Toda alteración o mutilación que hace que un mensaje recibido no sea una réplica fiel del mensaje transmitido.

Los errores pueden alterar el contenido del mensaje de dos formas:

- Que el mensaje quede totalmente invalidado.
- A pesar de los errores, el mensaje es útil porque aún puede interpretarse.

Existen dos tipos de transmisiones a considerar:

- De operador a operador; La intervención de los operadores permite la detección y corrección de los errores.
- De Máquina a máquina: En este caso se usa la información almacenada en el sistema para poder detectar automáticamente los errores analizarlos, aceptarlo o rechazarlos parcialmente o totalmente.

Tipos de errores

Clasificación (según su distribución en el tiempo)

- **Errores aislados o simples:** Aquellos que afectan a un solo bit cada vez y son independientes entre sí en cuanto al momento de ocurrencia.
- **Errores en ráfagas:** Afectan a varios bits consecutivos y ocurren en periodos indeterminados de tiempo.
- **Errores agrupados:** Ocurren en tandas sucesivas de una cierta duración y que no afectan necesariamente a varios bits seguidos.

Los errores y su tratamiento

Una forma es enviar datos adicionales en el contenido del mensaje, a esto se lo denomina redundancia.

Aunque se logra una mayor protección contra errores, cuanto mayor es la cantidad de bits adicionales, la eficiencia del proceso de transmisión disminuye. Por otra parte, cuando los bloques de mensaje son largos, una mayor proporción de estos bloques tendrán errores y será necesario retransmitirlos, lo que a largo plazo puede disminuir aún más su eficiencia.

La tasa de error en el diseño de sistemas de transmisión de datos

La presencia de errores no puede ser omitida nunca. Se deben tener presente los siguientes aspectos:

- La tasa de error (Tanto para las comunicaciones locales como para las remotas).
- Los medios para recuperar la información afectada por errores.
- La cantidad de información a transmitir por unidad de tiempo, que se corresponde con el concepto definido como **velocidad real de transferencia de datos**.
- La velocidad de transmisión que es necesaria para satisfacer los requerimientos del sistema.

Basándose en los aspectos anteriores se deberá determinar:

- El ancho de banda del canal que será necesario usar.
- El tipo de control de errores que se requiere.
- Los medios de comunicaciones y elementos técnicos que podrán cumplir con dichos requerimientos.

Técnicas de corrección de errores

Consideraciones generales

La corrección de errores es un hecho casi imprescindible. Existen dos estrategias fundamentales para la corrección de errores:

- **Corrección hacia atrás;** Consiste en el uso de sistemas de detección de errores. Cuando se detecta un error en el equipo receptor, este solicita al equipo transmisor la repetición del bloque de datos transmitido, de ahí la llamada corrección hacia atrás.
- **Corrección hacia adelante;** Esta técnica denominada Forward Error Correction (FEC), se basa en el uso de códigos auto-correctores que se diseñan sobre la base de sistemas de codificación redundante y corrigen los errores detectados en la misma estación que recibe el bloque de datos.

Control de paridad

Paridad par e impar

En la paridad par, si el número de unos de la palabra de información a transmitir es impar, el bit de control que se debe agregar será un uno, para que la suma total de ellos resulte un número par. Para el caso de paridad impar se busca que el número de unos sea impar.

Control de paridad vertical

Añade un bit adición al conjunto de 7 bits. Su uso está relacionado fundamentalmente con el código ASCII.

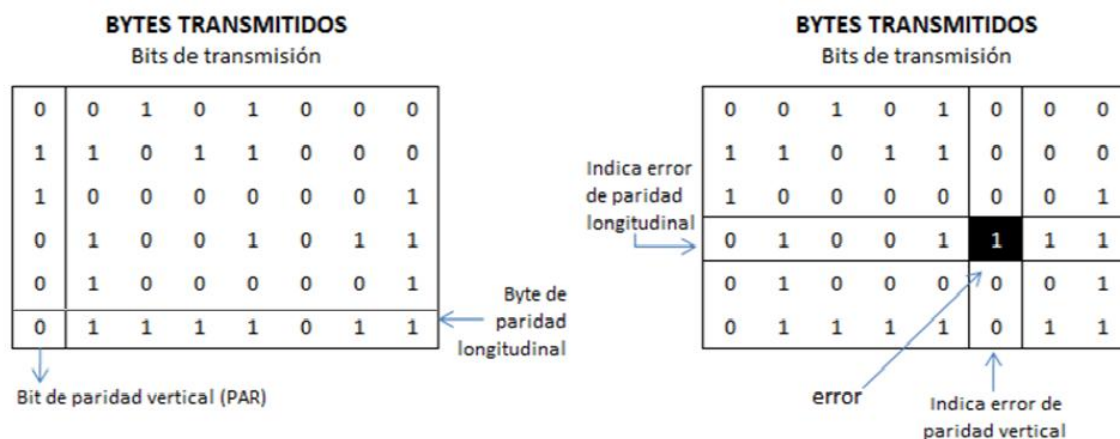
Control de paridad longitudinal o bidimensional

Se aplica a un conjunto de N caracteres de 7 bits cada uno. Por cada carácter de 7 bits transmitido se agrega un bit de paridad vertical.

Al final el bloque de N caracteres, se transmitirá un carácter completo denominado carácter de control de bloque (Block Check Character BCC). El BCC se calcula de la siguiente manera:

- Se consideran los bits en la posición uno de cada carácter y se calcula el bit de paridad para todo el bloque correspondiente a esa posición. Luego se realiza el mismo proceso por cada bit de carácter incluyendo el bit de paridad.

Método de paridad bidimensional



Control de paridad entrelazada o cíclica

Proporciona un nivel de detección de los errores de mayor calidad que el de paridad vertical y mejor que el de paridad longitudinal.

Requiere de dos bits adicionales para el control de paridad. El primer bit de paridad proporciona la paridad de los bits primero, tercero y quinto, mientras que el segundo proporciona la paridad de los bits segundo, cuarto y sexto.

Códigos Pre-Hamming

Paridad

- Añade un bit (bit de paridad): 0 Si la cantidad de 1s es par, 1 en caso contrario.
- Si un bit cambia por error se detecta el error pero no el bit erróneo.
- Si cambian dos bits, el bit de paridad será válido y el error no será detectado.
- No es un chequeo muy bueno pero produce poca sobrecarga (un único bit).

Dos entre cinco

- Basado en que cada bloque de cinco bits (penta-bit) tuviera exactamente dos unos y tres ceros (lo que forma 10 posibles estados, pudiendo representar todos los dígitos del rango 0-9)
- Si un bit cambia por error se detecta el error pero no el bit erróneo.

- Permite detectar cambios en un solo bit, pero si en un mismo penta-bit un 0 cambia por un uno y un uno por un cero, la regla se cumple y el error queda sin descubrir.

Repetición

- Consiste en repetir cada bit de datos varias veces (por ejemplo, si el bit de dato a enviar fuera un 1, con código de repetición $n=3$, enviaría 111).
- La detección de errores actúa como si el bit original se obtuviera por mayoría.
- Si el error en la transmisión provocara el cambio simultáneo en dos bits y el receptor recibiera 001, el sistema corregiría inadecuadamente el error.
- Este código es extremadamente ineficaz, pues reduce la velocidad de transmisión por n .

Distancia de Hamming

Distancia de Hamming: Numero de bits en que difieren dos secuencias binarias, S_1 y S_2 , de la misma longitud.

Si dos palabras de código difieren en una distancia d , se necesitan d errores para convertir una en la otra.

La distancia de hamming está relacionada con la probabilidad de error. Cuanto mayor sea la distancia mínima entre los símbolos de un código dado, menor será la probabilidad de cometer errores. Aumentando la redundancia se logra disminuir la probabilidad de cometer errores.

Código de Hamming

Los bloques de bits transmitidos estarán compuestos por n bits: k bits de datos y r bits de redundancia.

- Podrán formarse 2^n códigos y 2^k mensajes, es decir, 2^k códigos válidos.
- Si $n = k + r$, entonces habrá n códigos a distancia 1 de un código valido.
- Al haber $n+1$ códigos asociados a cada mensaje(o código valido) (n códigos inválidos a distancia 1 del mismo más el propio código valido) entonces: $(n+1) * 2^k \leq 2^n$

La cantidad de bits que puede tener el mensaje para ser corregido con r bits de redundancia ha de ser menor o igual a $2^r - 1$.

Es un método propuesto por R. W. Hamming usando la distancia mínima r . Con este método, por cada entero r existe un código de Hamming $2^r - 1$ bits que contiene r bits de paridad y $2^r - 1 - r$ bits de información.

Los bits de información y los bits de paridad se encuentran entremezclados de la siguiente forma:

- Todos los bits cuya posición es potencia de dos se utilizan como bits de paridad (posiciones 1, 2, 4, 8, 16, 32, 64, etc.).
- Los bits del resto de posiciones son utilizados como bits de datos (posiciones 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.)

El valor de cada bit de paridad es la paridad par de un número determinado de bit. Estos grupos se escogen de tal forma que ningún bit de información se cubra con la misma combinación de bits de paridad. Esto es lo que proporciona al código su capacidad de corrección.

Por ejemplo: Consideremos la palabra de datos de 7 bits "0110101"

	r1	r2	k1	r3	k2	k3	k4	r4	k5	k6	k7
Palabra de datos (sin paridad):			0		1	1	0		1	0	1
r1	1		0		1		0		1		1
r2		0	0			1	0			0	1
r3				0	1	1	0				
r4								0	1	0	1
Palabra de datos (con paridad):	1	0	0	0	1	1	0	0	1	0	1

Cálculo de los bits de paridad en el código Hamming

Almacenamiento secundario

Discos magnéticos

Compuestos por platos recubiertos con material magnetizable (óxido de hierro) y un armazón que sirve de soporte al brazo que porta las cabezas lectoras grabadoras, cada plato se denomina disco.

Los datos se graban y después se recuperan del disco a través de una bobina, llamada cabeza; en muchos sistemas hay dos, una de lectura y otra de escritura.

Durante la operación de lectura o escritura, esta permanece quieta mientras el plato rota bajo ella.

Escritura: Cuando circula electricidad a través de una bobina se produce un campo magnético. Los patrones magnéticos resultantes se graban en la superficie (diferentes patrones para corrientes + y -).

Lectura: Un campo magnético que se mueve por una bobina produce corriente eléctrica en ella. Cuando la superficie del disco pasa por debajo de la cabeza se genera una corriente de la misma polaridad grabada.

Los discos duros de hoy usan un mecanismo diferente para la lectura, siendo necesaria una cabeza de lectura separada posicionada, por conveniencia, cerca de la cabeza de escritura. La cabeza de lectura consiste en un sensor magneto-resistivo (MR) parcialmente blindado.

Se hace pasar una corriente a través del sensor MR y los cambios de resistencia detectan como señales de voltaje.

Este sensor provee mayores densidades de grabación y velocidad de operación que el mecanismo anterior.

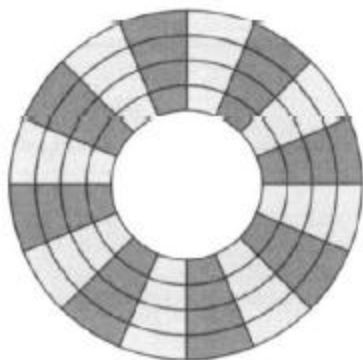
Organización y formato de los datos

La cabeza es un dispositivo relativamente pequeño, capaz de leer o escribir en una zona del plato que rota bajo ella. Esto da lugar a que los datos se organicen en un conjunto de anillos concéntricos, llamados pistas.

Cada pista es del mismo ancho que la cabeza.

Normalmente hay cientos de sectores por pista y estos pueden tener una longitud variable o fija. En la mayoría de los sistemas de hoy se utilizan sectores de longitud fija. Para evitar imposiciones de precisión ilógicas del sistema, los sectores adyacentes se separan con intra-pistas vacías.

Un bit cercano al centro de un disco girando, pasa por un punto fijo más despacio que un bit más externo. Por tanto, debe haber alguna forma de compensar la variación de la velocidad de forma que la cabeza pueda leer todos los bits a la misma velocidad. Esto se puede hacer incrementando el espacio entre bits de la información grabada en los segmentos del disco.



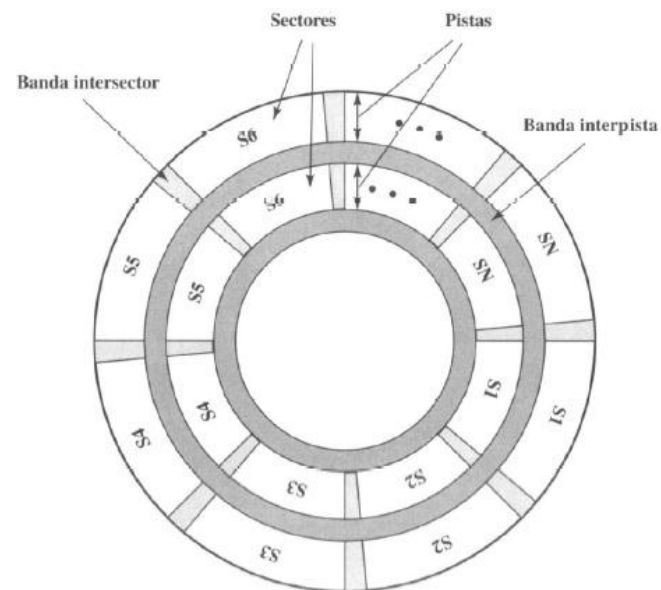
Velocidad angular constante

La información se puede escanear a la misma velocidad rotando el disco a una velocidad fija, conocida como velocidad angular constante (constant angular velocity, CAV).

El disco se divide en una serie de sectores en forma de trozo de tarta y en una serie de pistas concéntricas.

Ventajas:

- Los bloques individuales de datos se pueden direccionar directamente con la pista y sector. Para mover la cabeza desde su actual posición a una dirección específica.



Desventajas:

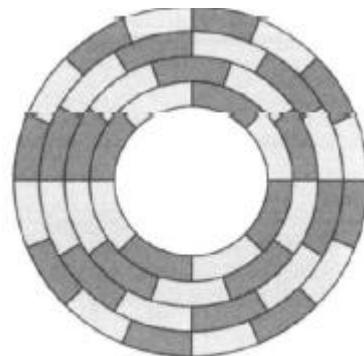
-La cantidad de datos que se pueden almacenar en las pistas más externas es solo la misma que la de las pistas internas.

Por esto la capacidad de almacenamiento viene limitada por la máxima densidad de grabación que se puede llevar a cabo en la misma más interna.

Para aumentar la capacidad los discos duros modernos utilizan una técnica llamada grabación en varias zonas (multiple zone recording), en la que la superficie se divide en varias zonas concéntricas. Dentro de una zona, el número de bits por pista es constante. Las zonas más lejanas del centro contienen más bits (más sectores) que las zonas próximas al centro. Esto permite capacidades de almacenamiento mayores a expensas de una circuitería de alguna forma más compleja.

Algún procedimiento es necesario para situar las posición del sector en una pista. Claramente, debe haber algún punto de comienzo de la pista y una manera de identificar el principio y el fin de cada sector.

Estos requisitos son gestionados mediante datos de control grabados en el disco. Por tanto, el disco se graba con un formato que contiene algunos datos extra usados solo por el controlador del disco y no accesibles al usuario.



Grabación en varias zonas

Características físicas

-Las cabezas pueden ser fijas o móviles con respecto a la dirección radial del plato. En un disco de cabeza fija hay una cabeza de lectura/escritura por pista. En un disco de cabeza móvil, hay solo una cabeza de lectura/escritura.

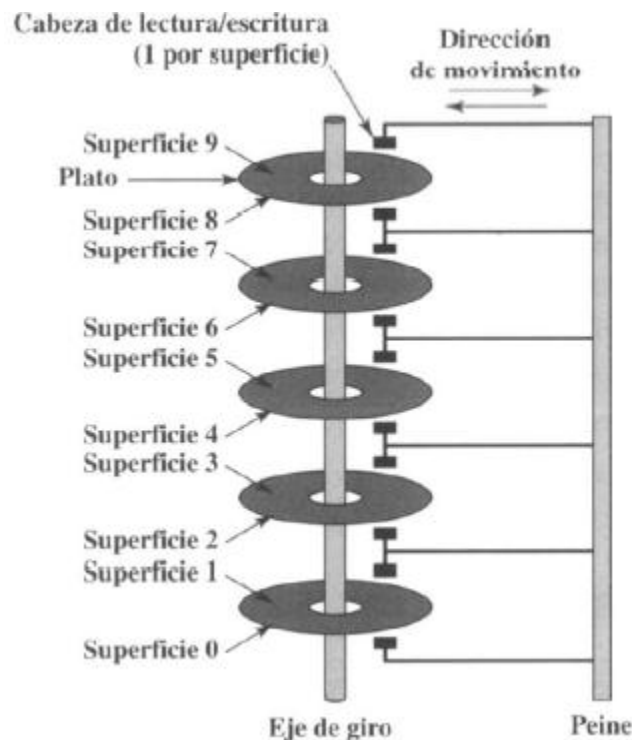
-Un disco no extraíble está permanentemente montado en la unidad de disco. Un disco extraíble, puede ser quitado y sustituido por otro disco. La ventaja de este último tipo es que es posible una cantidad de datos ilimitada con un número ilimitado de unidades de disco, además de que un disco puede ser utilizado en varios computadores.

-En la mayoría de los discos, la cubierta magnetizable se aplica a ambas caras de los platos, denominándose de doble superficie. Algunos discos, menos caros, son de una sola superficie.

-Algunas unidades de disco poseen varios platos apilados verticalmente. Disponen de varios brazos. Los discos de varios platos utilizan una cabeza que se mueve, con una cabeza de lectura/escritura por cada superficie del plato. El conjunto de todas las pistas que tienen la misma posición relativa en el plato se denomina cilindro.

-Finalmente, el mecanismo de la cabeza proporciona una clasificación en tres tipos. Tradicionalmente, la cabeza de lectura/escritura se posiciona a una distancia fija sobre el plato, dejando entre ambos una capa de aire. En el otro extremo está el mecanismo de la cabeza que realmente efectúa un contacto físico con el medio durante la operación de lectura o escritura (usado por disquetes).

Para el tercer tipo, la cabeza debe generar o detectar un campo magnético de intensidad suficiente para escribir y leer correctamente. Cuando más estrecha es la cabeza, más cercana de estar a la superficie del plato para funcionar. Esto es deseable, puesto que una cabeza más estrecha implica picas más estrechas y



por lo tanto, mayor densidad de datos. Sin embargo, cuanto más cerca este la cabeza del disco, mayor era el riesgo de error debido a impurezas o imperfecciones.

Paramentos para medir performance

Los detalles de las operaciones de E/S de un disco dependen del tipo de computador, del sistema operativo, de la naturaleza de los canales de E/S y del hardware de controlador del disco.

En un sistema de cabeza móvil, el tiempo que tarda la cabeza en posicionarse en la pista se conoce como **tiempo de seek**. En cualquier caso, una vez seleccionada la pista, el controlado del disco espera hasta que el sector apropiado rota hasta alinearse con la cabeza. El tiempo que tarda el sector en alcanzar la cabeza se llama retardo rotacional o **latencia rotacional**. La suma de tiempo de búsqueda, si lo hay, y la latencia rotacional se llama **tiempo de acceso**. Una vez posicionada la cabeza, se lleva a cabo la operación de lectura o escritura, desplazándose el sector bajo la cabeza; esta operación conlleva un **tiempo de transferencia de datos**

$$T = b/rN$$

donde T = tiempo de transferencia; b bytes a transferir;

r=velocidad de rotación en revoluciones por segundo; N = bytes por pista

Tiempo total de lectura/escritura

$$T = T_{\text{seek}} + \frac{1}{2} r + b/rN$$

RAID (Redundant Array of Independet Disks)

Con el uso de varios discos, hay una aplica variedad de formas en las que se pueden organizar los datos, y en las que se pueden añadir redundancia para mejorar la seguridad. Esto podría dificultar el desarrollo de esquemas de bases de datos que se pueden usar en numerosas plataformas y sistemas operativos.

Afortunadamente, la industria está de acuerdo con los esquemas estandarizados para el diseño de bases de datos para discos múltiples, conocidos como RAID. El Esquema RAID consta de síes niveles independientes, desde cero a cinco. Estos niveles no implican una relación jerárquica, sino que designan métodos diferentes que poseen tres características comunes:

- RAID es un conjunto de unidades físicas de disco vistas por el sistema operativo como una única unidad lógica.
- Los datos se distribuyen a través de las unidades físicas del conjunto de unidades.
- La capacidad de los discos redundantes se usa para almacenar información de paridad que garantice la recuperación de los datos en caso de fallo de disco.

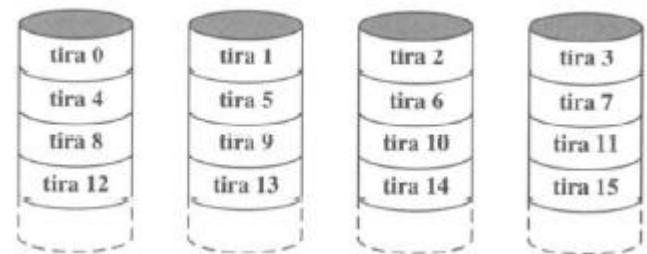
Nivel 0

El nivel 0 de RAID no es un verdadero miembro de la familia RAID, porque no incluye redundancia para mejorar las prestaciones. Sin embargo, hay algunas aplicaciones, como algunas ejecuciones en supercomputadores, n los que las prestaciones y la capacidad son la preocupación primaria y un costo bajo es más importante que mejorar la seguridad.

Para RAID 0, los datos del usuario y del sistema esta distribuidos a lo largo de todos los discos del conjunto. Esto tiene una notable ventaja frente al uso de un único disco: Si hay pendientes dos peticiones diferentes de E/S, para dos bloques de datos diferentes es muy probable que los bloques pedidos este en diferentes discos. Entonces las dos peticiones se pueden emitir en paralelo, reduciendo el tiempo de cola de E/S.

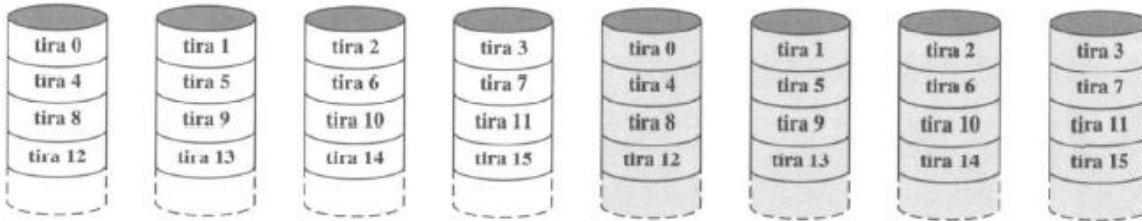
Los datos son organizados en forma de tiras de datos a traes de los discos disponibles.

Todos los datos del usuario y del sistema se ven como almacenados en un disco lógico. El disco se divide en tiras; estas tiras pueden ser bloques físicos, sectores o alguna otra unidad. Las tiras se proyectas cíclicamente, en miembros consecutivos del conjunto.



Nivel 1

RAID 1 se diferencia de los niveles 2 al 6 en cómo se consigue la redundancia. En estos otros esquemas RAID, se usan algunas formas de cálculo de paridad para introducir redundancia, en RAID 1 la redundancia se logra con el sencillo recurso de duplicar todos los datos.



En la organización de RAID 1 hay una serie de aspectos positivos:

- Una petición de lectura puede ser servida por cualquiera de los discos que contienen los datos pedidos. Cualquiera de ellos implica un tiempo de búsqueda mínima más la latencia rotacional.
- La escritura se hace en forma independiente en cada disco y nos penaliza
- Alta disponibilidad de datos
- La recuperación tras un fallo es sencilla. Cuando una unidad falla, se puede acceder a los datos desde la segunda unidad.

La principal desventaja es el coste; requiere el doble del espacio de disco del disco lógico que puede soportar. Debido a esto, una configuración RAID 1 posiblemente está limitada a unidades que almacenan el software del sistema y los datos, y otros ficheros altamente críticos.

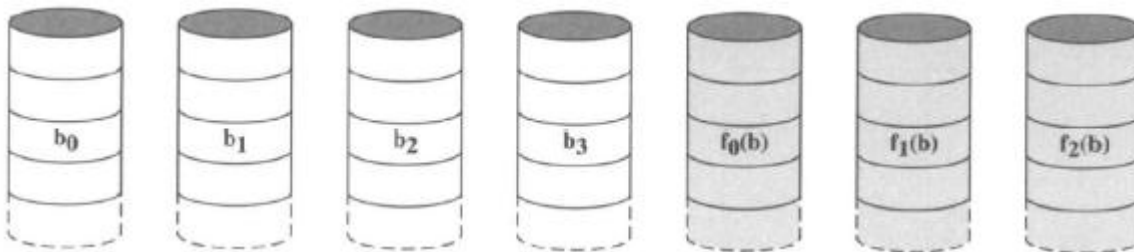
Nivel 2

Los niveles 2 y 3 de RAID usan una técnica de acceso paralelo. En un conjunto de acceso paralelo, los discos miembros participan en la ejecución de cada petición de E/S. Típicamente, el giro de cada unidad individual está sincronizado de forma que cada cabeza de disco está en la misma posición en cada disco en un instante dado.

Como en los otros esquemas RAID, se usa la descomposición de datos en tiras. En el caso de RAID 2 y 3, las tiras son muy pequeñas, a menudo tan pequeñas como un único byte o palabra.

Con RAID 2, el código de corrección de errores se calcula a partir de los bits de cada disco, y los bits del código se almacenan en las correspondientes posiciones de bit en varios discos de paridad. Normalmente se usa el código de Hamming, que permite corregir errores en un bit y detectar errores en dos bits.

RAID 2 debería ser solamente una elección efectiva en un entorno en el que haya muchos errores de disco. Se requieren $N+m$ discos (N , discos de datos; m , discos de redundancia).



Nivel 3

Se organiza de manera similar a RAID2. La diferencia es que requiere solo un disco redundante, sin importar lo grande que sea el conjunto de discos. Utiliza acceso paralelo, con datos distribuidos en pequeñas tiras. EN vez de un código de corrección de errores, se calcula un sencillo bit de paridad para el conjunto de bits individuales en la misma posición en todos los discos de datos.

Redundancia:

En el caso de un fallo en una unidad, se accede a la unidad de paridad y se reconstruyen los datos desde el resto de los dispositivos. Una vez que se sustituye la unidad que ha fallado, los datos que faltan se restauran en la nueva unidad y se reanuda la operación.

La reconstrucción de los datos es bastante sencilla. Consideraremos un conjunto de cinco discos, de los que X0 a X3 contiene datos y X4 es el disco de paridad. La paridad para el i-esimo bit se calcula de la siguiente forma:

$$X4(i) = X3(i) + X2(i) + X1(i) + X0(i)$$

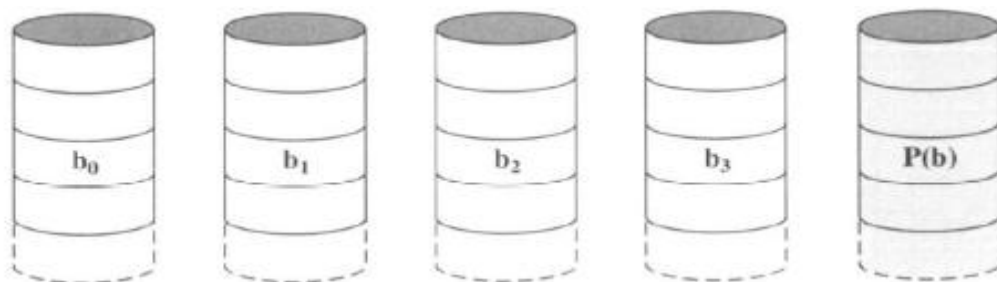
Donde + es la función or exclusiva.

Supongamos que la unidad X1 ha fallado. Si sumamos $X4(i) + X1(i)$ a ambos miembros de la ecuación tenemos:

$$X1(i) = X4(i) + X3(i) + X2(i) + X0(i)$$

Por lo tanto, se puede regenerar el contenido de cualquier tira de datos X1 a partir del contenido de las correspondientes tiras del resto de los discos del conjunto. Este principio es válido para los niveles 3 a 6 de RAID.

Puesto que los datos se dividen en tiras muy pequeñas. RAID 3 puede conseguir velocidades de transferencia de datos muy altas. Cualquier petición de E/S implica una transferencia de datos paralela desde todos los

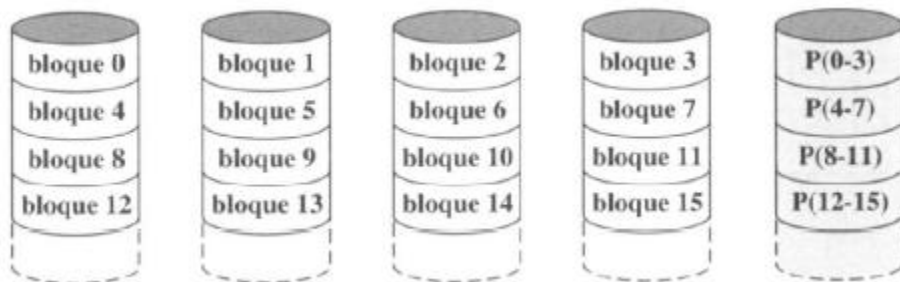


discos de datos. Para grandes transferencias, la mejora de prestaciones es especialmente notable. Por otra parte, solo se puede ejecutar a la vez una petición de E/S. Por lo tanto, en un entorno orientado a transacciones, el rendimiento sufre.

Nivel 4

Los niveles 4 la 6 de RAID usan una técnica de acceso independiente. En un conjunto de acceso independiente, cada disc opera independientemente, de forma que peticiones de E/S separadas se atienden en paralelo. Debido a esto, son más adecuados los conjuntos de acceso independiente para aplicaciones que requieran velocidad de petición de E/S altas, y menos adecuados para aplicaciones que requieren velocidades altas de transferencia de datos.

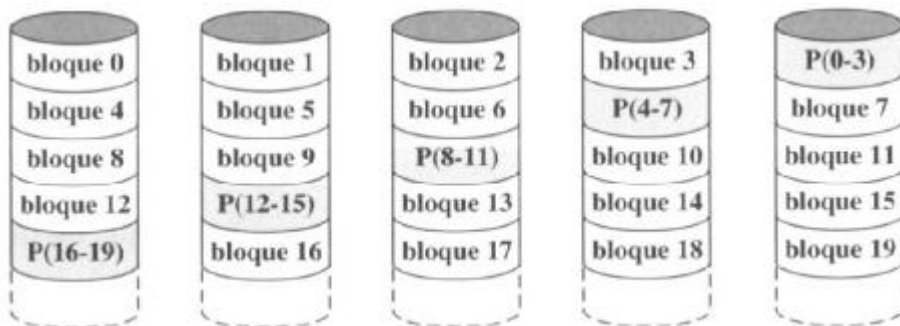
En este caso las tiras son relativamente grandes. Con RAID 4, se calcula una tira de paridad bit a bit a partir de las correspondientes tiras de cada disco de datos, y los bits de paridad se almacenan en la correspondiente tira del disco de paridad.



Para calcular la nueva paridad, el software de gestión del conjunto debe leer la antigua tira del usuario y la antigua tira de paridad. Entonces, se puede actualizar estas dos tiras con nuevos datos y calcular la nueva paridad. Por tanto, cada escritura de una tira implica dos lecturas y dos escrituras. En cualquier caso, cada operación de escritura implica al disco de paridad, que por consiguiente se convertirá en un cuello de botella.

Nivel 5

Este nivel está organizado de manera similar al RAID 4. La diferencia es que distribuye las tiras de paridad a lo largo de todos los discos. Una distribución típica es en un esquema cíclico, como se muestra. Para un conjunto de n discos, la tira de paridad esta en diferentes discos para las primeras n tiras, y este patrón se repite.

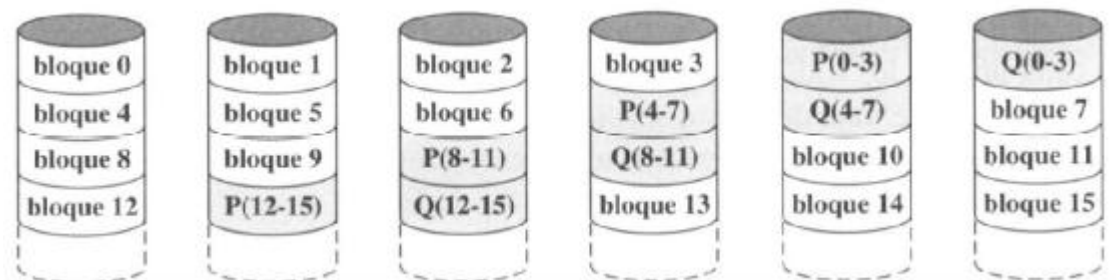


La distribución de las tiras de paridad a lo largo de todas las unidades evita el potencial cuello de botella de E/S encontrado en el RAID 4. La principal desventaja de este RAID es la complejidad del controlador.

Nivel 6

El nivel 6 de RAID se introdujo en un artículo de los investigadores de Berkeley. En el esquema del nivel 6 de RAID, se hacen dos cálculos de paridad distintos, que se almacenan en bloques separados en distintos discos. Por lo tanto, un conjunto RAID 6 con N discos, consta de $N+2$ discos.

P y Q son dos algoritmos de comprobación de datos distintos. Uno de los dos calcula OR-exclusivo y el otro un algoritmo de comprobación de



datos independiente. Esto hace posible la regeneración de los datos incluso si dos de los discos que contienen los datos de los usuarios fallan.

La ventaja de RAID 6 es que proporciona una disponibilidad de los datos extremadamente alta. Tendrían que fallar tres discos en el intervalo de tiempo medio de reparación para no poder disponer de los datos. Por otra parte, RAID 6 incurre en una penalización de escritura ya que cada escritura afecta a dos bloques de paridad.

Tabla de comparación de RAID

Nivel	Ventajas	Desventajas	Aplicaciones
0	Las prestaciones de E/S se mejoran mucho repartiendo la carga de E/S entre varios canales y unidades. No hay cálculo de paridad de cabecera. Diseño muy sencillo. Fácil de implementar.	El fallo de una sola unidad afectara a todos los datos de una estructura, perdiéndose.	Producción y edición de video. Edición de imágenes. Cualquier aplicación que requiera ancho de banda grande.
1	Una redundancia del cien por cien de todos los datos implica que no sea necesaria la reconstrucción en caso de fallo d disco, solo una copia del disco a reemplazar. Bajo ciertas circunstancias RAID 1 puede soportar varios fallos de unidades. El diseño del subsistema de almacenamiento RAID es sencillo.	La mayor sobrecarga de todos los tipos RAID (100%) ineficiente.	Contabilidad. Nominas. Finanzas. Cualquier aplicación que requiera una disponibilidad muy alta.
2	Son posibles velocidades de transferencia de datos extremadamente altas. Cuanto mayor es la velocidad de transferencia requerida, mejor es la relación entre discos de datos y disco ECC. Diseño del controlador relativamente sencillo en comparación con los niveles 3,4 y 5	Relación muy alta de discos ECC y discos de datos con tamaños de palabra pequeños (ineficiente). Coste de nivel de entrada muy alto (requisitos de velocidad de transferencias muy altas como para justificarlo).	No existen. No es comercialmente viable.
3	Velocidad de transferencia de datos de lectura y escritura muy alta. Un fallo de disco tiene un impacto insignificante en el rendimiento. Una baja de relación entre discos ECC (paridad) y discos de datos implica alta eficiencia.	Velocidad de transacción igual que la de una unidad de disco como mucho (si la velocidad de giro esta sincronizada). El diseño del controlado es bastante complejo	Producción de video y secuencias en vivo. Edición de imágenes. Aplicación de prueba de imprenta. Cualquier aplicación que requiera un alto rendimiento
4	Velocidad de transacción de datos de lectura muy alta. Una baja relación entre disco ECC (paridad) y discos de datos, implica una alta eficiencia	Diseño del controlador bastante complejo. Peor velocidad de transacción de escritura y velocidad de escritura total. Reconstrucción de datos difícil e ineficiente.	No existe
5	La mayor velocidad de transacción de datos. Una baja relación entre discos ECC (paridad) y discos de datos, implica una alta eficiencia. Buena velocidad de transferencia en su conjunto	Diseño del controlador más complejo. Difícil la reconstrucción en caso de fallo de disco	Servidores de ficheros y aplicaciones. Servidores de bases de datos. Servidores de páginas web. Nivel de RAID más versátil
6	Proporciona una tolerancia a fallos extremadamente alta y puede soportar varios fallos de unidades simultáneos	Diseño del controlador más compleja. Sobrecarga del controlador para calcular direcciones de paridad.	Solución perfecta para aplicaciones con objetivos críticos.

Medios ópticos

En 1983, se introdujo uno de los productos de consumo de más éxito de todos los tiempos: el disco compacto digital de audio. El CD es un disco no borrable que puede almacenar más de sesenta minutos de información de audio en una cara. El gran éxito comercial del CD posibilitó el desarrollo de la tecnología de discos de memoria óptica de bajo coste, que revolucionó el almacenamiento de datos en un computador.

CD-ROM

En discos ópticos la lectura se hace de acuerdo a la reflexión de un rayo láser sobre una superficie tratada en la cual se graban los datos mediante marcas llamadas pits. A las superficies que quedan entre dos pits se las denomina lands. La transición de land a pit y vuelta a land representa un uno. Un cierto período de tiempo sin transiciones representa un cero.

Debido a las características de los lectores no es posible mediante este esquema que existan dos unos seguidos en la superficie del disco. El problema se corrige empleando un sistema de traducción de los bits leídos desde el disco a bytes 'comunes'. Este sistema traduce 14 bits leídos del CD en un byte. El sistema se conoce como eight to fourteen modulation (EFM).

Cada CD tiene una única pista en espiral sobre la cual se graban los datos. Sobre esta pista se estructuran sectores todos ellos de la misma longitud lo cual permite grabar todos los sectores con la misma densidad de grabación. Este tipo de organización se denomina CLV (Constant Linear Velocity) pues la velocidad lineal de lectura es constante (espero hayan notado porque es que esto debe ser así) y se diferencia de la organización CAV (Constant Angular Velocity) empleada en los discos en donde la velocidad de rotación es constante. El esquema CLV perjudica el acceso random a los datos ya que para "saltar" desde una posición del disco hasta otra la unidad debe recalcular la velocidad a la cual deberá girar el disco. Esto se hace utilizando prueba y error. Los seeks en un CD son muy costosos, esto se debe a que los CDs de datos se basan en la industria ya existente para los CDs de música, donde la operación más frecuente es la lectura secuencial.

Un sector de un disco CLV se referencia mediante un número de la forma MM:SS:XX donde MM es el minuto, SS es el segundo, y XX el sector.

Cada CD puede almacenar aproximadamente 74 minutos de audio, y sabiendo que cada segundo se divide en 75 sectores nos da un total de 333000 sectores. Cada sector a su vez puede almacenar 2352 bytes, lo cual representa una capacidad de 747 megabytes.

CDs de Audio

- Capacidad: la capacidad estándar de un CD de audio es de 74 minutos.
- En un CD de audio el promedio de fallas es de un byte cada dos CDs.

CDs de Datos

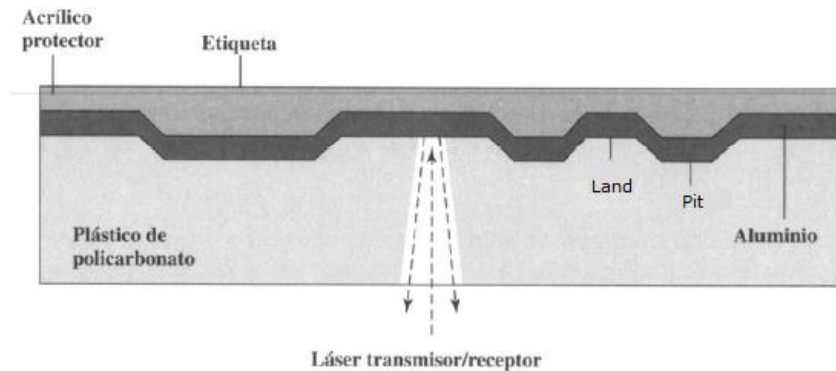
- Capacidad: por cada sector solamente 2048 bytes pueden contener datos. La capacidad de un CD de datos es entonces de 650 Mb.
- Con el agregado de los códigos correctores el promedio de fallas es de un byte equivocado cada 20.000 discos.

DVD-ROMs

Los DVD (*Digital Versatile Disc*) fueron creados en 1996, y surgieron como una necesidad de incrementar el mercado del video, ya que en un DVD la calidad de audio e imagen es alta. Actualmente también se utiliza para el almacenamiento de software en general (*office suites*, video juegos, etc.).

Hay cinco formatos de DVD:

1. DVD-ROM: medio de alta capacidad de almacenamiento de datos
2. DVD-Video: medio de almacenamiento digital de películas
3. DVD-Audio: formato de almacenamiento de audio únicamente, similar al CD de audio



4. DVD-R: ofrece un formato de almacenamiento de una sola grabación y múltiples lecturas
5. DVD-RAM: primer formato regrabable de DVD; luego compitió con DVD-RW y DVD+RW

Existen cuatro versiones del DVD:

1. DVD-5: de un solo lado y una sola capa, con capacidad de 4,7 GB
2. DVD-9: de un solo lado y dos capas, con capacidad de 8,5 GB
3. DVD-10: de dos lados y una capa, con capacidad de 9,4 GB
4. DVD-18: de dos lados y dos capas, con capacidad de 17 GB

La tecnología de capa dual tiene una capa reflectante abajo, cubierta por una capa semirreflejante.

Dependiendo de la distancia focal del láser, el haz se refleja en una capa o en la otra. La capa inferior necesita fosos y *lands* un poco mayores para que su lectura sea confiable, por lo que su capacidad es menor que la de la capa superior. 34

Además de sus cinco formatos físicos, el DVD también tiene varios formatos de aplicación como DVD de Video y DVD de Audio. La consola de juegos PlayStation 2 de Sony es un ejemplo de un formato especial de aplicación.

Cintas magnéticas

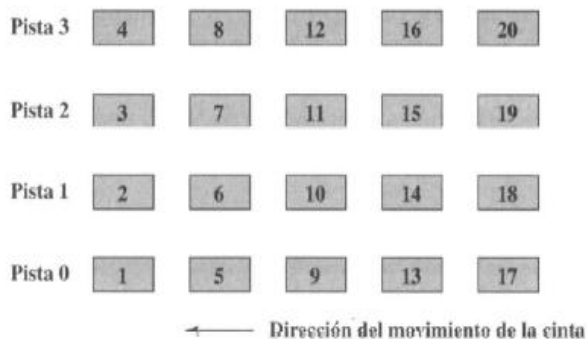
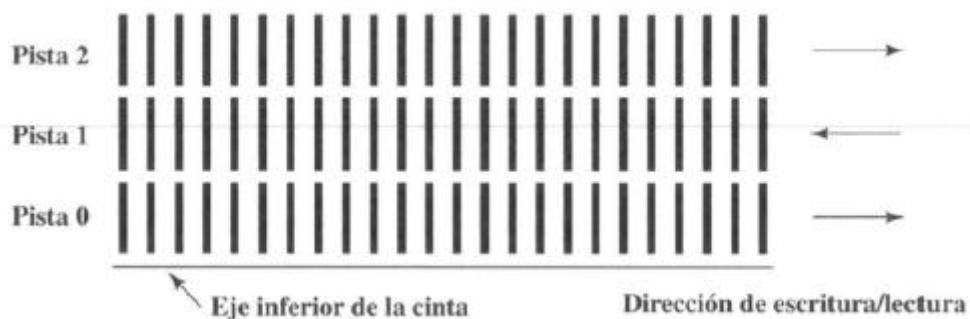
Los sistemas de cinta usan las mismas técnicas de lectura y grabación que los discos. El medio es una cinta de poliéster flexible cubierta por un material magnetizable. La cubierta puede consistir en partículas de un metal puro, en concreto, un revestimiento o película de metal plateado vaporizado. La cinta y la unidad de cinta son análogas a las cintas de grabación domésticas. Los anchos de las cintas pueden variar entre 0,38 cm y 1,27 cm. Una cinta ubicada en un carrete abierto, tiene que enrollarse en otro carrete ubicado en un segundo cabezal.

Hoy en día, prácticamente todas las cintas vienen cerradas en cartuchos.

Los datos en la cinta, se estructura en una serie de pistas paralelas longitudinales. Los primeros sistemas de cintas usaban nueve pistas. Esto hace posible almacenar datos de un byte en un instante dado, con un bit de paridad adicional, en una novena pista. Los nuevos sistemas de cintas usan 18 a 36 pistas, correspondiendo a una palabra digital. La grabación de datos de esta forma se denomina **grabación paralela**.

Los sistemas modernos utilizan en su lugar **grabación serie**, en la que los datos se disponen como una secuencia de bits a lo largo de una pista, como se hace en los discos magnéticos. Como con el disco, los datos se leen y escriben en bloques contiguos, llamados registros físicos de cinta. Los bloques en la cinta están separados por bandas vacías llamadas bandas inter-registros. Como en el disco, la cinta se formatea para facilitar la localización de los registros físico.

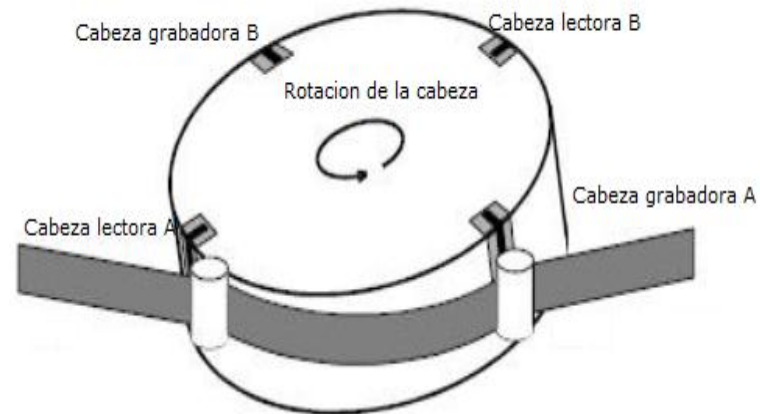
La técnica típica utilizada en la grabación de cintas en serie se denomina **grabación en serpentina**. En esta técnica, cuando se graban los datos, el primer conjunto de bits se graba a lo largo de toda la cinta cuando se alcanza el fin, las cabezas posicionan para



grabar una nueva pista y la cinta se graba de nuevo a todo lo largo, esta vez en dirección contraria. Este proceso continua, hacia atrás y adelante, hasta que se llene la cinta. Para aumentar la velocidad, la cabeza de lectura-escritura es capaz de leer y escribir una serie de pistas adyacentes simultáneamente (usualmente entre dos y ocho pistas). Los datos se graban en serie a lo largo de las pistas individuales, pero los bloques se almacenan en pistas adyacentes.

Grabación helicoidal

- Cabeza de grabación rotatoria.
- Símil video caseteras.
- Evita problemas de movimiento veloz de cintas de las otras técnicas.
- La cinta se mueve en forma lenta mientras que la cabeza rota en forma rápida.
- Las pistas pueden estar más cercanas unas a otras.



Modos de operación

Modo start-stop por bloque

- Viejo uso de grabación por registro/bloque.
- La cinta se usaba para guardar archivos para procesamiento posterior.
- Se podía actualizar un registro/bloque particular siempre y cuando no cambiara su tamaño.
- Los datos se grababan en bloques físicos.
- Entre los bloques había espacios (IRG) para sincronización de la unidad.

Modo streaming

- Uso de backup 0 archivo de información.
- No se requiere operación de start-stop por bloque.
- No se requiere actualización de bloques particulares dentro de un archivo.
- Se describen archivos completos como un “stream” de datos contiguo.
- La información se graba físicamente en bloques pero no se pueden localizar o modificar particulares.