Resumen Arquitectura de Computadoras

**Que es un Bus?**

Un bus es un camino de comunicación entre dos o mas dispositivos. Un bus se usa como medio de transmisión compartido, puede ser transmisión de datos, direcciones, señales de control entre diferentes componentes del sistema, como la CPU, la memoria y dispositivos de e/s.

Al bus se conectan varios dispositivos y cualquier señal emitida por un dispositivo está disponible para que los demás dispositivos conectados puedan accederla.

Solo un dispositivo puede transmitir con éxito en un momento dado.

**Posibles conflictos en un bus:**

* Tener varios dispositivos conectados al mismo bus puede producir retardos de propagación.
* Si dos dispositivos transmiten durante el mismo periodo de tiempo, sus señales pueden distorsionarse.

El bus permite la transferencia de datos dentro del sistema.

La velocidad y la capacidad del bus pueden tener un impacto en el rendimiento del sistema.

***Estructura de un bus: Un bus está constituido por:***

**Líneas de datos**: Proporcionan un camino para transmitir datos entre los módulos del sistema. El conjunto constituido por estas líneas se denomina bus de datos.

Cada línea puede transportar de a 1 bit a la vez. El número de líneas determina cuantos bits se pueden transferir en ese ciclo de reloj.

**Líneas de dirección:** Se utilizan para designar fuente o destino del dato.

Además las líneas de direcciones se utilizan para direccionar los puertos de e/s.

**Líneas de control:** Se utilizan para controlar el acceso y el uso de las líneas de datos y controles. Las señales de control transmiten ordenes como información de temporización entre los modulos del sistema.

***Los aspectos clave del diseño de los buses son:***

* El arbitraje (si el permiso para enviar señales se controla de forma distribuida o centralizada).
* La temporización (si las señales del bus se sincronizan mediante un reloj comun o se enviar asincrónicamente
* Anchura del bus (cantidad de datos que puede transmitir, en base a las líneas de dirección y de datos)

Ancho de banda: Se refiere a la cantidad de datos que puede transmitir en un periodo de tiempo. Un bus de mayor ancho de banda permite una transferencia de datos más rápida.

Velocidad del bus: Indica la velocidad con la que los datos pueden ser transmitidos a lo largo del bus.

Líneas de control: Se utilizan para controlar el acceso y el uso de las líneas de datos y direcciones- Las señales de control transmiten órdenes como información de temporización entre los módulos del sistema.

**Funcionamiento del bus:**

Si un modulo desea enviar un dato a otro debe hacer dos cosas:

(1) obtener el uso del bus (2) transferir datos a través del bus.

Si un modulo desea pedir datos a otro modulo debe:

(1) obtener el uso del bus (2) transferir la petición al otro modulo mediante líneas de control y dirección.

Tipo de bus: Existen varios tipos de buses, como el bus de datos, bus de direcciones y bus de control. Cada uno tiene una función especifica en la transferencia de información.

**Jerarquia de buses:**

Cuando se conecta un gran número de dispositivos a un bus, las prestaciones pueden disminuir debido a:

* A medida que aumenta el número de dispositivos, el retardo de propagación también aumenta.
* A medida que las solicitudes de transferencia se acercan a su capacidad, el bus puede convertirse en un cuello de botella.

Se pueden incrementar las prestaciones mejorando la anchura y la velocidad del bus.

Buses Sincrónicos:

* La transmisión de datos esta sincronizada con una señal de reloj común para todos los dispositivos conectados al bus.
* Los datos se transmiten en intervalos determinados por un pulso de reloj.
* Todos los dispositivos conectados al bus deben operar a la misma frecuencia para evitar problemas de sincronización.

Buses Asincrónicos:

* Cada dispositivo puede operar a su propio ritmo.
* Los datos se envían de manera independiente, sin depender de una señal de reloj.
* Son adecuados cuando se están utilizando dispositivos con diferentes frecuencias de reloj

**Sistemas de Arbitraje:** Ya que en un instante dado solo una unidad puede transmitir a través del bus, se requiere algún método de arbitraje.

* Arbitraje centralizado: Un único dispositivo *conocido como controlador del bus* es el encargado de asignar tiempo del bus a cada dispositivo
* Arbitraje Distribuido: Cada modulo dispone de lógica para controlar el acceso. Y los modulos actúan conjuntamente para compartir el bus.

Tipos de Buses:

* Buses Dedicados: Es un bus dedicado en un canal en especifico. Se encarga de una función o dispositivo en particular. Por ejemplo, un bus dedicado podría ser solo para la transferencia de datos entre la CPU y la memoria principal
* Buses Multiplexados: Es un bus que comparte varias funciones o dispositivos. Transmite diferentes tipos e información en el mismo conjunto de líneas de comunicación. Puede encargarse de transmitir datos, señales de control, direcciones, etc, en el mismo conjunto de líneas.

**Que es la segmentación de cauce?**

Es una técnica en la que las instrucciones son divididas en etapas.Múltiples instrucciones se ejecutan simultáneamente en diferentes etapas, cada etapa es ejecutada por una unidad de hardware. Es una técnica usada para mejorar el rendimiento y la eficiencia del procesamiento de instrucciones.

La segmentación de cause se basa en el paralelismo a nivel de instrucciones, puede ejecutar varias instrucciones simultáneamente. Es decir, para ejecutar otra instrucción, no necesariamente debe esperar a que la ejecución de la instrucción en ejecución termine. De esta manera puede ejecutar instrucciones paralelamente.

Con la posibilidad de ejecutar instrucciones paralelamente surgen diferentes conflictos de dependencia:

**Conflictos de Dependencia de Control:** Surgen cuando la ejecución de una instrucción **depende de cómo se ejecute otra. Ejemplo: 1 salto y 2 posibles caminos.**

**Conflictos de Dependencia de Datos:** Surgen cuando dos o mas instrucciones comparten el mismo dato, y una instrucción necesita el resultado de otra que aun no esta disponible.

**Conflictos Estructurales:** Surgen cuando dos o mas partes del hardware compiten por el mismo recurso, ejemplo: Memoria, ALU y Registros.

**Conceptos de maquinas CISC y RISC:**

**RISC vs CISC:**

Los procesadores RISC se caracterizan por tener un conjunto de instrucciones reducido y simples. Los procesadores RISC tienden a utilizar un enfoque de pipeline, dividiendo la ejecución de instrucciones en etapas para logar un rendimiento optimo.

Los procesadores CISC tienen un conjunto de instrucciones mas extenso y complejo. Permite realizar tareas mas avanzadas en una sola instrucción. Cada instrucción puede requerir mas de un ciclo de reloj. Los CISC se utilizan para instrucciones mas complejas.

**Diferencias entre RISC y CISC:**

**RISC tiene un conjunto de instrucciones mas simple y reducido, mientras que el CISC es mas complejo y extenso. Los RISC tienden a utilizar un enfoque pipeline, lo que hace énfasis en el paralelismo.**

|  |  |
| --- | --- |
| CISC | RISC |
| Se utiliza para instrucciones más complejas.  Tienen un conjunto de instrucciones más extenso y completo  Permite realizar tareas más avanzadas en una sola instrucción.  Cada instrucción puede requerir mas de un ciclo de reloj.  Hace énfasis en la ejecución secuencial.  Produce programas mas pequeños y rapidos | Se utiliza para instrucciones más simples  Tienen un conjunto de instrucciones reducido y simple.  Tienden a utilizar un enfoque pipeline, dividiendo la ejecución de la instrucción en etapas. Lo que hace énfasis en el paralelismo.  Ejecuta una instrucción de maquina en cada ciclo maquina.  La mayoría de las operaciones deben ser de tipo registro a registro.  Modos de direccionamiento sencillo. |

*Analisis de prestaciones en procesadores RISC:*

Ciclos de reloj por instrucción: Cuantos ciclos de reloj se necesitan promedio para ejecutar una instrucción.

Ancho de banda de memoria: Evaluar la eficiencia en el acceso a memoria.

Throughput: Cuantas instrucciones se pueden ejecutar por unidad de tiempo.

Eficiencia del Pipeline: Evaluar el rendimiento del pipeline y la capacidad para ejecutar instrucciones en paralelo.

Tasa de Fallos de Caché: Evaluar la eficiencia de la jerarquía de la memoria caché.

**Que es un Bus PCI y un Bus SCSI?**

Bus PCI: Se utiliza para conectar periféricos a la placa madre de una computadora. Su función principal es proporcionar una interfaz común para una variedad de dispositivos.

Bus SCSI: conecta dispositivos de almacenamiento y otros periféricos a una computadora. Utiliza un enfoque en cadena o conexión en paralelo, permitiendo que varios dispositivos estén conectados en serie.

**Diferencias entre bus PCI y bus SCSI:**

El bus PCI se centra en proporcionar una interfaz para tarjetas de expansión, mientras que el bus SCSI conecta varios dispositivos y periféricos.

**Que son las interrupciones?**

Las interrupciones son un mecanismo que permite alterar el proceso de ejecución normal de la CPU. Una interrupción permite que la CPU suspenda lo que esta ejecutando y responda a una rutina de interrupción.

Mientras la CPU esta ejecutando, si recibe un pedido de interrupción, y esa interrupción esta habilitada, se detiene la ejecución del programa en curso, la cpu salva su contexto, guarda el PC y el registro de estado.

Atiende la interrupción (Dependiendo si es interrupción por Software o Hardware se le pasa el control a unidades diferentes para atender la interrupción.) Una vez que termina de atender la interrupción, se recupera el PC, y se sigue con la siguiente instrucción, siempre y cuando haya otra instrucción que ejecutar y que no haya ningún otra interrupción para atender.

*Existen dos tipos de interrupciones:*

Interrupciones por Hardware: Son producidas por dispositivos de entrada/salida y no están relacionadas con el programa en curso, es decir, este tipo de interrupción puede ocurrir en cualquier momento, por eso se dice que son asincrónicas.

Al momento de atender este tipo de interrupciones, se le pasa el control a un vector al manejador de interrupciones, el cual es el encargado de atender y ejecutar las rutinas de interrupción necesarias.

Interrupciones por Software: Son producidas por el programa en curso, algunos ejemplos pueden ser, división por cero, overflow en una operación.

El que interrumpe es el mismo programa, y se le pasa el control al S.O que atiende las interrupciones ejecutando las rutinas de interrupción correspondientes.

**Hay 3 Formas de detectar pedidos de interrupción:**

Opción 1: Líneas de interrupción:

* Cada dispositivo tiene una línea física conectada ala CPU, de esta forma, se puede reconocer al dispositivo que emite una solicitud de interrupción.

Opción 2: Encuesta o Polling:

* Existe una línea física de pedidos de interrupción, la cual comparten varios dispositivos. En este método, la CPU pregunta a cada dispositivo conectado, para saber cual fue el que emitió la solicitud de interrupción.
* Este método puede resultar ineficiente, ya que se le debe preguntar a cada dispositivo conectado.

Opción 3: Vector de Interrupciones:

* Cada interrupción está asociada con un id único, este id sirve como índice para acceder a la rutina de interrupción especifica.
* Con el id de interrupción se busca en la tabla de interrupción, la dirección del comienzo de la rutina de interrupción para ser atendida.

Opcion 4: Daisy Chain:

* Todos los dispositivos se conectan en cadena, y a cada dispositivo se le asignan prioridades. De esta manera, cuando se genera una interrupción se atiende por orden de prioridad de manera decreciente. Es decir, se atiende primero el dispositivo de prioridad más alta, hasta el dispositivo de prioridad mas baja.
* Una vez que se identifica al dispositivo que genero la interrupción, se puede ejecutar las rutinas de interrupción correspondientes.

Interrupciones ENMASCARABLES y NO ENMASCARABLES

*Las interrupciones Enmascarables* son aquellas que pueden ser ignoradas temporalmente, es decir, no indican eventos críticos o de alta prioridad.

*Las interrupciones No Enmascarables* son aquellas que no pueden ser ignoradas, ya que indican eventos de alta prioridad, y deberían ser atendidas lo antes posible.

***Tratamiento de múltiples interrupciones:***

Múltiples dispositivos pueden generar interrupciones simultáneamente, Es importante tener un buen manejo de las interrupciones para evitar conflictos y perdidas de datos.

Se pueden seguir dos alternativas para tratar las interrupciones multiples.

* **Primer Alternativa:** desactivar/enmascarar las interrupciones mientras se esta ejecutando una interrupción. Una interrupción enmascarada significa que el procesador puede y debe ignorar la señal de petición de interrupción por el momento. De esta manera, si se produce una interrupción mientras se atiende una interrupción, quedaría pendiente y será atendida una vez que se hayan activado las interrupciones. La única desventaja es que no tiene en cuenta la prioridad de las instrucciones,
* **Segunda Alternativa:** Definir prioridades para las interrupciones y permitir que una interrupción de prioridad más alta pueda interrumpir a una interrupción de menor prioridad. De esta manera, las interrupciones de mayor prioridad se atenderán primero. Las interrupciones de prioridad alta indican eventos críticos que deben atenderse lo antes posible.

***Cual es la relación entre las interrupciones y el manejo de operaciones de e/s?***

e/s programada por interrupción: El controlador de e/s se configura para emitir una interrupción cuando se complete la operación de e/s o cuando necesite atencion. De esta manera, cuando un dispositivo de e/s completa la operación o necesite atención, puede generar una interrupción para notificarle al procesador. De esta manera, el procesador no tiene que esperar activamente, sino que puede atender y realizar otras tareas mientras espera la respuesta del dispositivo.

**Que es la e/s**

La entrada/salida se refiere a la comunicación entre los componentes del sistema y sus dispositivos externos. Los dispositivos de e/s pueden ser, teclados, mouse, discos duros, impresora.

La CPU gestiona la e/s encargándose de la transferencia de datos entre la memoria y los disposiivos de e/s. Los dispositivos de e/s operan en velocidades mucho mas lentas que la CPU, por lo que es importante que la CPU coordine estas operaciones de manera eficiente.

La capacidad de comunicarse con los dispositivos externos es una función clave, facilitando la transferencia de datos de manera eficiente y confiable.

El módulo de E/S se define como una pieza central para la coordinación efectiva de las operaciones de entrada y salida, integrando funciones de control, comunicación, almacenamiento y detección de errores para facilitar una interacción fluida entre el procesador y los dispositivos externos.

**C­­ontrolador de e/s**

El controlador de e/s, también conocido como subsistema de e/s se encarga de gestionar la comunicación entre el procesador, la memoria principal y los dispositivos de e/s, asegurando que la transferencia sea eficiente y ordenada.

**Estructura de un modulo de e/s:**

El modulo de e/s se conecta al resto de los componentes a través de un conjunto de líneas, por ejemplo líneas de bus del sistema. Los datos que se transfieren hacia y desde el modulo se almacenan en uno o mas registros de datos. Ademas puede haber registros de estado: Un registro de estado puede funcionar como un registro de control, pueden ser utilizados como señales de arbitraje y estado.

**Tipos de puertos de e/s:**

**Puertos paralelos**: Hay varias líneas de datos (n) que transfieren n bits simultáneamente entre el puerto de e/s y el periférico. Se requiere de una conexión mediante un cable costoso. (Mayor línea de datos, más rápida, más costosa)

**Puerto serie**: Hay 1 linea de dato para la transferencia entre el puerto de e/s y el periférico. Se requiere una conexión mediante un cable sencillo. Se pueden transmitir de a 1 bit a la vez. Es mucho mas lento que transmitir n bits simultáneamente.

(Menor línea de datos, mas lento, menos costoso).

**Hay dos tipos de canales de e/s:**

Canal Multiplexor: Varios dispositivos de e/s comparten un único canal de comunicación con la CPU. Tiene la capacidad de controlar a los dispositivos simultáneamente. Puede encargarse de la transferencia de datos, direcciones, señales de control, al mismo tiempo.

Canal Dedicado: Controla varios dispositicos, pero de a 1 a la vez. Selecciona un dispositivo y efectua la transferencia. Puede ser transferencia de datos, direcciones o señales de control, pero debe ser un solo tipo de transferencia.

**Hay 3 tecnicas de e/s :**

**e/s programada con espera de respuesta: :** Se refiere a que el procesador emite una solicitud de operación de e/s y espera activamente hasta que el dispositivo e e/s complete la operación y le avise al procesador que la operación ha finalizado. Durante este periodo de espera, el procesador permanece inactivo, ya que se dedica a verificar repetidamente el estado de la operación de e/s. Una vez que el procesador recibe la señal de finalización de la operación, puede seguir ejecutando.

Este método, resulta ineficiente para operaciones prolongadas, ya que el procesador quedaría inactivo durante largos periodos de tiempo. Es mas adecuado usarlo cuando tenemos operaciones cortas, donde el tiempo de espera sea breve.

**e/s por interrupción:** En este método, antes de iniciar la operación, el controlador de e/s se configura para generar una interrupción cuando la operación se complete. De esta manera, mientras el controlador de e/s se encara de realizar la operación, el procesador puede continuar ejecutando otras tareas y no queda inactivo. Una vez que e complete la operación, se genera una interrupción, entonces el procesador interrumpe sus tareas y ejecuta una rutina de servicio de interrupciones. Despues de manejar la interrupción, el procesador retorma la tarea que estaba ejecutando antes de ser interrumpido**. Este método permite al procesador realizar otras tareas mientras espera que se complete la operación de e/s**

**DMA (Acceso Directo a Memoria):** El DMA es una técnica de transferencia de datos entre el periférico y hacia o desde la memoria, sin intervención directa de la CPU. Esta llevada a cabo por un controlador de DMA (DMAC), el cual es el responsable de llevar a cabo la transferencia.

**Modulo de e/s:**

La función principal de un módulo de E/S es facilitar la interacción eficiente entre el procesador y los dispositivos externos. Esto implica coordinar el flujo de datos, comunicarse tanto con el procesador como con los dispositivos externos, almacenar temporalmente datos cuando sea necesario y detectar posibles errores para mantener la integridad y confiabilidad del sistema.

**Transmisión sincrónica y asincrónica:**

***Transmisión sincrónica:*** En este tipo de transmisión, los dispositivos comparten un reloj común para sincronizar la transferencia de datos. Los datos se envían en bloques y cada bloque se transmite en función de pulsos de reloj. Esto garantiza que el emisor y el receptor estén coordinados en cuanto al tiempo.

***Transmisión asíncrona:*** No hay un reloj que determine una sincronización para enviar, se utilizan bits de inicio y de parada para delimitar cada byte de datos. Cada byte se envía de forma independiente y el receptor debe interpretar la secuencia de bits sin depender de un reloj.

**Pasajes de Argumentos a Subrutinas:**

Los argumentos son usados para la comunicación de datos entre el programa principal y una subrutina.

Los tipos de pasajes de parámetros son:

**Via Registros:** Los argumentos son pasados a través de los registros del procesador.

Estos registros son rapidos de acceder. Este método esta limitado por la cantidad de registros, por lo tanto, es eficiente para un pequeño numero de argumentos.

**Via Memoria:** Los argumentos son almacenados en ubicaciones especificas de la memoria RAM, la subrutina accede a estos valores a través de direcciones de memoria.

Puede ser útil cuando se tienen mas argumentos que los registros, pero el acceso a la memoria es mas lento en comparación con los registros..

**Via Pila (stack):** Los argumentos son apilados antes de invocar a la subrutina, mediante la instrucción de PUSH, y en la subrutina se desapilan para acceder a los valores con la instrucción de POP. Puede ser mas lento debido a las operaciones de apilado y desapilado. La única limitación es tener un buen manejo y entender como trabaja la pila.

**Que es el PIC?**

Se encarga de gestionar las interrupciones, permitiendo que multiples dispositivos y periféricos compartan líneas de interrupción del procesador. El PIC es el responsable de recibir señales de interrupción de varios dispositivos como periféricos y dirigir estas solicitudes a la CPU. Se puede configurar para asignar prioridades a las interrupciones, es decir, permite establecer niveles de prioridad para las distintas fuentes de interrupción. Ademas, permite habilitar o deshabilitar (enmascarar) las interrupciones. Cuando una interrupción esta enmascarada, se ignorará. Ademas el PIC puede operar las interrupciones internamente o en modo cascada, y cuenta con registros internos como son:

IRR:Especifica que interrupciones están pendientes de reconocimiento

ISR: Especifica que interrupciones fueron conocidas y están siendo atendnidas.

EOI: Marca el final de una interrupción. Cuando la CPU ha completado el servicio de interrupción, se envía una señal EOI al PIC.

IMR: Especifica que instrucciones deben ser ignoradas.

**Conceptos de la memoria cache:**

El objetivo de la memoria cache es lograr que la velocidad de la memoria sea lo mas rápida posible. La cache almacena copias de partes de la memoria principal. Se conecta al procesador a través de líneas de datos, control y direcciones.

La memoria cache se basa en el principio de localidad, se divide en dos tipos:   
Localidad espacial: se basa en el acceso a direcciones de memoria cercanas. Es probable que se acceda a direcciones cercanas en un futuro.   
Localidad temporal: se basa en el acceso repetido a las mismas direcciones de memoria.

La cache se organiza en bloques o líneas de cache. Un bloque representa la cantidad de datos transferidos entre la memoria principal y la caché en una única operación.

Resumen: La cache, opera mediante la gestión de datos basada en patrones de acceso frecuente, utilizando bloques para agilizar las transferencias.

**Formas de correspondencia de la cache:** La memoria caché es un tipo de memoria de acceso rápido que almacena copias de datos utilizados con frecuencia, con el objetivo de que el acceso a los datos sea lo mas rápido posible.

**Hay tres formas de correspondencia de la cache:**

**Correspondencia directa:** Cada bloque de memoria de la caché se mapea directamente en un bloque correspondiente en la memoria principal. Esto significa que cada ubicación en la memoria principal tiene una única ubicación en la caché donde puede almacenarse.

**Correspondencia asociativa por conjuntos:** Cada bloque de memoria de la caché tiene varias ubicaciones en la memoria principal donde puede almacenarse. Esto permite cierto grado de flexibilidad y reduce la probabilidad de conflictos de caché.

**Correspondencia totalmente asociativa:** Cualquier bloque de memoria de la caché puede almacenarse en cualquier ubicación de la memoria principal.

**Politicas de Reemplazo / algoritmos de sustitucion de cache:**

***LRU (Least Recently Used - Menos Recientemente Utilizado):*** Reemplaza el bloque de caché que no ha sido utilizado durante el período más largo de tiempo. Es decir, se elimina el bloque que ha sido accedido menos recientemente.

***FIFO (First In, First Out - Primero en Entrar, Primero en Salir):*** El bloque que ha estado en la caché durante el período de tiempo mas largo, es el primero en ser reemplazado.

***LFU (Least Frequently Used - Menos Frecuentemente Utilizado):*** Reemplaza el bloque de caché que ha sido menos frecuentemente accedido. Es decir, se cuenta el número de accesos a cada bloque de la caché y se elimina el bloque con el menor número de accesos.

***Random (Aleatorio):*** El bloque de caché a reemplazar se elige al azar. No se tiene en cuenta el historial de acceso o la frecuencia de uso.

**Politicas de escritura en cache:**

***(Escritura Directa):*** Cada escritura en la caché también se escribe de inmediato en la memoria principal. Esto garantiza que la memoria principal esté siempre actualizada, pero puede generar un mayor tráfico en el bus de memoria.

***(Escritura Postergada):*** Las escrituras se realizan solo en la caché y se escribe en la memoria principal solo cuando el bloque se reemplaza en la caché. Esto reduce el tráfico en el bus de memoria y puede mejorar el rendimiento.

**Diferencias en la invocación y finalización de subrutinas y rutinas de interrupción**

Invocación: Las subrutinas se invocan explícitamente mediante una instrucción de llamada. El control del programa pasa a la subrutina y una vez finalizada vuelve al punto donde fue llamada. A una subrutina se le pueden pasar argumentos, pueden ser via pila, memoria o registros. Las subrutinas generalmente son utilizadas para modularizar el código, facilitando el mantenimiento y legibilidad del programa.

Mientras que, las rutinas de interrupción son activadas en respuesta a eventos específicos, como pueden ser interrupciones de hardware o de software, es decir, no son invocadas por el programador y pueden ocurrir en cualquier tiempo. El control se transfiere a la rutina de interrupción y una vez atendida la interrupción el control vuelve al punto del programa principal donde fue interrumpido. Para que una interrupción sea atendida, debe estar habilitadas las interrupciones o esa interrupción en especifico, caso contrario se ignorara el pedido de interrupción.

En ambos casos, antes de ejecutar una subrutina o una rutina de interrupción, se guarda el contexto del programa, se salva el estado y se guarda el PC que contiene la próxima instrucción a ejecutar.

**Que es el DMA? Cual es su funcionamiento?**

El DMA es una técnica de transferencia de datos entre el periférico y hacia o desde la memoria, sin intervención directa de la CPU. Esta llevada a cabo por un controlador de DMA (DMAC), el cual es el responsable de llevar a cabo la transferencia.

El DMA es usado para mejorar la transferencia de datos entre dispositivos periféricos y la memoria principal a través del bus del sistema. Cuando un dispositivo necesita transferir datos a la memoria, el DMA toma el control temporalmente del bus del sistema. Al liberar la CPU de la tarea de gestionar cada transferencia, el DMA reduce la carga del procesador.   
**Existen dos modos de transferencia de DMA:**

Transferencia por bloques: El DMA solicita el control del bus tantas veces como sea necesario hasta finalizar la transferencia del bloque completo.

Transferencia por ráfaga: El DMA no libera el bus hasta haber finalizado la transferencia de todo el bloque de datos.

En general, para transferencias de e/s mas prolongadas, la técnica por bloques es la mas eficiente, ya que permite implementar la transferencia al mismo tiempo que la CPU continua trabajando en otras tarea.

**Conceptos De Memoria:**

**Memoria Principal (RAM - Random Access Memory):**

La memoria RAM almacena temporalmente datos e instrucciones que la CPU necesita en tiempo real.Es Volatil, lo que significa que pierde su contenido cuando se apaga la computadora.La memoria RAM permite un acceso rápido y aleatorio a cualquier ubicación de almacenamiento.

**Memoria Secundaria:**

La memoria secundaria es otro tipo de almacenamiento que se utiliza para guardar datos de forma no volátil. A diferencia de la RAM, la memoria secundaria retiene su contenido incluso cuando se apaga la computadora.

Se usa para almacenar el sistema operativo, programas y datos de manera permanente. Es mas lenta que la RAM en términos de acceso a los datos, pero proporciona un almacenamiento masivo y persistente.

**Memoria Virtual:**

La memoria virtual es un concepto que permite que un sistema operativo utilice tanto la memoria RAM como el almacenamiento secundario (disco duro) como si fueran una única memoria continua.

Permite a los programas utilizar más memoria de la que realmente está disponible físicamente en la RAM.

Permite que los programas utilicen más memoria de la que realmente está presente físicamente, mejorando así el rendimiento al evitar limitaciones de memoria.

**Segmentacion de cauce:**

La segmentación de cauce consiste en descomponer el proceso de ejecución de instrucciones en varias fases o etapas. Estas fases o etapas son ejecutadas por unidades separadas, y son capaces de trabajar simultáneamente. Las instrucciones se ejecutan a medida que se liberan unidades, sin la necesidad de esperar a que termine una instrucción para ejecutar la siguiente.

***Al estar ejecutando multiples instrucciones en paralelo, pueden producirse conflictos*:**

Conflicto por Dependencia de datos: Ocurren cuando dos o mas instrucciones comparten un mismo dato, y una instrucción necesita el resultado de otra.

Algunas soluciones son: Reordenamiento de instrucciones o Forwarding (adelantamiento de operandos).

Conflctos estructurales: Ocurren cuando dos o mas partes del hardware compite por el mismo recurso. Es decir, esta provocado por el uso de recursos, como memoria, alu, y registros.

Conflictos de dependencia de control: Surgen cuando la ejecución de una instrucción depende de cómo se ejecute otra. Ejemplo: 1 salto y 2 posubkes caminos.

Alguna soluciones son: Usar técnicas de predicción de saltos, o poner instrucciones NOP.

Las instrucciones NOP, retrasa ciclos del reloj.

**Procesamiento Paralelo:**

El procesamiento paralelo hace énfasis en que múltiples tareas se ejecuten simultáneamente para mejorar la eficiencia y el rendimiento de un sistema.El procesamiento paralelo distribuye y ejecuta múltiples tareas al mismo tiempo, ya sea en diferentes núcleos de procesamiento dentro de un procesador, en diferentes procesadores en un sistema multiprocesador, o incluso en sistemas conectados en red. Este enfoque permite abordar problemas más grandes y complejos de manera más rápida y eficiente al dividirlos en tareas más pequeñas y ejecutarlas en paralelo

**Procesamiento Paralelo a nivel de instrucciones:**

El procesamiento paralelo a nivel de instrucciones se refiere a la ejecución simultánea de múltiples instrucciones de un programa en un solo procesador. En lugar de ejecutar una instrucción a la vez, el procesador intenta identificar y ejecutar instrucciones independientes de manera simultánea, aprovechando el paralelismo a nivel de instrucción dentro de un flujo de instrucciones. Esto se logra mediante técnicas como la ejecución fuera de orden, la predicción de saltos y la segmentación de instrucciones para mejorar el rendimiento y la eficiencia del procesador.

El procesamiento paralelo a nivel de instrucciones es fundamental para aumentar el rendimiento de los procesadores modernos al aprovechar al máximo su capacidad.

**Paralelismo a nivel de maquina:**

Es una medida de la capacidad del procesador para sacar provecho al paralelismo a nivel de instrucciones. El paralelismo de la maquina depende del numero de instrucciones que pueden captarse y ejecutarse al mismo tiempo.

**Prediccion de saltos:**

Con la llegada de las maquinas RISC se exploró la estrategia de salto retardado. Esta estrategia permite al procesador calcular el resultado de las instrucciones de bifurcación condicional antes de que se precapte cualquier instrucción inservible.

**Caracteristica de los procesadores superescalares:**

Los procesadores superescalares tienen la capacidad de ejecutar múltiples instrucciones en paralelo, pueden realizar varias operaciones en cada ciclo de reloj. Utilizan la segmentación de instrucciones para dividir la ejecución de las instrucciones en varias etapas. Tienen varias unidades funcionales, conocidas como unidades de ejecución. Los procesadores superescalares, usan renombramiento de registros para evitar conflictos de dependencia de datos, y tienen técnicas avanzadas en predicción de saltos.

**Que características posee un procesador supersegmentado frente a un superescalar?**

Procesador supersegmentado: Divide el flujo de instrucciones en segmentos mas pequeños, ejecutando varias etapas de instrucciones en paralelo. Aprovecha las instrucciones mas cortas para lograr una mayor eficiencia en el uso del pipeline. Este tipo de procesadores puede ser más eficiente para instrucciones más simples y cortas.

El supersegmentado dobla la velocidad del reloj interno y permite la realización de tareas en un solo ciclo de reloj.

El procesador superescalar puede ser más eficiente para instrucciones complejas y largas, ya que puede aprovechar las dependencias entre instrucciones para ejecutarlas en paralelo.

**Renombramiento de registros en procesadores superescalares:**

El renombramiento de registros es una técnica que permite ejecutar instrucciones en paralelo de manera eficiente. Cuando una instrucción se empieza a ejecutar, se le asigna un registro físico en lugar de uno lógico. Esto permite que múltiples instrucciones utilicen los mismos registros lógicos sin interferencias. Cuando las instrucciones se completan, los resultados se desalojan de los registros físicos y se escriben en los registros lógicos. Este proceso evita conflictos de dependencia de datos, permitiendo una ejecución en paralelo eficiente y de forma independiente.

**Cual es el objetivo de usar la técnica de renombramiento de registros?**

El objetivo de usar la técnica de renombramiento de registros, es eliminar los conflictos de dependencia de datos y optimizar la ejecución de las instrucciones en paralelo. El renombramiento de registros aborda este problema asignando registros físicos a registros lógicos. Permitiendo que multiples instrucciones usen registros lógicos sin generar conflictos.

**Que es el MIMD de la taxonomía Flynn:** En un sistema MIMD, múltiples procesadores independientes ejecutan diferentes instrucciones al mismo tiempo en conjuntos distintos de datos. Cada procesador tiene su propio flujo de control y puede ejecutar instrucciones de manera independiente. Ademas,en un sistema MIMD, los procesadores pueden trabajar en conjunto para resolver problemas mas complejos mediante la ejecución simultanea de tareas independiente. Los procesadores en un sistema MIMD son capaces de ejecutar tareas en paralelo.

Los sistemas MIMD son comúnmente usados en entornos de computación de alto rendimiento.

Los MIMD pueden clasificarse en dos categorías:

**MIMD con memoria Compartida**: Varios procesadores comparten un espacio de memoria en común. Pueden comunicarse en esta memoria compartida mediante la lectura y escritura.

**MIMD con memoria distribuida**: Cada procesador tiene su propia memoria local y no comparte memoria física con otros procesadores. La comunicación se lleva a cabo mediante mensajes u cualquier otro mecanismo de comunicación.

**Que son las interrupciones vectorizadas?**

Las interrupciones vectorizadas son un mecanismo en el que se utiliza una tabla de vectores de interrupción para gestionar interrupciones. Este mecanismo permite que el sistema maneje multiples tipos de interrupciones de manera eficiente.

Cada interrupción tiene un id que la identifica en la tabla de vectores, esta tabla contiene las direcciones de memoria de las rutinas de manejo de interrupción. Cada entrada en la tabla esta asociada con un numero de interrupción.

Cuando ocurre una interrupción, el procesador utiliza el numero de interrupción para acceder a la tabla y obtener la dirección de la rutina de manejo de interrupcion, y luego ejecuta esa rutina de interrupción.

***Clasificación de arquitecturas paralelo y taxonomía de Flynn***

La taxonomía de Flynn clasifica los sistemas en función de dos características principales: el número de flujos de instrucciones y el número de flujos de datos que se manejan simultáneamente.

***SISD (Single Instruction, Single Data - Instrucción Única, Datos Únicos)***:

Un único procesador interpreta una única secuencia de instrucciones para operar con los datos almacenados en una única memoria.

***SIMD (Single Instruction, Multiple Data - Instrucción Única, Múltiples Datos):***

Una única instrucción maquina controla el paso a paso de la ejecución simultanea y sincronizada de elementos del proceso.

Se utiliza en sistemas que realizan operaciones paralelas en grandes conjuntos de datos, como gráficos por computadora, procesamiento de imágenes y algunas aplicaciones científicas.

***MISD (Multiple Instruction, Single Data - Múltiples Instrucciones, Datos Únicos):***

Se transmite una secuencia de datos a un conjunto de procesadores, cada uno de ellos ejecuta una secuencia de instrucciones diferente.

***MIMD (Multiple Instruction, Multiple Data - Múltiples Instrucciones, Múltiples Datos):***

Multiples procesadores ejecutan simultáneamente una secuencia de instrucciones diferentes con conjuntos de datos diferentes.

Los SMP, los clusters y los sistemas NUMA abarcan esta categoria

**Arquitecturas Multiprocesador:**

Son sistemas informáticos que tienen múltiples unidades de procesamiento (procesadores) trabajando en paralelo para ejecutar tareas y programas. Estos sistemas se utilizan para mejorar el rendimiento y la capacidad de procesamiento de las computadoras, permitiendo que múltiples procesadores compartan la carga de trabajo y trabajen juntos en la resolución de problemas.

***Multiprocesamiento simétrico (SMP):*** Todos los procesadores tienen acceso uniforme a la memoria compartida y al bus de sistema. Todos los procesadores tienen acceso equitativo a los recursos del sistema. Los sistemas SMP son comunes en servidores y estaciones de trabajo de gama alta.

***Multiprocesamiento asimétrico (AMP):*** Tienen procesadores con roles diferentes y niveles de acceso a los recursos del sistema. Por ejemplo, puede haber un procesador principal más poderoso que maneje ciertas tareas críticas mientras que otros procesadores más simples se encargan de tareas menos intensivas en recursos. Los sistemas AMP se utilizan a menudo en dispositivos embebidos y sistemas integrados.

**SMP:** Es un tipo de arquitectura de procesamiento en paralelo, que permite ejecutar dos o mas proceso en paralelo en un mismo sistema. En un sistema SMP, cada procesador tiene acceso a la misma memoria y recursos de e/s. Esto significa que los procesos pueden acceder y compartir datos entre si de forma sencilla. Son especialmente adecuados para procesamiento de imágenes y análisis de datos.

**Que son los CLUSTERS?**

Los clusters son un conjunto de computadoras interconectadas entre si, que trabajan juntas como si fueran una única unidad. Estan conectadas entre si para colaborar en tareas especificas y compartir recursos. Los clusters son utilizados en aplicaciones que requieren una gran cantidad de procesamiento o almacenamiento de datos.

**Que es el DMA? Cual es su funcionamiento?**

El DMA es una técnica de transferencia de datos entre el periférico y hacia o desde la memoria, sin intervención directa de la CPU. Esta llevada a cabo por un controlador de DMA (DMAC), el cual es el responsable de llevar a cabo la transferencia.

El DMA es usado para mejorar la transferencia de datos entre dispositivos periféricos y la memoria principal a través del bus del sistema. Cuando un dispositivo necesita transferir datos a la memoria, el DMA toma el control temporalmente del bus del sistema. Al liberar la CPU de la tarea de gestionar cada transferencia, el DMA reduce la carga del procesador.   
Existen dos modos de transferencia:

Transferencia por bloques: El DMA solicita el control del bus tantas veces como sea necesario hasta finalizar la transferencia del bloque completo. Transferencia por ráfaga: El DMA no libera el bus hasta haber finalizado la transferencia de todo el bloque de datos.

En general, para transferencias de e/s mas prolongadas, la técnica por bloques es la mas eficiente, ya que permite implementar la transferencia al mismo tiempo que la CPU continua trabajando en otras tareas.

**Describir un modulo de e/s:** Un modulo de e/s permite que el procesador gestione varios dispositivos periféricos. El modulo de e/s se conecta con el procesador a través de un conjunto de líneas de datos, dirección y control. Los datos que se transfieren se almacenan temporalmente en un registro de datos. El registro de control permite configurar y programar diferentes funciones del modulo.

Un modulo de e/s dispone de la lógica especifica para su conexión con uno o mas dispositivos periféricos. Ademas la lógica de la interfaz oermite al procesador interactuar con diferentes dispositivos periféricos de manera consistente.

El modulo de e/s se ocupa de la comunicación entre un programa y el entorno externo, que puede ser el usuario, archivos en el disco, etc.

**El modulo de e/s tiene dos puertos:**

Puerto paralelo: Hay varias líneas de datos, que transfieren n bits simultáneamente entre el puerto e/s y el periférico. Se requiere una conexión mediante un cable, lo cual lo hace un método costoso.

(Mayor línea de datos, mas rápido, mas costoso)

Puerto serie: hay 1 linea de dato para la transferencia entre el puerto de e/s y el periférico. Se requiere de una conexión mediante un cable sencillo. Los datos deben transmitirse de a 1 bit a la vez

(Menor línea de datos, mas lentos, menos costoso)

**Formas de direccionamiento de e/s:** El direccionamiento de e/s se refiere a como el procesador accede a los dispositivos periféricos para enviar o recibir datos.

Algunas formas comunes de direccionamiento de e/s son:

Direccionamiento por Puertos (I/O PORTS): Implica asignar un conjunto de direcciones de memoria especificas a los puertos de e/s. Cada puerto se considera una ubicación de memoria única a la que el procesador puede acceder para enviar o recibir datos desde o hacia dispositivos periféricos.

Direccionamiento por Mapeo de Memoria: En este método, los registros de control y de estado de los periféricos se asignan a direcciones de memoria especificas. El procesador utiliza instrucciones de carga y almacenamiento (LOAD / STORE) para interactuar con los periféricos.

***Preguntas de finales con frecuencia:***

**Como es la estructura de un modulo de e/s?**

En general, el modulo de e/s suele constar de los siguientes componentes:

Controlador de e/s: Es el componente encargado de gestionar y coordinar las operaciones de e/s.

Registro de Control y Estado: Almacenan información sobre el estado actual de la operación de e/s y permiten la configuración y control del modulo por parte del CPU.

Buffer de Datos: Es un area de almacenamiento temporal que se utiliza para almacenar datos que entran o salen dispositivos e/s. Facilita la transferencia eficiente de datos entre el dispositivo y la memoria principal.

Interfaz de Datos: Permite la conexión física y lógica entre el modulo de e/s y el dispositivo periférico.

**Describa los algoritmos de reemplazo de Bloque y políticas de escrituras en la caché:**

**Politicas de Reemplazo / algoritmos de sustitucion de cache:**

***LRU (Least Recently Used - Menos Recientemente Utilizado):*** Reemplaza el bloque de caché que no ha sido utilizado durante el período más largo de tiempo. Es decir, se elimina el bloque que ha sido accedido menos recientemente.

***FIFO (First In, First Out - Primero en Entrar, Primero en Salir):*** El bloque que ha estado en la caché durante el período de tiempo mas largo, es el primero en ser reemplazado.

***LFU (Least Frequently Used - Menos Frecuentemente Utilizado):*** Reemplaza el bloque de caché que ha sido menos frecuentemente accedido. Es decir, se cuenta el número de accesos a cada bloque de la caché y se elimina el bloque con el menor número de accesos.

***Random (Aleatorio):*** El bloque de caché a reemplazar se elige al azar. No se tiene en cuenta el historial de acceso o la frecuencia de uso.

**Politicas de escritura en cache:**

***(Escritura Directa):*** Cada escritura en la caché también se escribe de inmediato en la memoria principal. Esto garantiza que la memoria principal esté siempre actualizada, pero puede generar un mayor tráfico en el bus de memoria.

***(Escritura Postergada):*** Las escrituras se realizan solo en la caché y se escribe en la memoria principal solo cuando el bloque se reemplaza en la caché. Esto reduce el tráfico en el bus de memoria y puede mejorar el rendimiento.

**Politicas de escritura en la caché:**

Write Allocate: Si se necesita escribir datos en la memoria y esa línea de cache donde se desea escribir no esta presente en la caché, primero se trae esa línea completa a la cache antes de realizar la escritura

No write allocate: Si necesitas escribir datos en la memoria y la línea de cache donde se desea escribir no esta presente en la caché. Se escribe directamente en la memoria principal sin traer la línea a la caché.

**Mencionar tres características importantes de las arquitecturas RISC:**

¡) Repertorio de instrucciones reducidos: Las arquitecturas RISC se caracterizan por tener un conjunto de instrucciones reducido. Esto significa que las instrucciones son mas simples y se centran en operaciones fundamentales

¡ ¡) Segmentación de Cauce: Las arquitecturas RISC aprovechan la segmentación de cauce, o pipeling para mejorar la eficiencia del procesamiento. En este enfoque, la ejecución de una instrucción se divide en etapas, y cada etapa se realiza en paralelo.

¡ ¡ ¡) Enfoque en Rendimiento y Velocidad: Las arquitecturas RISC están diseñadas para maximizar el rendimiento y la velocidad de ejecución de instrucción.

***Memoria caché: Describa el mapeo asociativo por conjuntos. Analice las policitas de escritura desde el punto de vista de coherencia de datos:***

El mapeo asociativo por conjuntos, es una técnica en la cual la cache se divide en grupos o conjuntos. Cada conjunto tiene varias líneas de caché. Y cada línea de caché puede almacenar un bloque de datos.

**Describa las características funcionales del Acceso Directo a Memoria (DMA)**

El Acceso Directo a Memoria es un mecanismo el cual permite que multiples dispositivos periféricos transfieran datos hacia o desde la memoria principal, sin la intervención directa de la CPU. Lo cual, esto permite a la CPU realizar otras tareas mientras se efectua la transferencia.

Ademas, el dma tiene su propio controlador (DMAC) el cual gestiona las operaciones de transferencias ade datos, este controlador se puede programar con la información necesaria para realizar operaciones de transferencias.

**Que es la segmentación de cause? Que ventajas proporciona su implementación?**

La segmentación de cauce, es una técnica usada para mejorar el rendimiento y la eficiencia del procesamiento de instrucciones. En un procesador segmentado, el ciclo de ejecución de una instrucción se divide en etapas y cada etapa es ejecutada por una unidad de hardware diferente. Cada unidad se especializa en realizar una tarea especifica.

Entre sus ***ventajas***, se destacan:

Mejora del rendimiento, al dividir el proceso de ejecución en etapas, multiples instrucciones pueden estar en diferentes etapas del cause simultáneamente.

Se basa en el concepto de paralelismo a nivel de instrucción. Puede ejecutar varias instrucciones simultáneamente.

**FORMAS DE CORRESPONDENCIA DE LA CACHE:**

***Correspondencia Directa:*** Cada bloque de memoria principal tiene un único lugar en la caché donde puede residir. La dirección de memoria determina directamente la ubicación en caché del bloque.

***Correspondencia Asociativa por Conjuntos (Set-Associative***): La caché se divide en conjuntos, y cada bloque de memoria principal puede residir en cualquier ubicación dentro de su conjunto correspondiente. Esto permite cierto grado de flexibilidad en la asignación de bloques.

***Correspondencia Totalmente Asociativa (Fully Associative):*** Cada bloque de memoria principal puede residir en cualquier ubicación dentro de toda la caché, sin restricciones sobre la ubicación. Esto ofrece la mayor flexibilidad pero puede requerir una búsqueda más compleja.

**POLITICAS DE ESCRITURAS EN LA CACHE (FALLO Y ACIERTO)**

***Política de Escritura en Acierto:***

***Escritura Directa:*** Cada vez que se escribe en una ubicación de memoria caché, también se escribe en la ubicación correspondiente en la memoria principal.

*Esto garantiza que la memoria principal y la caché estén siempre actualizadas, pero puede resultar en un mayor tráfico de memoria.*

***Escritura Postergada:*** Las escrituras solo se realizan en la caché. La memoria principal se actualiza solo cuando se reemplaza un bloque de caché que contiene datos modificados. Esto reduce el tráfico de memoria, pero puede llevar a inconsistencias entre la caché y la memoria principal hasta que se realice la escritura de vuelta.

***Política de Escritura en Fallo de caché:***

***Escritura y Asignación:*** Significa que, en caso de un fallo de caché durante una escritura, el bloque de memoria principal se carga en la caché y luego se realiza la escritura en la caché.

***No escritura y asignación:*** Si hay un fallo de caché durante una escritura, los datos se escriben directamente en la memoria principal sin cargar el bloque en la caché. Esto evita la carga innecesaria de datos en la caché para escrituras que no se leerán pronto.

***Caracteristicas que posea un procesador superescalar:***

Los procesadores superescalares tienen la capacidad de ejecutar múltiples instrucciones en paralelo, pueden realizar varias operaciones en cada ciclo de reloj. Utilizan la segmentación de instrucciones para dividir la ejecución de las instrucciones en varias etapas. Tienen varias unidades funcionales, conocidas como unidades de ejecución. Los procesadores superescalares, usan renombramiento de registros para evitar conflictos de dependencia de datos, y tienen técnicas avanzadas en predicción de saltos.

**Que es el MIMD de la taxonomía Flynn:** En un sistema MIMD, multiples procesadores independientes ejecutan diferentes instrucciones al mismo tiempo en conjuntos distintos de datos. Cada procesador tiene su propio flujo de control y puede ejecutar instrucciones de manera independiente. Ademas,en un sistema MIMD, los procesadores pueden trabajar en conjunto para resolver problemas mas complejos mediante la ejecución simultanea de tareas independiente. Los procesadores en un sistema MIMD son capaces de ejecutar tareas en paralelo.

Los sistemas MIMD son comúnmente usados en entornos de computación de alto rendimiento.

Los MIMD pueden clasificarse en dos categorías:

**MIMD con memoria Compartida**: Varios procesadores comparten un espacio de memoria en común. Pueden comunicarse en esta memoria compartida mediante la lectura y escritura.

**MIMD con memoria distribuida**: Cada procesador tiene su propia memoria local y no comparte memoria física con otros procesadores. La comunicación se lleva a cabo mediante mensajes u cualquier otro mecanismo de comunicación.

Finales resueltos:

Final 1:

1. Describir el mecanismo de interrupción. Mencionar cuales son los tipos de interrupciones. Describir el tratamiento de múltiples interrupciones.

Las interrupciones son un mecanismo que permite alterar el proceso de ejecución normal de la cpu. Una interrupción, permite que la cpu suspenda la tarea que esta ejecutando y responda a una rutina de interrupción.

Mientras la CPU está ejecutando, si recibe un pedido de interrupción, y la interrupción **no se encuentra enmascarada** (es decir, esta habilitada), la cpu suspende la ejecución del programa en curso, guarda su contexto, carga el pc (program counter) que contiene la dirección de la próxima instrucción a ejecutar y salva el registro de estado.

Una vez que guarda todo el contexto de lo que estaba ejecutando antes de ser interrumpido, se atiende dicha interrupción, se ejecutan las rutinas de interrupción correspondientes. Al finalizar la rutina de interrupción, evalúa si hay otra interrupción pendiente, si hay algún pedido de interrupción, y esa interrupción se encuentra habilitada, se debe guardar el contexto del programa y atender la interrupción.

Si no hay ningún pedido de interrupción, el procesador recupera el pc y el registro de estado y continua con su proceso de ejecución.

***Hay dos tipos de interrupciones:***

***Interrupciones por Hardware***: Este tipo de interrupciones son generadas por dispositivos de e/s. Pueden no estar relacionadas con el programa en curso. Se dice que son “asincronicas” porque pueden generarse en cualquier momento. Al momento de atender la interrupción se le pasa el control al vector de interrupción, el cual es el encargado de llevar a cabo las rutinas de interrupción necesarias.

***Interrupciones por Software:*** Este tipo de interrupciones están relacionadas con el programa en curso, pueden generarse por división por cero, overflow en una operación, entre otras cosas. Para atender estas interrupciones se le pasa el control al S.O, el cual es el encargado de llevar a cabo las rutinas de interrupción necesarias.

***Interrupciones internas:*** Son producidas por eventos dentro del procesador, o del sistema en si mismo, algunos ejemplos pueden ser división por cero, intentar acceder a zonas de memoria prohibidas. Este tipo de interrupciones se asocian con las interrupciones de Software.

***Interrupciones externas:*** Son producidas por eventos externos al procesador, por dispositivos de e/s, periféricos y no están relacionadas con el programa en curso. Se asocian con las interrupciones de Hardware.

**Tratamiento de multiples interrupciones:**

Multiples dispositivos pueden interrumpir simultáneamente, por eso se debe tener un manejo correcto y eficiente de las interrupciones. Para el tratamiento de multiples interrupciones hay dos casos:

Primera alternativa: Atender las interrupciones según sean generadas, es decir, se genera como una cola de interrupciones al deshabilitar las interrupciones mientras se atienda una interrupción. Es decir, mientras se atiende una interrupción se deshabilitan las demás interrupciones para evitar que interrumpa el servicio de atención actual, de esta manera, si se genera un pedido de interrupción mientras se atiende otra, quedaría pendiente. Cuando se termine de tratar la interrupción actual, se activarían las interrupciones y se atenderá. Esta alternativa no tiene en cuenta las prioridades de las interrupciones.

Segunda alternativa: Asignarle prioridades a las interrupciones y atenderlas por orden de prioridad. Es decir, una interrupción de mayor prioridad debe ser atendida antes que una de menor prioridad. Las interrupciones de prioridad alta indican eventos críticos que deben ser atendidos lo antes posible. De esta manera , si se esta atendiendo una interrupción de prioridad baja, y llega una interrupción de prioridad alta, se interrumpe la ejecución de la proridad menor y se atiende la interrupción de prioridad mayor.

1. **Como es la estructura de un modulo de e/s?** Describir la funcion del DMA

El modulo de e/s facilita la interaccion entre la CPU y los dispositivos periféricos.

La estructura de un modulo de e/s es la siguiente:

* **Controlador de e/s:** *Se encarga de gestionar la comunicación entre la CPU y los dispositivos periféricos*. Tiene varios registros internos para controlar el flujo de datos.
* **Registro de control**: Estos registros *son usados por la CPU para configurar el funcionamiento del dispositivo periférico.*
* **Buffers de datos:** Los buffers de datos son *áreas de memoria usadas para almacenar temporalmente los datos que se transfieren entre la CPU y el dispositivo periférico.*
* **Interfaz de e/s:** *Proporciona los medios físicos para la conexión entre el controlador de e/s y el periférico*. Pueden ser puertos de e/s, conectores, cables
* **Circuitos de control y temporización**: Son los *encargados de sincronizar las operaciones de e/s y garantizar la integridad de los datos durante la transferencia*.

***Funcion del DMA:***

La función del DMA es transferir los datos entre la memoria principal y los periféricos sin la intervención directa de la CPU. Esta transferencia esta llevada a cabo por un controlador de dma (DMAC). El uso del DMA libera a la CPU de la tarea de gestionar cada transferencia. Por lo tanto, esta técnica reduce la carga del procesador.

Para llevar a cabo cada transferencia, el DMA debe solicitar el uso del bus del sistema.

Hay dos tipos de transferencias DMA:

Transferencia por bloques: El DMAC solicita el uso del bus tantas veces como sea necesario hasta finalizar la transferencia.

Transferencia por ráfaga: El DMAC solicita el uso del bus, y no lo libera hasta finalizar la transferencia.

1. Mencionar los tipos de correspondencia de la memoria caché. Describir las políticas de escritura (en acierto y fallo).

Hay 3 tipos de correspondencia de caché:

**Correspondencia Directa:** Cada bloque de memoria principal puede mapearse en una única ubicación de la memoria caché. Es decir, a cada bloque de memoria principal le corresponde solo una ubicación de la memoria caché

Puede generar conflictos si varios bloque de memoria principal se asignan a la misma ubicación de memoria caché.

**Correspondencia asociativa por conjuntos:** Cada bloque de memoria principal se divide en conjuntos y cada conjunto puede mapearse en varias ubicaciones de la memoria caché.

**Correspondencia totalmente asociativa:** Cualquier bloque de la memoria principal puede mapearse sobre cualquier bloque de la memoria caché.

Este enfoque elimina por completo los conflictos de caché.

Politicas de escritura en la caché:

**Escritura Directa:** Es una forma de política de escritura en acierto.

Cada vez que se escribe en la memoria caché, también se escribirá en la memoria principal. Esto garantiza que la memoria principal esté siempre actualizada. Se tiene un mayor uso del bus de memoria, lo cual puede generar un pequeño trafico.

**Escritura Postergada:** Es una forma de política de escritura en fallo.

Cuando se escribe en un bloque de memoria caché, los datos no se reflejan inmediatamente en la memoria principal. Los datos se escriben en la memoria principal cuando se reemplacen en la caché.

1. Que es la segmentación de instrucciones, como mejora el rendimiento? Describir tipos de dependencia que afectan el funcionamiento de los cauces.

La segmentación de instrucciones, consiste en dividir el proceso de ejecución de las instrucciones en varias fases o etapas. Cada etapa es ejecutada por una unidad independiente. Esto introduce un nivel de paralelismo a nivel de instrucciones, donde varias instrucciones pueden ejecutarse simultáneamente. Las instrucciones se ejecutan a medida que se liberan unidades, sin la necesidad de esperar que termine una ejecución para ejecutar la siguiente.

La segmentación de instrucciones mejora el rendimiento de los procesadores modernos, ya que puede ser más eficiente y optimizar el rendimiento ejecutar instrucciones paralelamente. Entre las mejoras se encuentran:

*Mejora del tiempo de ejecución:* En aplicaciones que ejecutan un gran numero de instrucciones, es mas eficiente ejecutarlas paralelamente antes que ejecutarlas secuencialmente.

*Mejora de la utilización de recursos:* Permite que diferentes unidades funcionales trabajen independientemente sin tener que esperar que otras instrucciones se completen.

Al estar ejecutando instrucciones paralelamente, pueden producirse conflictos por dependencias que afecten el funcionamiento y el tiempo de ejecución de los causes.

Entre los conflictos de dependencia se encuentran:

*Conflictos por Dependencia de Datos:* Son generados cuando dos o mas instrucciones comparten un mismo dato, y una instrucción necesita de un dato que todavía no esta disponible.

*Conflicto por Dependencia de Control:* Son generados cuando la ejecución de una instrucción depende de cómo se ejecute otra. Un ejemplo es un salto y dos posibles caminos.

*Conflictos Estructurales:* Son generados cuando dos o mas partes del hardware comparten el mismo recurso. Ejemplo : memoria, alu.

Algunas soluciones para estos conflictos pueden ser:

Reordenamiento de instrucciones, forwarding (adelantamiento de operandos),

instrucciones NOP (tener en cuenta que la instrucción nop retrasa un ciclo de reloj.)

1. Caracteristicas que posee un procesador superescalar

Los procesadores superescalares tienen la capacidad de ejecutar multiples instrucciones en paralelo, pueden realizar varias operaciones en cada ciclo de reloj.

Utilizan la segmentación de instrucciones para dividir la ejecución de las instrucciones en varias fases o etapas. Tienen varias unidades funcionales, conocidas como unidades de ejecución. Los procesadores superescalares usan la técnica de renombramiento de registros para evitar conflictos de dependencia de datos, y tienen técnicas avanzadas en predicción de saltos.

Final 2:

1. Describir el pasaje de argumentos a subrutinas:

El pasaje de argumentos a subrutinas se utiliza para comunicar datos entre el programa principal y la subrutina. Las formas de pasajes de parámetros son:

Via Registros: Los registros tienen una ventaja, los registros son de rápido acceso. En este método, los datos se cargan en los registros antes de invocar a la subrutina y dentro de la subrutina acceden a los datos a través de los registros. La única desventaja es que los registros son limitados. Este método puede ser eficiente cuando se tiene una pequeña cantidad de argumentos.

Via Memoria: Los datos se almacenan en direcciones especificas de la memoria RAM. Dentro de la subrutina se accede a las direcciones de memoria para acceder a los datos. Este método es de acceso mas lento que los registros, pero puede ser eficiente cuando se tiene una cantidad mayor de argumentos.

Via Pila (stack): Se considera el verdadero pasaje de parámetros. En este método, los datos son apilados antes de invocar a la subrutina, y dentro de la subrutina se desapilan para acceder a los datos. Existen instrucciones para apilar y desapilar.

Este método puede resultar mas lento que los anteriores, debido a las operaciones de apilado y desapilado.

1. Describir el PIC

El pic se encarga de gestionar las interrupciones, permitiendo que multiples dispositivos y periféricos compartan líneas de interrupción del procesador. El PIC es el responsable de recibir señales de interrupcion de varios dispositivos, como periféricos y dirigir estas solicitudes a la CPU. El PIC, se puede configurar para asignar prioridades a las interrupciones, además, podría configurarse para enmascarar (deshabilitar) algunas interrupciones. Por eso se dice que el PIC es programable. El PIC puede operar las instrucciones internamente o en modo cascada. El PIC cuenta con registros internos como son:

IRR: Especifica las interrupciones que están pendientes de reconocimiento

ISR: Especifica las interrupciones que están siendo atendidas.

IMR: Especifica que interrupciones están enmascaradas y no enmascaradas.

EOI: Indica el final de una interrupcion. Cuando una interrupcion finaliza, se manda una senal de EOI al PIC.

1. Conceptos de la memoria cache:

El objetivo de la memoria cache es lograr que la velocidad de la memoria sea lo mas rápida posible. La cache almacena copia de partes de la memoria principal. Se conecta al procesador a través de líneas de datos, control y direcciones.

La memoria cache se organiza en bloques o líneas de cache, un bloque representa la cantidad de datos transferidos entre la memoria principal y la cache en una única operación.

La memoria cache se basa en el principio de localidad, que se divide en dos tipos:

Localidad Temporal: Es posible que la misma dirección de memoria sea referenciada, e accedida más de una vez.

Localidad Espacial: Al referenciarse una dirección de memoria, es posible que se acceda a una dirección cercana en un futuro.

1. Describir la técnica de renombramientos de registros

La técnica de renombramiento de registros que permite ejecutar instrucciones en paralelo de manera eficiente. Esta técnica consiste en que, al ejecutar las instrucciones, se le asigna un registro físico en vez de un registro lógico. Esto permite que multiples instrucciones usen los mismos registros lógicos sin interferencias.

Cuando la ejecución de una instrucción se completa, el resultado se desaloja del registro físico y se escribe en el registro lógico. El objetivo de usar la técnica de renombramiento de registros es evitar conflictos por dependencias de datos.

1. Caracteristicas de un bus

Un bus, es un camino de comunicación entre dos o mas dispositivos. Un bus, se usa como medio de transmisión compartido, puede ser transmisión de datos, de direcciones, señales de control, entre diferentes componentes del sistema.

Al bus se conectan varios dispositivos, cualquier señal emitida en el bus, va a estar disponible para todos los dispositivos conectados.

Dado que en un bus, solo un dispositivo puede transmitir en un momento dado con éxito, se necesita un control o un arbitraje. De esta manera, hay dos tipos de arbitrajes:

*Arbitraje Centralizado:* Un dispositivo, conocido como controlador o arbitro del bus, es el encargado de asignar el acceso al bus a los distintos dispositivos.

*Arbitraje Distribuido:* En este tipo de arbitraje, no hay un controlador, sino que cada modulo dispone de la lógica necesaria para controlar y tomar el acceso del bus. En este tipo de arbitraje, los módulos deben trabajar coordinadamente

*A medida que aumentan los dispositivos conectados al bus, el retardo de propagación también aumenta.*

Hay dos tipos de buses:

Buses Sincronicos:

* + La transmisión de datos esta sincronizada por un reloj común para todos los dispositivos conectados al bus.
  + Los datos se transmiten en intervalos determinados por un pulso de reloj
  + Todos los dispositivos conectados al bus, deben operar a la misma frecuencia para evitar problemas de sincronización.

Buses Asincronicos:

* + Cada dispositivo puede operar a su propio ritmo.
  + Los datos se envían de manera independiente, sin la necesidad de depender de una señal de reloj.

1. Que es el MIMD de la Taxonomía de Flynn

En un sistema MIMD, múltiples procesadores independientes, ejecutan instrucciones al mismo tiempo en conjunto distintos de datos. Cada procesador tiene su propio flujo y puede ejecutar instrucciones de manera independiente. Ademas en un sistema MIMD, los procesadores pueden trabajar en conjunto para resolver problemas mas complejos. Los procesadores en un sistema MIMD son capaces de ejecutar tareas en paralelo.

Los sistemas MIMD pueden clasificarse en dos categorías:

***MIMD con memoria compartida:*** Varios procesadores comparten un espacio de memoria en común, y se pueden comunicar a través de esta memoria con operaciones de lectura y escritura.

***MIMD con memoria distribuida:*** Cada procesador tiene su propia memoria local, y no comparten espacio de memoria con otros procesadores. La comunicación se puede dar por mensajes o cualquier otro mecanismo de comunicación.

Final 3:

1. Explicar las formas de direccionamiento de e/s:

El direccionamiento de e/s se refiere a como el procesador accede a los dispositivos periféricos para enviar o recibir datos.

Algunas formas de direccionamiento de e/s son:

***Direccionamiento por Puertos( I / O PORTS):*** Implica asignar un conjunto de direcciones de memoria especificas a los puertos de e/s. Cada puerto se considera una ubicación de memoria única, a la que el procesador puede acceder para enviar o recibir datos desde o hacia dispositivos periféricos.

***Direccionamiento por Mapeo de Memoria:*** Los registros de control y estado de los periféricos se asignan a direcciones de memoria especificas. El procesador utiliza instrucciones de carga y almacenamiento (LOAD / STORE) para interactuar con los periféricos.

***Interrupciones de e/s:*** El procesador utiliza interrupciones para ser notificado de eventos en los dispositivos. Cuando un dispositivo requiere atención del procesador, genera una interrupcion y el procesador interrumpe su ejecución en curso para atender la interrupcion.

***Acceso Directo a Memoria (DMA):*** Permite que los dispositivos periféricos transfieran datos hacia o desde la memoria, sin la necesidad de la intervención directa del procesador. El controlador de DMA (DMAC) es el encargado de llevar a cabo las transferencias, liberando a la CPU de la tarea de gestionar cada transferencia, de esta manera, el DMA reduce la carga del procesador.