ORGANIZACIÓN DEL COMPUTADOR

1)¿Cuáles son las ventajas y desventajas de la arquitectura Harvard en relación a la arquitectura Von Neumann? Grafique ambas arquitecturas y justifique.

Arquitectura Harvard

La arquitectura Harvard utiliza buses de datos separados para las instrucciones y los datos, lo que significa que puede acceder a ambas fuentes de manera simultánea. Las ventajas de esta arquitectura incluyen:

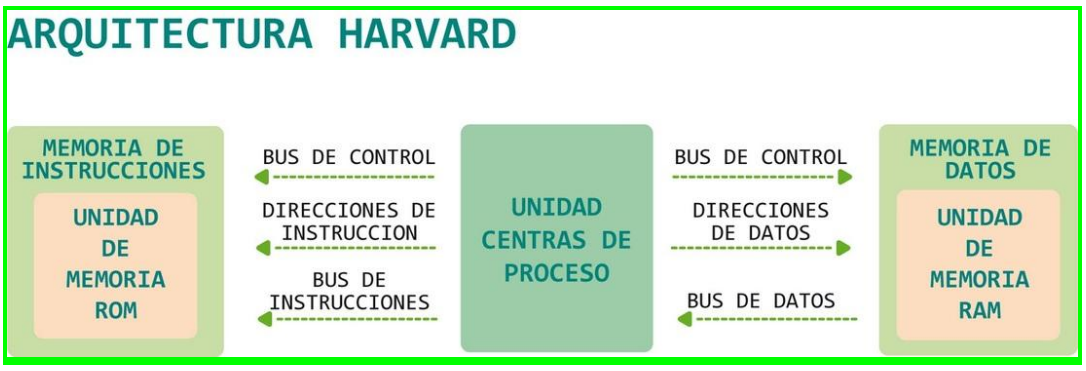
Ventajas:

1. Mayor velocidad: Al separar los buses de datos para las instrucciones y los datos, la arquitectura Harvard puede acceder simultáneamente a ambas fuentes, lo que aumenta la velocidad de procesamiento. Esto hace que la arquitectura Harvard sea ideal para aplicaciones que requieren un alto rendimiento, como el procesamiento de imágenes y el reconocimiento de voz.
2. Menor latencia: Al separar los buses de datos, la arquitectura Harvard reduce la latencia entre la memoria y la CPU, lo que permite que la CPU procese los datos más rápidamente.
3. Mayor seguridad: La arquitectura Harvard es más segura que la arquitectura Von Neumann, ya que la CPU solo puede acceder a la memoria de instrucciones y no puede ejecutar código malicioso que se encuentre en la memoria de datos.

Desventajas:

1. Mayor costo: La arquitectura Harvard es más costosa de implementar que la arquitectura Von Neumann, debido a la necesidad de tener buses de datos separados.
2. Menor flexibilidad: La separación de los buses de datos limita la flexibilidad de la arquitectura Harvard, lo que la hace menos adecuada para aplicaciones que requieren una mayor flexibilidad en el manejo de los datos.

A continuación se muestra un gráfico de la arquitectura Harvard:



Arquitectura Von Neumann

La arquitectura Von Neumann utiliza un bus de datos único para las instrucciones y los datos, lo que significa que solo puede acceder a una fuente de datos a la vez. Las ventajas de esta arquitectura incluyen:

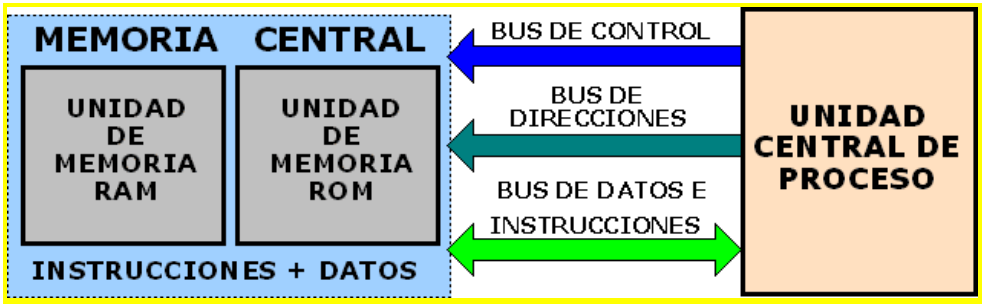
Ventajas:

1. Menor costo: La arquitectura Von Neumann es menos costosa de implementar que la arquitectura Harvard debido a que solo requiere un bus de datos.
2. Mayor flexibilidad: Al utilizar un bus de datos único, la arquitectura Von Neumann ofrece una mayor flexibilidad en el manejo de los datos, lo que la hace más adecuada para aplicaciones que requieren una mayor flexibilidad en el manejo de los datos.

Desventajas:

1. Menor velocidad: La arquitectura Von Neumann solo puede acceder a una fuente de datos a la vez, lo que reduce la velocidad de procesamiento en comparación con la arquitectura Harvard.
2. Mayor latencia: Al utilizar un bus de datos único, la arquitectura Von Neumann aumenta la latencia entre la memoria y la CPU, lo que reduce la velocidad de procesamiento.

A continuación se muestra un gráfico de la arquitectura Von Neumann:



2)¿Por qué se dice que los formatos de instrucción de la arquitectura Intel x86 son variables? De algún ejemplo que justifique su respuesta.

Los formatos de instrucción de la arquitectura Intel x86 se consideran variables porque pueden variar en tamaño y estructura, dependiendo del tipo de instrucción que se esté utilizando.

Por ejemplo, la instrucción ADD en la arquitectura x86 se puede utilizar de varias maneras, dependiendo de los operandos que se estén utilizando. Si se está sumando un registro y una constante de 8 bits, la instrucción tendría una estructura diferente a si se estuvieran sumando dos registros de 32 bits.

Además, los formatos de instrucción también pueden variar en tamaño dependiendo del modo de direccionamiento utilizado. Por ejemplo, la instrucción MOV puede tener una estructura diferente si se está moviendo un valor directamente a un registro, en comparación con si se está moviendo un valor desde una dirección de memoria indirecta.

En resumen, la arquitectura x86 permite una gran variedad de modos de direccionamiento y tipos de operandos, lo que resulta en una gran cantidad de posibles formatos de instrucción. Esta flexibilidad es una de las características clave de la arquitectura x86, pero también puede hacer que la programación en esta arquitectura sea más compleja en comparación con otras arquitecturas con formatos de instrucción más fijos y uniformes.

3)Explique claramente cómo funciona el linking estático. Ejemplifique y grafique dicho funcionamiento.

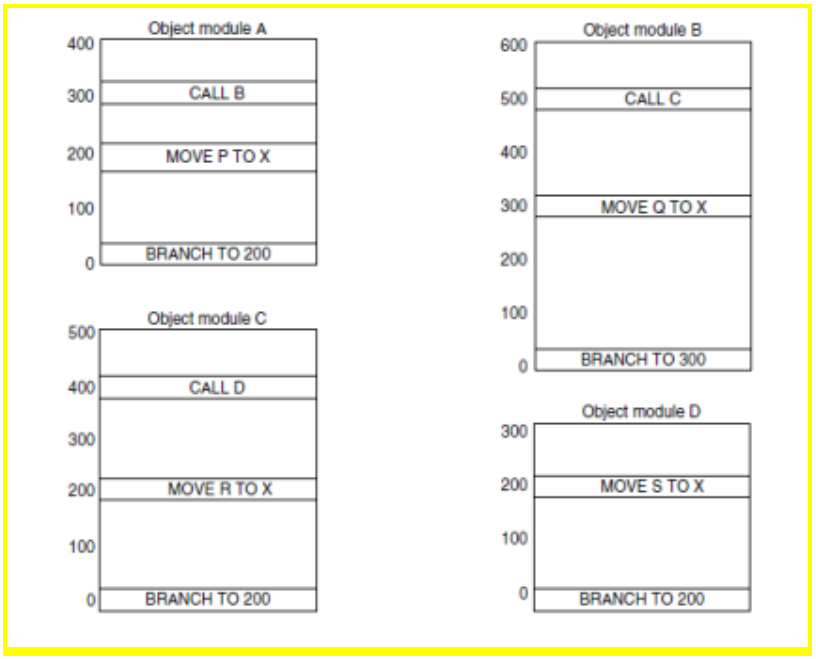
El linking estático es un proceso utilizado en el desarrollo de software que consiste en unir o combinar múltiples archivos objeto compilados en un solo archivo ejecutable. Este proceso se realiza durante la fase de compilación y enlazado del código fuente, y se utiliza para producir un archivo ejecutable independiente que no requiere de bibliotecas externas para funcionar.

Durante el proceso de compilación, el compilador genera archivos objeto que contienen el código compilado para cada módulo del programa. Estos archivos objeto contienen tanto el código de la aplicación como las referencias a las funciones y variables que se utilizan en la misma.

En el proceso de linking estático, el enlazador (linker) toma todos los archivos objeto y los combina en un solo archivo ejecutable. Durante este proceso, el enlazador resuelve todas las referencias a funciones y variables externas que se utilizan en la aplicación, buscando las definiciones correspondientes en los archivos objeto y en las bibliotecas estáticas (si las hay) para enlazarlas y producir un archivo ejecutable completo.

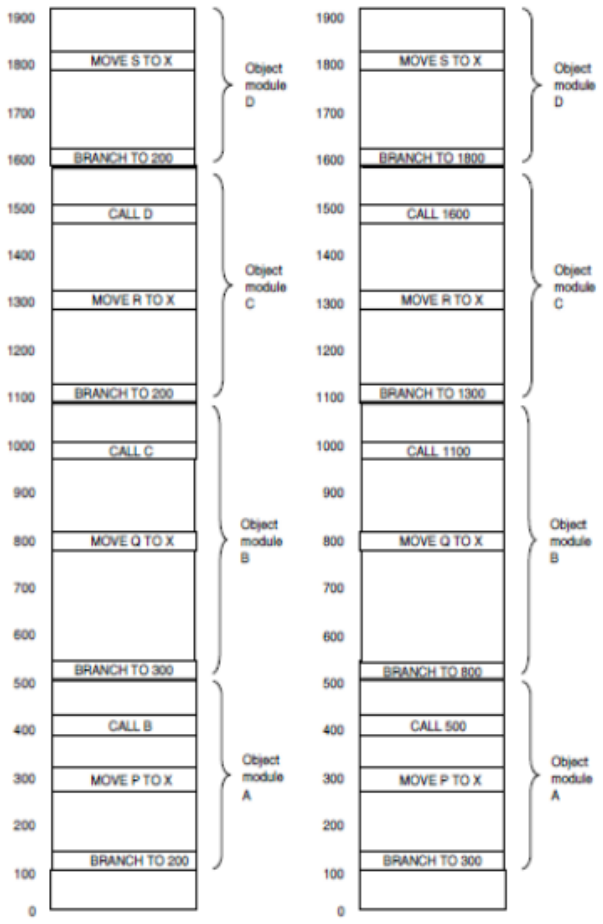
Una vez que se ha completado el linking estático, el archivo ejecutable resultante contiene todo el código necesario para ejecutar la aplicación, incluyendo todas las funciones y variables externas que se han enlazado. Esto hace que el archivo ejecutable sea independiente de las bibliotecas externas, lo que significa que se puede ejecutar en cualquier sistema que tenga el mismo sistema operativo y arquitectura.

Módulos objecto:



Generación del load module:

Module	Length	Starting address
A	400	100
B	600	500
C	500	1100
D	300	1600



4)Explique claramente que es un ciclo de instrucción en un procesador, indique qué etapas contempla y que ocurre durante la ejecución de cada una de ellas.

Un ciclo de instrucción es el proceso que sigue un procesador para ejecutar una instrucción. Cada instrucción que se ejecuta en un procesador sigue un ciclo de instrucción que consta de varias etapas que se realizan en secuencia. A continuación, se describen las etapas típicas de un ciclo de instrucción en un procesador moderno:

1. Obtención (Fetch): en esta etapa, el procesador busca la instrucción en la memoria principal o caché de instrucciones. La dirección de la siguiente instrucción a buscar se encuentra en el registro de programa (PC), que se incrementa para apuntar a la siguiente dirección de memoria después de cada ciclo de instrucción. La instrucción se almacena en el registro de instrucción (IR).
2. Decodificación (Decode): en esta etapa, el procesador interpreta la instrucción que se ha obtenido en la etapa anterior. La unidad de control del procesador lee el código de operación de la instrucción del registro de instrucción (IR) y determina qué operación debe realizar la instrucción.
3. Ejecución (Execute): en esta etapa, el procesador lleva a cabo la operación especificada por la instrucción. La operación puede implicar la lectura o escritura de datos de la memoria o de los registros, la realización de una operación aritmética o lógica, o la transferencia de datos entre registros.
4. Almacenamiento (Store): en esta etapa, el resultado de la operación se almacena en la memoria o en un registro. Si la instrucción implica escribir en memoria, el procesador utiliza la dirección de memoria que se ha calculado durante la etapa de ejecución para almacenar los datos en la memoria principal o caché de datos. Si la instrucción implica escribir en un registro, el resultado se almacena en el registro correspondiente.

Después de la etapa de almacenamiento, el ciclo de instrucción termina y el procesador se prepara para ejecutar la siguiente instrucción. El proceso se repite hasta que se completa la secuencia de instrucciones que se han cargado en la memoria o hasta que se produce una instrucción de salto o condicional que cambia la secuencia de instrucciones que se ejecutan.

Es importante destacar que, en algunos procesadores modernos, estas etapas se solapan o ejecutan simultáneamente en unidades de ejecución dedicadas, lo que permite que el procesador pueda ejecutar múltiples instrucciones en paralelo y aumentar así su rendimiento.

5)Identifique y explique cuáles son las principales desventajas del medio de almacenamiento en cinta.¿Cuáles son sus aplicaciones actuales? ¿Qué ventaja comparativa tiene con respecto al resto de los medios de almacenamiento secundario?

Las principales desventajas del medio de almacenamiento en cinta magnética son las siguientes:

1. Velocidad de acceso: La cinta magnética es un medio de almacenamiento secuencial, por lo que acceder a un archivo o dato específico puede llevar mucho tiempo. Además, las velocidades de transferencia de datos son más lentas en comparación con otros medios de almacenamiento como los discos duros o las unidades de estado sólido.
2. Capacidad limitada: Si bien las cintas magnéticas pueden almacenar grandes cantidades de datos, su capacidad de almacenamiento es limitada en comparación con otros medios de almacenamiento. Además, las cintas magnéticas más antiguas pueden tener una capacidad aún menor.
3. Fragilidad: Las cintas magnéticas son más susceptibles a daños físicos que otros medios de almacenamiento, lo que las hace menos confiables. Incluso una pequeña cantidad de polvo o suciedad puede afectar la calidad de los datos almacenados en la cinta.
4. Vida útil limitada: Las cintas magnéticas tienen una vida útil limitada y pueden experimentar degradación con el tiempo. Además, la exposición a campos magnéticos puede causar la pérdida de datos.

Las aplicaciones actuales de la cinta magnética incluyen la copia de seguridad y la archivación de grandes cantidades de datos, especialmente en entornos empresariales. También se utilizan en la producción de películas y programas de televisión como una forma de respaldo de los datos de grabación originales.

La principal ventaja comparativa de la cinta magnética en relación a otros medios de almacenamiento secundario, como los discos duros y las unidades de estado sólido, es su costo. La cinta magnética es una tecnología de almacenamiento más antigua y, por lo tanto, es más económica que otros medios de almacenamiento más modernos.

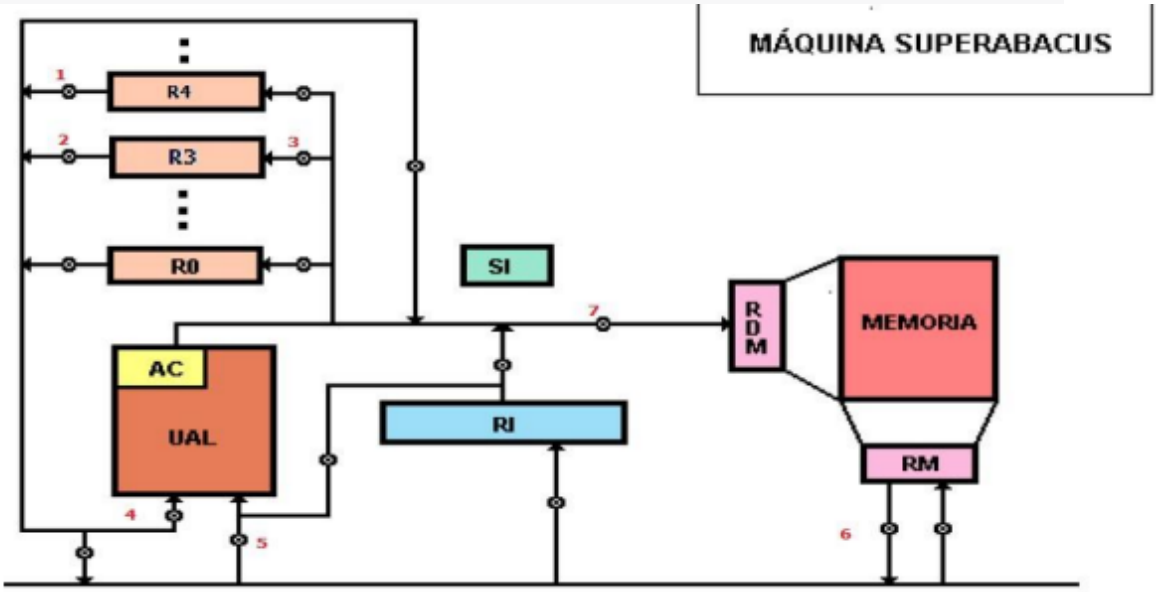
Además, las cintas magnéticas tienen la capacidad de almacenar grandes cantidades de datos, lo que las hace ideales para la copia de seguridad y la archivación a largo plazo de datos críticos y de archivo. Las cintas magnéticas también son menos vulnerables a la corrupción de datos debido a errores de escritura, lo que las hace ideales para la preservación a largo plazo de datos críticos y de archivo.

Sin embargo, es importante tener en cuenta que la velocidad de acceso a los datos en la cinta magnética es mucho más lenta que en los discos duros y las unidades de estado sólido, y la capacidad de almacenamiento de la cinta magnética es limitada en comparación con los discos duros y las unidades de estado sólido. Además, las cintas magnéticas son más susceptibles a daños físicos y tienen una vida útil limitada en comparación con otros medios de almacenamiento.

6)Indique cuales son las microinstrucciones necesarias para ejecutar la instrucción SUMAR 3,20(4) en la máquina SuperAbacus, siendo 3 y 4 registros de uso general y 20 un offset en base 10. Se pide además graficar en el esquema de la máquina el flujo de apertura de compuertas usadas en la fase de ejecución de dicha instrucción.

Para ejecutar la instrucción SUMAR 3,20(4) en la máquina SuperAbacus, se requieren las siguientes microinstrucciones:

1. Cargar el contenido del registro 3 en el acumulador.
2. Sumar el contenido del acumulador con el contenido de la dirección de memoria 20+contenido del registro 4.
3. Almacenar el resultado de la suma en el registro 3.



AC	←	(R4)	1,4
AC	←	(AC)+20	
RDM	←	(AC)	7
RM	←	((RDM))	
AC	←	(RM)	6,5
AC	←	(AC)+(R3)	2,4
R3	←	(AC)	3

7)Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo a la ubicación de los operandos. Ejemplifique y/o grafique cada una.

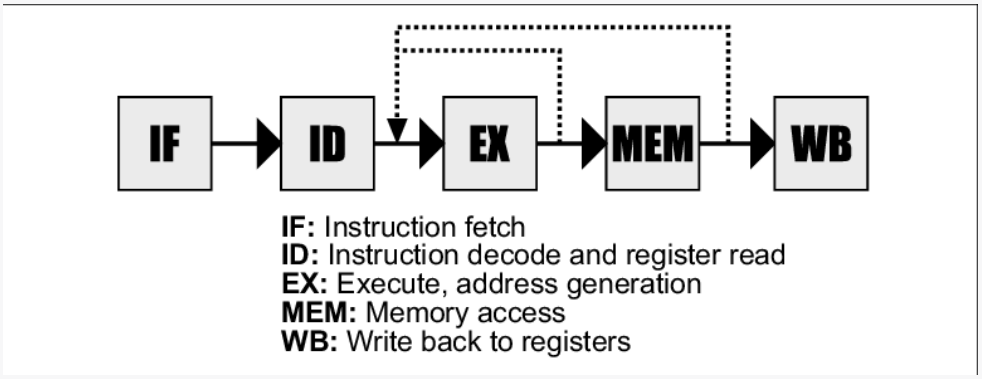
LA RESPUESTA ESTA EN RESULTADO DE FINAL

8)Explique claramente de qué se trata la técnica de procesamiento en paralelo por pipelining en un procesador. Qué complejidades pueden presentarse en el manejo de instrucciones y quienes pueden resolverlas.Grafique un pipeline de 5 stages.

La técnica de procesamiento en paralelo por pipelining en un procesador se refiere a la forma en que se dividen las instrucciones de un programa en etapas o "stages" para que puedan ser ejecutadas en paralelo. En lugar de esperar a que una instrucción termine de ejecutarse antes de comenzar con la siguiente, el procesador divide la ejecución de cada instrucción en varias etapas, que se realizan en paralelo con otras instrucciones en diferentes etapas del proceso.

Por ejemplo, un procesador con una arquitectura de pipeline de cinco stages puede dividir la ejecución de una instrucción en cinco etapas, como la búsqueda de la instrucción, decodificación de la instrucción, ejecución de la instrucción, acceso a memoria y escritura de los resultados. Cada instrucción en el programa pasaría por estas cinco etapas secuencialmente, pero varias instrucciones podrían estar en diferentes etapas al mismo tiempo, permitiendo que se ejecuten en paralelo.

La complejidad en el manejo de instrucciones que puede surgir con esta técnica radica en que algunas instrucciones pueden depender de los resultados de otras instrucciones que aún no han sido ejecutadas en etapas anteriores del pipeline. Si no se maneja adecuadamente, esto puede provocar errores y mal funcionamiento del procesador. Para resolver esta complejidad, se utilizan técnicas como el forwarding, que permite que los resultados de una instrucción se pasen a la siguiente etapa antes de que se hayan almacenado en la memoria, o la técnica de espera o "stalling", que detiene temporalmente el pipeline para esperar a que se completen las instrucciones anteriores antes de continuar con las instrucciones dependientes.



9)En la arquitectura de discos RAID de nivel 3: ¿qué ocurre con las peticiones de E/S mientras un disco queda inutilizable? Explique además el algoritmo que permite recuperar la información perdida.

En la arquitectura de discos RAID de nivel 3, cuando un disco queda inutilizable, el controlador RAID sigue recibiendo las solicitudes de E/S y las distribuye entre los discos restantes del conjunto. Como la paridad se distribuye en todos los discos, los bloques de datos y de paridad en los discos restantes pueden ser utilizados para reconstruir los datos del disco fallido. Durante la recuperación de datos, el rendimiento de E/S puede verse afectado, ya que se necesitan leer y escribir datos adicionales para reconstruir los bloques de datos que faltan. Sin embargo, una vez que se completa la recuperación de datos, el rendimiento vuelve a ser el normal. Es importante destacar que la velocidad de recuperación de datos depende del tamaño del conjunto de discos y de la carga de trabajo del sistema.

Cuando un disco falla en un conjunto RAID 3, se utiliza la paridad distribuida en los otros discos para recuperar los datos del disco fallido. El algoritmo para la recuperación de datos utiliza la siguiente fórmula para calcular el valor del bit de paridad faltante:

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

donde $X0$, $X1$, $X2$, $X3$ y $X4$ representan los bloques de datos y paridad almacenados en los discos. El símbolo "+" denota la operación "or exclusivo".

Por ejemplo, si el disco $X1$ falla, la fórmula anterior se utiliza para reconstruir los datos faltantes. Se suman los bloques de datos y de paridad correspondientes de los otros discos para obtener el valor del bloque de datos faltante en el disco $X1$.

Es importante destacar que la reconstrucción de los datos puede ser un proceso costoso en términos de tiempo y recursos del sistema, especialmente si se están recuperando grandes cantidades de datos o si hay un alto nivel de carga de trabajo en el sistema. Por lo tanto, es importante tener en cuenta las implicaciones de rendimiento de la recuperación de datos al seleccionar una configuración RAID adecuada para un sistema determinado.

10)Indique claramente que es el linking dinámico en tiempo de ejecución y cuáles son las diferencias frente al linking estático.

El linking dinámico en tiempo de ejecución es un proceso mediante el cual se vinculan bibliotecas compartidas con un programa en tiempo de ejecución, en lugar de hacerlo en tiempo de compilación como ocurre en el linking estático. En otras palabras, las bibliotecas se cargan en memoria cuando el programa se está ejecutando y se unen dinámicamente en lugar de vincularlas permanentemente en el momento de la compilación.

Las principales diferencias entre el linking dinámico y el estático son:

- En el linking dinámico, el programa no depende de las bibliotecas en el momento de la compilación, sino que las bibliotecas se cargan en tiempo de ejecución. En el linking estático, el programa depende de las bibliotecas en el momento de la compilación y se vinculan permanentemente.
- El tamaño del archivo ejecutable en el linking dinámico es más pequeño que en el estático, ya que el archivo ejecutable sólo contiene la información necesaria para cargar las bibliotecas compartidas en tiempo de ejecución. En el linking estático, el archivo ejecutable contiene toda la información necesaria para la ejecución del programa.
- El linking dinámico permite la actualización de las bibliotecas compartidas sin tener que volver a compilar el programa que las utiliza, lo que resulta en una mayor flexibilidad y facilidad de mantenimiento. En el linking estático, se requiere la recompilación del programa cada vez que se actualiza una biblioteca.

En general, el linking dinámico se utiliza cuando se desea reducir el tamaño del archivo ejecutable, permitir la actualización de bibliotecas compartidas y mejorar la flexibilidad y facilidad de mantenimiento. El linking estático se utiliza cuando se desea garantizar la compatibilidad con una versión específica de una biblioteca o cuando se desea una mayor velocidad de ejecución, ya que las bibliotecas se cargan en memoria en el momento de la vinculación.

11)Explique claramente por qué se dice que la UAL en SuperAbaus se utiliza tanto para la suma de datos como de direcciones. Justifíquelo con microinstrucciones y el gráfico de la máquina.

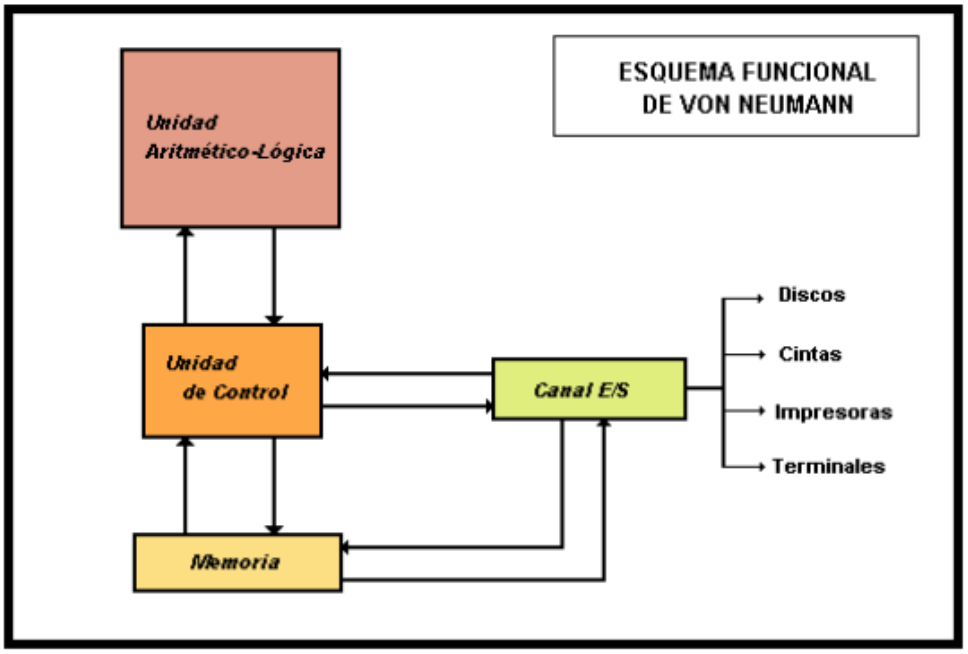
La Unidad Aritmético Lógica (UAL) es un componente fundamental de un procesador, encargado de realizar operaciones aritméticas y lógicas. En la arquitectura de la máquina SuperAbacus, la UAL se utiliza tanto para la suma de datos como para la suma de direcciones.

En la suma de datos, la UAL toma como entrada dos operandos que representan los datos que se quieren sumar, y produce como salida el resultado de la suma. Por ejemplo, si se quiere sumar el contenido de los registros R1 y R2, la UAL tomaría estos registros como entrada y produciría el resultado de la suma en un registro de salida.

En la suma de direcciones, la UAL se utiliza para calcular la dirección de memoria donde se encuentra el dato que se quiere acceder. En este caso, uno de los operandos que se le proporciona a la UAL es una dirección base y el otro es un offset que indica el desplazamiento desde la dirección base. La UAL realiza la suma de la dirección base y el offset para obtener la dirección final donde se encuentra el dato.

En la máquina SuperAbacus, el hecho de que la UAL pueda realizar tanto la suma de datos como la suma de direcciones se debe a que los operandos de la UAL son registros de uso general que pueden contener tanto datos como direcciones. Esto permite que la UAL pueda ser utilizada de manera eficiente para realizar tanto operaciones aritméticas como operaciones de acceso a memoria.

En términos de microinstrucciones, la UAL en SuperAbacus es programada para realizar tanto operaciones aritméticas como de suma de direcciones. En la microinstrucción que controla la UAL para la suma de datos, se especifican los registros de entrada y salida, así como la operación que se quiere realizar (suma, resta, etc.). En la microinstrucción que controla la UAL para la suma de direcciones, se especifican la dirección base y el offset que se quieren sumar, y la dirección resultante se almacena en un registro de salid



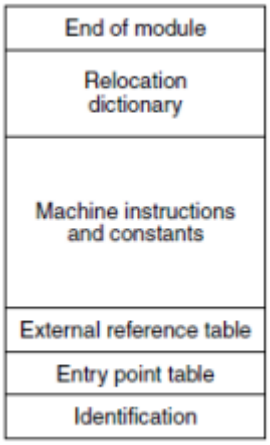
12)Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo al número de direcciones. Ejemplifique cada una.

Los cuatro tipos principales son:

1. 0 direcciones: Las instrucciones operan directamente en la pila de la CPU o stack. Estas instrucciones no requieren argumentos explícitos y suelen ser utilizadas para realizar operaciones aritméticas simples. Ejemplo: add TOS, TOS, Next.
2. 1 dirección: Las instrucciones utilizan un registro especial llamado acumulador y una dirección de memoria. La dirección de memoria se utiliza para acceder a un operando y el resultado se almacena en el acumulador. Ejemplo: add A, AC, Mem[A].
3. 2 direcciones: Las instrucciones utilizan dos registros de propósito general o una combinación de un registro y una dirección de memoria. El resultado se almacena en uno de los registros de propósito general. Ejemplo: add R1, A, R1 + Mem[A].
4. 3 direcciones: Las instrucciones utilizan tres registros de propósito general o una combinación de dos registros y una dirección de memoria. El resultado se almacena en uno de los registros de propósito general. Ejemplo: add R1, R2, R3, R1 = R2 + R3.

13)Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo al número de direcciones. Ejemplifique cada una.

1. Identificación: nombre del módulo, longitudes de las partes del módulo
2. Tabla de punto de entrada: lista de símbolos que pueden ser referenciados desde otros módulos
3. Tabla de referencias externas: lista de símbolos usados en el módulo, pero definidos fuera de él y sus referencias en el código
4. Código ensamblado y constantes
5. Diccionario de reubicabilidad: lista de direcciones a ser reubicadas
6. Fin de módulo



Ejemplo anterior:

1. header_section: información de cabecera sobre el archivo objeto
2. text_section: código del programa en formato de código de máquina
3. data_section: datos estáticos utilizados por el programa
4. relocation_section: información de reubicación
5. symbol_table_section: información sobre los símbolos definidos y utilizados en el programa

A pesar de las diferencias en la organización, ambas respuestas describen los diferentes componentes que componen un archivo de código objeto, como la información de identificación, las tablas de símbolos y referencias, el código y los datos del programa, y la información de reubicación.

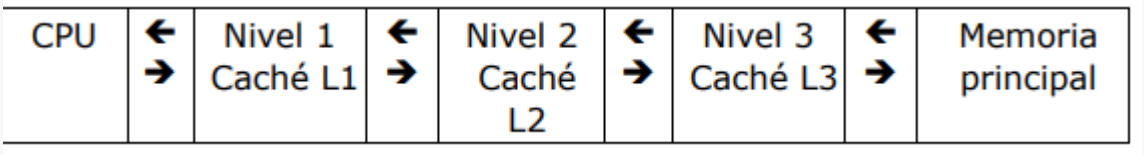
14)En un sistema de memoria, ¿qué función cumple la memoria cache? ¿En qué principio se basa su efectividad? Grafique un ejemplo de arquitectura de cache de 3 niveles.

La memoria caché es un tipo de memoria de acceso rápido que se encuentra entre la CPU y la memoria principal en un sistema de memoria. Su función principal es reducir el tiempo de acceso a los datos que se utilizan con mayor frecuencia en el sistema, lo que permite acelerar el rendimiento

de la CPU. La memoria caché almacena copias de datos y/o instrucciones que han sido recientemente utilizados o que se espera que se utilicen pronto.

La efectividad de la memoria caché se basa en el principio de localidad temporal y espacial. La localidad temporal se refiere a la tendencia de que los datos accedidos recientemente se vuelvan a acceder pronto, mientras que la localidad espacial se refiere a la tendencia de que los datos cercanos a los que se acceden se vuelvan a acceder pronto. La memoria caché aprovecha estos principios almacenando los datos que se acceden con mayor frecuencia en una memoria más rápida y cercana a la CPU que la memoria principal.

Cuando la CPU necesita acceder a un dato o una instrucción, primero busca en la memoria caché. Si el dato o la instrucción se encuentra en la memoria caché, se accede rápidamente. Si no se encuentra en la memoria caché, se debe acceder a la memoria principal, lo que es mucho más lento. La efectividad de la memoria caché depende de la frecuencia con la que se acceden los datos, la cantidad de datos que se almacenan en caché y la capacidad de la memoria caché.



14)¿Qué es la codificación 8-14 (EFM) y para qué se usa? ¿Por qué es necesaria?

La codificación 8-14 (EFM, del inglés "Eight-to-Fourteen Modulation") es una técnica utilizada en la grabación de CD (Compact Disc) y DVD (Digital Versatile Disc) para convertir los datos digitales en una secuencia de símbolos que pueden ser grabados en el disco.

La codificación 8-14 se utiliza para convertir los datos binarios en un conjunto de símbolos de 8 bits de longitud que se representan como una secuencia de 14 bits. Esto se hace para garantizar que la señal grabada en el disco tenga una transición de frecuencia suficientemente alta, lo que reduce el riesgo de errores durante la lectura del disco. La codificación 8-14 utiliza una tabla de asignación que convierte cada byte de datos de entrada en un símbolo de 14 bits de salida que cumple con ciertas reglas de transición de frecuencia y de sincronización.

La codificación 8-14 es necesaria debido a las limitaciones físicas del proceso de grabación de CD y DVD. La información se graba en la superficie del disco utilizando pequeñas marcas llamadas "pits" y "lands". La velocidad a la que se pueden grabar estos pits y lands es limitada por la física del proceso de grabación. Además, la longitud de onda del láser utilizado para leer la información en el disco es constante, lo que significa que no se puede aumentar la densidad de la información simplemente reduciendo el tamaño de los pits y lands.

La codificación 8-14 permite que la información se grabe en el disco de manera más eficiente, lo que aumenta la capacidad de almacenamiento y reduce el riesgo de errores de lectura. Además, es una técnica estándar utilizada en la grabación de CD y DVD, lo que significa que es compatible con la mayoría de los reproductores de discos existentes.

15)

- **[1,5 puntos] ¿Qué mecanismos provee el estándar IEEE 754 para el manejo de operaciones matemáticas con resultados indeterminados o indefinidos? De ejemplos de dichas operaciones e indique cual sería la configuración en el formato para representar dichos resultados.**

Valores no-numéricos: Denominados NaN (Not a number). Se identifican por un exponente con todos sus valores en 1, y un significando distinto de cero. Existen dos tipos de QNaN (Quiet NaN) y SNaN (Signalling NaN), que se distinguen dependiendo del valor 0/1 del bit más significativo del de la mantisa. QNaN tiene el primer bit en 1, y significa "Indeterminado". Resultado de todas aquellas operaciones aritméticas con resultados matemáticamente no definidos. SNaN tiene el primer bit en 0, y significa "operación no valida". Es la ejecución de una operación inválida. Ejemplos:

Operación	Resultado
Cualquier operación contra un NaN	NaN
+-0 /+-0	NaN
Infinito – Infinito	NaN
+-Infinito / Infinito+-	NaN
+-Infinito X 0	NaN

	Signo	Exponente en exceso	Mantisa
QNaN	0/1	11111111	**1**01010101001010101101
SNaN	0/1	11111111	**0**10101011101010101010

16)Enuncie al menos 4 características de la arquitectura de procesadores CISC .De ejemplo de esas características en dicha arquitectura.

CISC (Complex Instruction Set Computer) es una arquitectura de procesadores que se caracteriza por tener un conjunto de instrucciones complejas y variadas. A continuación, se enuncian al menos 4 características de la arquitectura de procesadores CISC junto con un ejemplo de cada una:

1. Conjunto de instrucciones amplio y complejo: los procesadores CISC tienen un conjunto de instrucciones amplio y complejo que pueden realizar múltiples tareas en una sola instrucción. Por ejemplo, la instrucción "MOV" en el procesador x86 puede mover datos entre registros, memoria y puertos de E/S.
2. Modos de direccionamiento complejos: los procesadores CISC tienen modos de direccionamiento complejos que permiten acceder a diferentes tipos de datos en memoria.

Por ejemplo, en el procesador x86, se pueden usar modos de direccionamiento como indirecto, indexado y basado en registro.
3. Uso intensivo de la memoria: los procesadores CISC realizan un uso intensivo de la memoria para almacenar sus instrucciones y datos. Por ejemplo, el procesador x86 utiliza una gran cantidad de memoria para almacenar su complejo conjunto de instrucciones.
4. Instrucciones de longitud variable: los procesadores CISC tienen instrucciones de longitud variable que pueden tener diferentes longitudes dependiendo de los operandos involucrados. Por ejemplo, en el procesador x86, las instrucciones pueden tener entre 1 y 15 bytes de longitud, lo que hace que la decodificación de las instrucciones sea más compleja.

17)Indique claramente cuales son las acciones que realiza un ensamblador específicamente en la primera pasada de proceso de ensamblado.¿qué información genera como salida dicho proceso y para que se usará?

Durante la primera pasada del proceso de ensamblado, el ensamblador realiza las siguientes acciones:

1. Lectura del código fuente: El ensamblador lee el código fuente escrito en lenguaje ensamblador.
2. Análisis sintáctico: El ensamblador analiza la sintaxis del código fuente para detectar errores de gramática y sintaxis. También se encarga de identificar y procesar los pseudo-instrucciones y directivas que no generan código máquina.
3. Generación de la tabla de símbolos: Durante la primera pasada, el ensamblador genera una tabla de símbolos que contiene los nombres y ubicaciones de las etiquetas y símbolos definidos en el código fuente.
4. Asignación de direcciones: El ensamblador asigna direcciones a las etiquetas y símbolos definidos en el código fuente. Estas direcciones se utilizan para generar referencias de salto y para la resolución de símbolos externos en la segunda pasada del ensamblado.
5. Generación del archivo de salida: Como resultado de la primera pasada, el ensamblador genera un archivo objeto que contiene el código máquina generado a partir del código fuente. Este archivo también incluye la tabla de símbolos y la asignación de direcciones.

La información generada durante la primera pasada se utiliza como entrada en la segunda pasada del ensamblado, donde se generará el archivo final ejecutable. La tabla de símbolos es utilizada para la resolución de símbolos externos y para la generación de referencias de salto. La asignación de direcciones es utilizada para generar referencias de salto y para la resolución de símbolos externos.

18)Explique claramente que ventajas aporta la tecnica de E/S por DMA frente a otras mas limitadas.Grafique algunas posibles topologias de configuracion.

La técnica de E/S por DMA (Direct Memory Access) aporta varias ventajas frente a otras técnicas de E/S más limitadas, tales como la E/S programada o la E/S por interrupciones:

1. Mayor velocidad de transferencia: Con la técnica de DMA, la transferencia de datos se realiza directamente entre los dispositivos de E/S y la memoria principal, sin intervención del

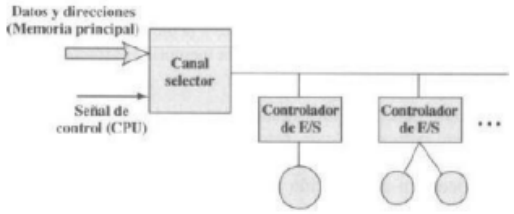
procesador. Esto permite una transferencia de datos mucho más rápida, ya que el procesador no tiene que esperar a que se complete cada transferencia de datos.

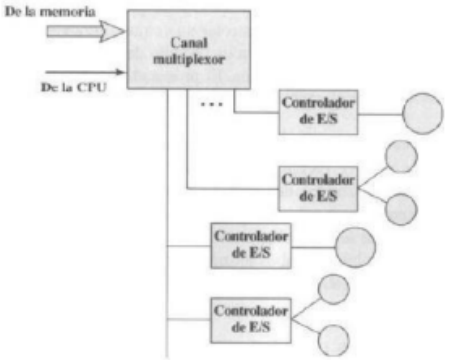
2. Liberación del procesador: Al realizar las transferencias de datos directamente entre los dispositivos de E/S y la memoria principal, el procesador queda liberado para realizar otras tareas. Esto aumenta la eficiencia del procesador y permite que se realicen más operaciones en menos tiempo.
3. Reducción del tráfico en el bus: Al utilizar la técnica de DMA, se reduce el tráfico en el bus de sistema, ya que el procesador no tiene que realizar todas las transferencias de datos. Esto permite una mayor cantidad de dispositivos de E/S y reduce la posibilidad de congestión en el bus.
4. Flexibilidad: La técnica de DMA permite una mayor flexibilidad en el diseño de sistemas de E/S, ya que se pueden utilizar diferentes topologías de configuración, según las necesidades del sistema.

El canal de E/S representa una aplicación del concepto 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.

Son comunes dos tipos

- Un canal selector
  - Que controla varios dispositivos de velocidad elevada. El canal selecciona un dispositivo y efectúa la transferencia de datos. Cada dispositivo, o pequeño grupo de dispositivos, es manejado por un controlador.
- Canal multiplexor
  - Puede manejar E/S de varios dispositivos al mismo tiempo. Para dispositivos de velocidad reducida.





19)cual es la finalidad de la exixtencia de los numeros desnormalizados en el formato de punto flotante IEEE 754?grafique todos los valores extremos del rango de numeros desnormalizados y de sus configuraciones hexadecimales en el formato.

La existencia de números desnormalizados en el formato de punto flotante IEEE 754 permite representar valores muy pequeños que están por debajo del límite mínimo de normalización (llamado "épsilon de máquina"), que de otra manera se representarían como cero. Estos números desnormalizados se conocen como "subnormales" o "denormalizados" y son útiles en ciertas aplicaciones científicas y

de ingeniería donde se requiere una precisión extrema y se necesitan valores muy pequeños.

En el formato de punto flotante IEEE 754, los números desnormalizados tienen una mantisa que no tiene el bit implícito (es decir, el bit que se omite en los números normalizados) y tienen un exponente que es menor que el valor mínimo normalizado. Esto significa que los números desnormalizados tienen una menor precisión que los números normalizados y que su rango es muy limitado.

Los valores extremos del rango de números desnormalizados se muestran en la siguiente tabla:

Mantisa	Exponente	Valor en base 10	Valor en hexadecimal
0	-126	1.4013 × 10^-45	0x0000000000000001
1/2^23	-126	1.1755 × 10^-38	0x000000007FFFFF
1 - 1/2^23	-126	1.1755 × 10^-38	0x0000000080000000
1	-125	2.8026 × 10^-45	0x0000000080000001
1 - 1/2^23	-149	1.4013 × 10^-45	0x0000000000000000

Es importante destacar que los números desnormalizados pueden tener un efecto en el rendimiento, ya que su procesamiento es más lento debido a que requieren más operaciones de redondeo y ajuste. Sin embargo, su existencia es necesaria para lograr una representación precisa de valores extremadamente pequeños.

20)

- **[1 punto] Indique como se puede clasificar el repertor instrucciones de una arquitectura de computadores de acuerdo c número de direcciones. Ejemplifique y/o grafique cada uno.**

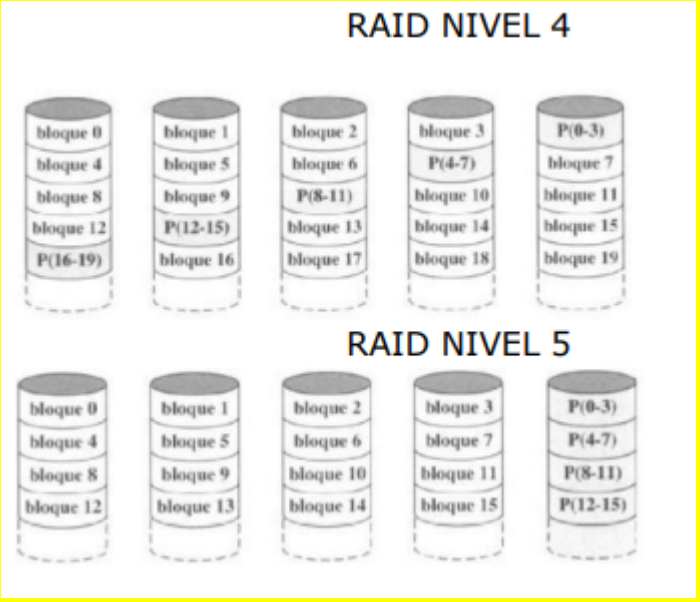
El conjunto de

1. 0 direcciones (Stack)
   Ejemplo: add — TOS ← TOS + Next
2. 1 dirección (Acumulador)
   Ejemplo: add A — AC ← AC + Mem[A]
3. 2 direcciones (Reg-Mem/Reg-Reg/Mem-Mem)
   Ejemplo: add R1, A — R1 ← R1 + Mem[A]
4. 3 direcciones (Reg/Mem)
   Ejemplo: add R1, R2, R3 — R1 ← R2 + R3

21)] ¿Cuáles son las ventajas del nivel 5 de la arquitectura de discos RAID con respecto al nivel 4? Grafique la distribución de la información en los discos en ambos niveles.

Ventajas - Distribuye las tiras de paridad a lo largo de todos los discos. Lo cual evite un potencial cuello de botella de E/S encontrado en el RAID 4.

Desventaja - Controlador sumamente más complejo que el de su nivel inferior.



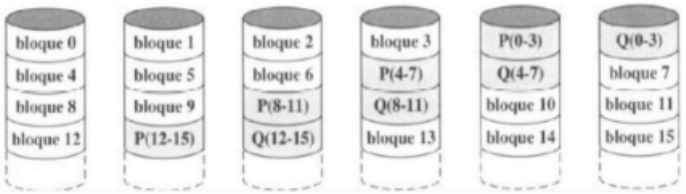
22)

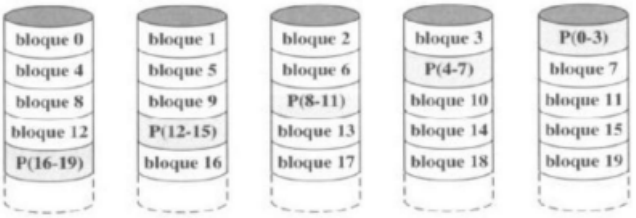
- **[1,5 puntos] ¿Cuáles son las ventajas y desventajas del nivel 6 de la arquitectura de discos RAID respecto al nivel 5? Grafique la distribución de la información en los discos de ambos niveles.**

RAID 6

Ventajas:
- Proporciona una tolerancia a fallos mayor.
- Puede soportar la rotura de hasta 2 unidades simultáneamente.

Desventajas:
- Diseño del más complejo.
- Más costoso ya que se necesita un disco más (en nivel 5, n discos de datos +1 de redundancia; en nivel 6, n discos de datos +2 de redundancia).





RAID 5

22)Explica claramente de que se trata la técnica de procesamiento en paralelo Superscalar en un procesador.Grafique en forma esquemática cómo funciona dicha técnica y de un ejemplo de algún procesador comercial que la incluya.

La técnica de procesamiento en paralelo Superscalar en un procesador es una técnica que permite la ejecución de varias instrucciones al mismo tiempo. Para lograr esto, se utilizan varias técnicas avanzadas que permiten que las instrucciones se ejecuten en paralelo sin afectar la correctitud del resultado final.

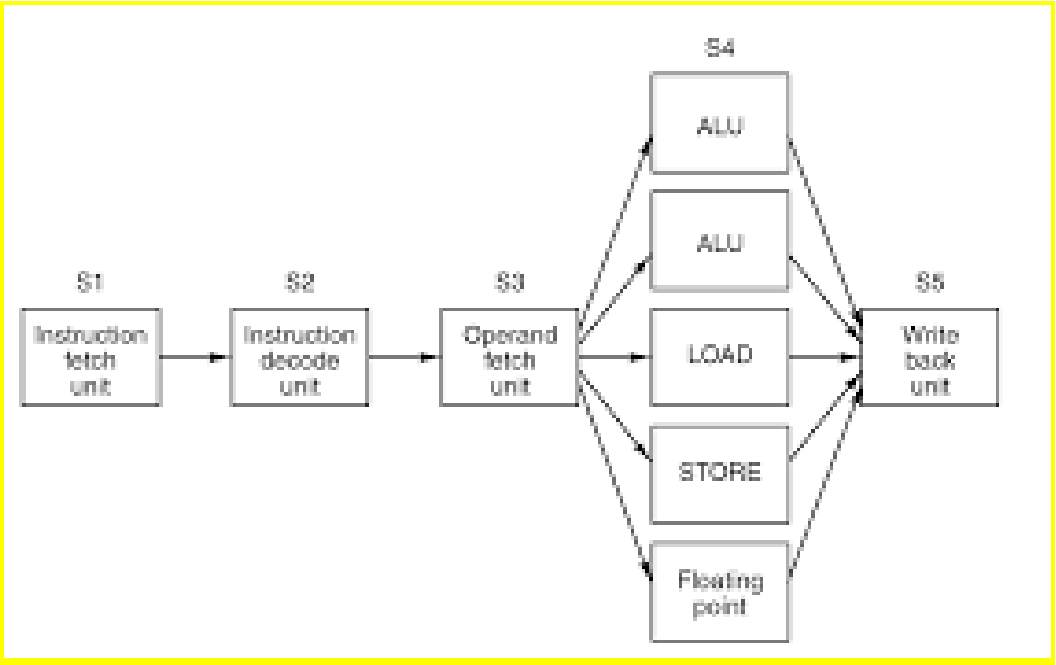
La primera técnica utilizada en un procesador Superscalar es la emisión de múltiples instrucciones. En lugar de emitir una sola instrucción en cada ciclo de reloj, un procesador Superscalar es capaz de emitir varias instrucciones al mismo tiempo. Las instrucciones se emiten a un conjunto de unidades de ejecución que son capaces de procesar diferentes tipos de instrucciones al mismo tiempo.

La siguiente técnica utilizada es la renumeración de registros. En lugar de depender de un conjunto de registros limitados, los procesadores Superscalar utilizan una técnica de renombrado de registros que les permite asignar temporalmente registros virtuales a las instrucciones. Esto permite que las instrucciones se ejecuten en paralelo sin afectar la correctitud del resultado final.

Hay varios procesadores comerciales que utilizan la técnica de procesamiento en paralelo Superscalar, incluyendo:

1. Intel Core i7 y i9: Estos procesadores de gama alta tienen múltiples núcleos y utilizan la técnica de procesamiento en paralelo Superscalar para mejorar el rendimiento de la CPU.

2. AMD Ryzen: Los procesadores Ryzen de AMD utilizan la técnica de procesamiento en paralelo Superscalar para mejorar el rendimiento de la CPU y ofrecer un mayor número de hilos.
3. IBM Power9: El procesador Power9 de IBM utiliza la técnica de procesamiento en paralelo Superscalar para mejorar el rendimiento y la capacidad de procesamiento en sistemas de alta gama, como servidores y supercomputadoras.



23)explique claramente que significan los terminos big y little endian ,en que contexto se aplican y que los diferencian.De ejemplo de arquitecturas en donde se use cada uno.

Los términos "big endian" y "little endian" se refieren a la forma en que se almacenan los datos en la memoria de una computadora. Estos términos son comúnmente utilizados en arquitectura de computadoras, lenguajes de programación y protocolos de comunicación.

En la notación "big endian", los bytes más significativos se almacenan en la dirección de memoria más baja, mientras que en la notación "little endian", los bytes menos significativos se almacenan en la dirección de memoria más baja.

Por ejemplo, en el número hexadecimal 0x12345678, si se utiliza la notación big endian, el byte más significativo sería 0x12 y el byte menos significativo sería 0x78. Si se utiliza la notación little endian, el byte más significativo sería 0x78 y el byte menos significativo sería 0x12.

La notación big endian se utiliza comúnmente en arquitecturas de procesadores como PowerPC y SPARC, mientras que la notación little endian se utiliza en arquitecturas de procesadores como x86 y ARM.

Un ejemplo de una arquitectura que utiliza big endian es el procesador PowerPC de IBM, que se utiliza en algunos sistemas embebidos, servidores y supercomputadoras. Por otro lado, un ejemplo de una arquitectura que utiliza little endian es el procesador x86 de Intel, que se utiliza en la mayoría de los PC de escritorio y portátiles.

Es importante tener en cuenta que el uso de big endian o little endian no afecta la funcionalidad del sistema, sino que es una elección de diseño que puede tener impactos en el rendimiento, la facilidad de programación y la compatibilidad con otros sistemas.

23)Que ventajas presenta el modo de direccionamiento por desplazamiento relativo al PC (referencia al programa)frente al direccionamiento directo?

La ventaja del direccionamiento por desplazamiento relativo al PC. Tanto el direccionamiento por referencia al programa como el por desplazamiento relativo al PC permiten la reubicación de los programas en la memoria sin tener que cambiar las direcciones de los datos. En el caso del direccionamiento por referencia al programa, la dirección de la instrucción es la referencia utilizada, mientras que en el direccionamiento por desplazamiento relativo al PC se utiliza la dirección del programa (PC). En ambos casos, esto hace que los programas sean más flexibles y portátiles y puede ayudar a ahorrar memoria en el sistema. Además, como mencioné anteriormente, el direccionamiento por desplazamiento relativo al PC también puede mejorar la seguridad del sistema al restringir el acceso a direcciones arbitrarias en la memoria.

24)Loading

Claro, el término "loading" se refiere al proceso de cargar un programa o datos en la memoria de un ordenador. Aquí te explico algunos tipos de loading que se utilizan comúnmente:

1. Carga estática: este tipo de carga implica que el programa completo se carga en la memoria antes de su ejecución. Esto puede ser beneficioso porque el programa se ejecutará más rápidamente ya que no se requiere tiempo adicional para cargar datos o instrucciones adicionales. Sin embargo, la carga estática puede requerir más memoria y puede ser menos flexible que otros tipos de carga.
2. Carga dinámica: en este tipo de carga, los datos y las instrucciones se cargan en la memoria a medida que son necesarios durante la ejecución del programa. Esto puede ahorrar memoria y hacer que los programas sean más flexibles, ya que solo se cargan los datos e instrucciones necesarios en un momento dado. Sin embargo, la carga dinámica puede ser

más lenta que la carga estática, ya que se requiere tiempo adicional para cargar los datos e instrucciones necesarios en la memoria.

3. Carga perezosa (lazy loading): este tipo de carga es una variante de la carga dinámica en la que los datos y las instrucciones se cargan solo cuando se necesitan por primera vez durante la ejecución del programa. Esto puede ahorrar aún más memoria y hacer que los programas sean aún más flexibles, ya que solo se cargan los datos e instrucciones necesarios en un momento dado. La carga perezosa es común en lenguajes de programación como Python y Ruby.

25)Explique claramente de qué se trata la técnica de procesamiento en paralelo multithreading en un procesador.Explique sintéticamente la diferencia entre un thread y un proceso y como es el cambio de contexto en un caso y otro.

La técnica de procesamiento en paralelo multithreading en un procesador implica la ejecución simultánea de varios subprocesos o "threads" en un solo núcleo de procesamiento del procesador. Esta técnica permite que el procesador realice múltiples tareas en paralelo, lo que puede mejorar significativamente el rendimiento y la eficiencia del procesador.

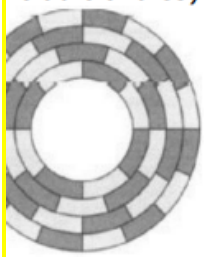
En cuanto a la diferencia entre un thread y un proceso, un proceso es una instancia única de un programa en ejecución, mientras que un thread es una forma de subdividir un proceso en unidades de ejecución más pequeñas que comparten recursos. Los procesos tienen su propio espacio de memoria y recursos del sistema asignados, mientras que los threads comparten los recursos del proceso padre, pero tienen su propia pila y contexto de registro.

En cuanto al cambio de contexto, el cambio de contexto en un proceso implica guardar y cargar grandes cantidades de datos de memoria, mientras que el cambio de contexto en un thread solo implica el cambio del contexto de registro y la pila del thread. Por lo tanto, el cambio de contexto en un thread es menos costoso y se puede realizar con mayor frecuencia que el cambio de contexto en un proceso.

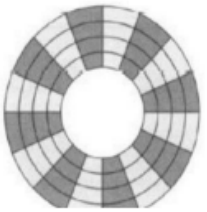
En resumen, el procesamiento en paralelo multithreading en un procesador implica la ejecución simultánea de varios subprocesos en un solo núcleo de procesamiento, lo que mejora la eficiencia y el rendimiento del procesador. Un thread es una unidad de ejecución más pequeña que comparte recursos con el proceso padre, mientras que un proceso es una instancia única de un programa en ejecución con su propio espacio de memoria y recursos del sistema asignados. El cambio de contexto en un thread es menos costoso que el cambio de contexto en un proceso, lo que permite cambios de contexto más frecuentes y eficientes.

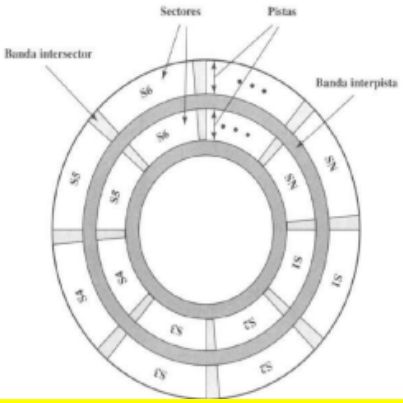
25)ME PODRÍAS EXPLICAR LAS VENTAJAS Y DESVENTAJAS DE LA ORGANIZACION TRADICIONAL DE LOS DISCOS MAGNÉTICOS VERSUS LA ORGANIZACION MULTIZONA.

Ventajas:
Mayor capacidad de almacenamiento. En los discos tradicionales se tenían igual cantidad de bits en los sectores más alejados del centro y en los más cercanos. Esto era así debido a que se leía a una velocidad angular constante. En los discos multi zona esto cambia, y permite mayor densidad de bits en las zonas más alejadas. 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 (en los discos tradicionales).



Desventajas:
La mayor capacidad de almacenamiento viene acompañada de una circuitería más compleja y es más difícil situar a la cabeza en la zona que deseamos, ya que en el método tradicional esto se puede direccionar directamente con la pista y el sector.





En resumen, la organización tradicional de los discos magnéticos es simple y fácil de implementar, pero puede tener limitaciones en el rendimiento y consistencia. La organización multizona puede proporcionar una mejor gestión de las zonas y un rendimiento más consistente, pero requiere una gestión más compleja y puede tener limitaciones en el rendimiento si los datos se encuentran en diferentes zonas con diferentes características de rendimiento.

26)Indique claramente como es el funcionamiento del modo de direccionamiento por registro base y desplazamiento.Dame un ejemplo de su uso en una arquitectura.

El modo de direccionamiento por registro base y desplazamiento se utiliza para acceder a una dirección de memoria sumando un valor de desplazamiento a un valor almacenado en un registro base. Este modo de direccionamiento es muy útil para acceder a estructuras de datos almacenadas en memoria, ya que permite la indexación de datos utilizando un puntero y un desplazamiento.

El funcionamiento del modo de direccionamiento por registro base y desplazamiento implica los siguientes pasos:

1. Se carga la dirección base de la estructura de datos en un registro base.
2. Se carga el desplazamiento requerido en otro registro o como parte de la instrucción.
3. Se realiza la operación de suma del registro base y el registro de desplazamiento.
4. Se accede a la dirección de memoria resultante.

Por ejemplo, en una arquitectura x86, se podría utilizar el modo de direccionamiento por registro base y desplazamiento para acceder a un elemento de un arreglo. Si se tiene un arreglo de enteros

almacenado en memoria y se quiere acceder al elemento en la posición 3, se podría utilizar el registro ESI como registro base y el valor 3 como desplazamiento. La instrucción podría ser algo como:

```
mov eax, [esi+3*4]
```

Esta instrucción carga en el registro EAX el valor almacenado en la dirección base almacenada en ESI sumado con el desplazamiento de 3 enteros multiplicado por 4 bytes (tamaño de un entero en la arquitectura x86). Esto permitiría acceder al elemento del arreglo en la posición 3.

27)Nombra y explica tren funciones que cumplen los módulos de E/S para la interconexión de periférico con el resto del sistema.

Los módulos de E/S (entrada/salida) tienen como función principal la interconexión de los periféricos con el resto del sistema. A continuación, se explican tres funciones importantes que cumplen estos módulos:

1. Control de interrupciones: los módulos de E/S son responsables de controlar las interrupciones generadas por los periféricos para que puedan ser atendidas adecuadamente por el procesador. Los módulos de E/S pueden generar una señal de interrupción al procesador para notificar la presencia de una solicitud de interrupción, y también pueden proporcionar los registros necesarios para almacenar la información relevante de la interrupción, como la dirección del dispositivo que la generó y el tipo de solicitud.
2. Gestión de la memoria del búfer: los módulos de E/S también deben administrar el búfer de entrada/salida para garantizar que los datos se transmitan correctamente entre el periférico y el procesador. Esto puede implicar la implementación de técnicas de control de flujo, como la espera activa, la interrupción de transmisión y la administración de colas para garantizar que la transferencia de datos se realice de manera segura y sin pérdida de información.
3. Conversión de señales: en algunos casos, los módulos de E/S también pueden ser responsables de la conversión de señales entre el formato requerido por el periférico y el formato utilizado por el sistema. Esto puede implicar la conversión de señales analógicas a digitales o la conversión de señales de un protocolo de comunicación específico a otro.

En resumen, los módulos de E/S son fundamentales para garantizar la correcta interconexión de los periféricos con el resto del sistema. Además de las tres funciones mencionadas, también pueden cumplir otras tareas importantes, como el control de la energía y la velocidad del reloj de los periféricos, la gestión de las interrupciones de DMA y la implementación de protocolos de comunicación específicos para cada periférico.