Arquitectura de computadores – Final

Introducción:

La arquitectura de computadoras se refiere a los atributos de un sistema que son visibles a un programador. Entre los ejemplos de atributos se encuentran el conjunto de instrucciones, mecanismos de e/s y técnicas para direccionamiento de memoria,

Ciclo de instrucción:

Un ciclo de instrucción consiste en la captación de la instrucción, seguida de ninguno o varios accesos a operadnos, ninguno o varios almacenamientos de operandos y la comprobación de las interrupciones (si están habilitadas).

El procesamiento que requiere una instrucción se denomina ciclo de instrucción.

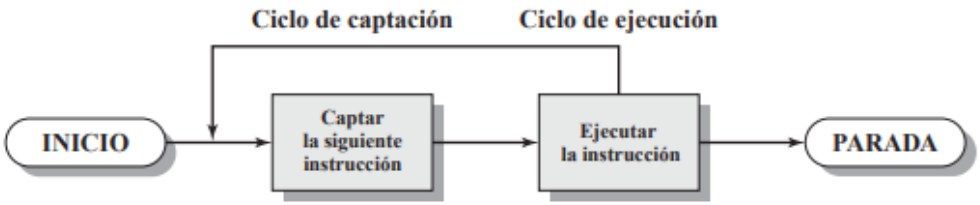


Figura 3.3. Ciclo de instrucción básico.

Al comienzo de cada ciclo de instrucción, la CPU capta una instrucción de memoria, se utiliza un registro llamado PC (program counter) para saber cuál es la próxima instrucción a ejecutar. (A no ser que se indique otra cosa, la CPU siempre incrementa el PC después de captar cada instrucción).

La instrucción captada se almacena en un registro de la CPU, el cual es el IR (Instruction Register)

Principales componentes de la computadora:

Procesador, memoria principal, módulos de e/s: Necesitan estar interconectados para intercambiar datos y señales de control. El medio de comunicación mas popular es un bus compartido constituido por un conjunto de líneas. En las computadoras actuales, es usual usar una jerarquía de buses para mejorar el nivel de prestaciones.

Interrupciones:

Una **interrupción** es un mecanismo que **suspende temporalmente la ejecución normal de la CPU** para atender un evento específico que requiere su atención inmediata. Un dispositivo externo o una condición interna señala al procesador para interrumpir su flujo de ejecución actual y responder a un evento específico.

Tipos de interrupciones:

Interrupciones internas ((INTERRUPCIÓN DE SOFTWARE)): Generadas como resultado de la ejecución de una instrucción del programa actual. Por ejemplo overflow, dividir por cero, intentar ejecutar una instrucción inexistente, o intentar acceder fuera del espacio de memoria permitido. Estas interrupciones son sincrónicas porque ocurren como resultado de la ejecución del programa actual.

Manejo típico de las interrupciones internas: El SO invoca una rutina para gestionar el error (puede ser mostrar un mensaje de error, detener el programa,etc).

Interrupciones externas ((INTERRUPCIÓN DE HARDWARE)): Generadas por eventos externos al procesador, como dispositivos periféricos o señales de control. Por ejemplo interrupción por temporización (un reloj interno genera interrupciones periódicas para que el sistema realice algún tipo de tarea.), otro ejemplo son los controladores de e/s (un dispositivo periférico indica que ha terminado una operación de e/s o que requiere atencion). Estas interrupciones son asincrónicas, no están vinculadas a la ejecución del programa actual, sino que este tipo de interrupciones pueden ocurrir en cualquier momento.

Manejo típico de las interrupciones externas: Se ejecuta una rutina especifica asociada al evento externo, por ejemplo leer datos de un disco, o procesar una entrada del usuario.

Interrupciones por fallo del hardware: Generadas por un fallo como falta de potencia de alimentación, o un error de paridad en la memoria (pocos comunes)

¿Por qué son necesarias las interrupciones?

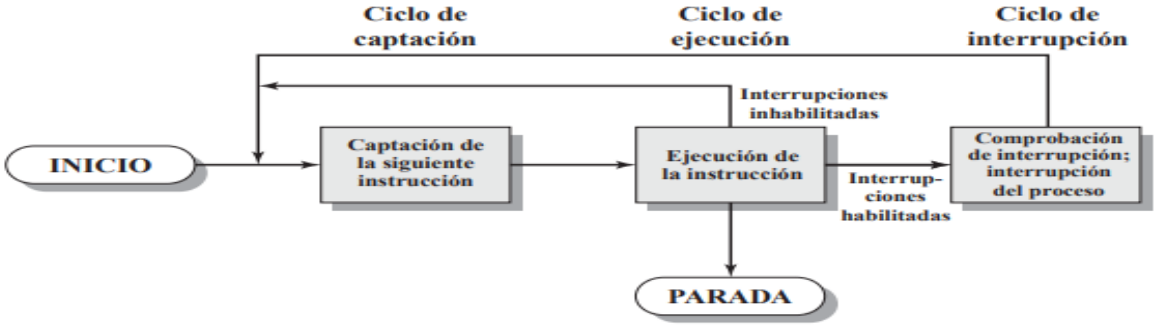
1. **Mejoran la eficiencia del procesador**:
   - Sin interrupciones, el procesador tendría que emplear técnicas como el *polling*, donde revisa continuamente el estado de los dispositivos. Esto resulta ineficiente porque consume ciclos de reloj incluso cuando no hay eventos que manejar.
   - Con las interrupciones, el procesador puede concentrarse en otras tareas y responder solo cuando ocurre un evento.
2. **Permiten la multitarea**:
   - Las interrupciones son esenciales para implementar sistemas operativos multitarea. Estas permiten cambiar entre procesos (context switching) y atender múltiples tareas aparentemente al mismo tiempo.
3. **Soporte para dispositivos de E/S más lentos**:
   - La mayoría de los dispositivos de E/S son más lentos que el procesador. Las interrupciones permiten que el procesador no desperdicie tiempo esperando que un dispositivo esté listo.
4. **Respuesta inmediata a eventos críticos**:
   - Permiten manejar eventos urgentes, como señales de hardware críticas (fallas en la alimentación eléctrica, fallas de hardware, etc.), sin demoras.

Interrupciones y el ciclo de instrucción:

Con el uso de interrupciones, el procesador puede dedicarse a ejecutar otras instrucciones mientras una operación de E/S está en curso.

Cuando el dispositivo externo pasa a estar preparado para actuar, es decir, cuando está listo para aceptar más datos del procesador, el módulo de E/S de este dispositivo externo envía una señal de petición de interrupción al procesador. El procesador responde suspendiendo la operación del programa que estaba ejecutando y salta a un programa, conocido como gestor de interrupción.

El ciclo de instrucción con interrupciones es similar al ciclo básico, pero incluye una etapa adicional al final de cada instrucción: después de ejecutar una instrucción y antes de captar la siguiente, el procesador verifica si las interrupciones están habilitadas y si hay una interrupción pendiente. Si ambas condiciones se cumplen, suspende temporalmente la ejecución del programa principal, guarda su contexto, atiende la interrupción ejecutando la rutina correspondiente y luego restaura el contexto para continuar con la ejecución del programa que se interrumpió.



Explicado desde el libro:

En el ciclo de interrupción, el procesador comprueba si se ha generado alguna interrupción, indicada por la presencia de una señal de interrupción.

- Si no hay señales de interrupción: El procesador continua el ciclo de captación y accede a la siguiente instrucción del programa en curso.
- Si hay alguna interrupción pendiente, el procesador hace lo siguiente:
    - Suspende la ejecución del programa en curso y guarda su contexto (guarda la dirección de la próxima instrucción a ejecutar)
    - Carga el PC (program counter) con la dirección de comienzo de una rutina de gestión de interrupción.

Tratamiento de múltiples interrupciones:

Existen dos alternativas principales para tratar las interrupciones múltiples:

Primera alternativa: Desactivar las interrupciones
En este enfoque, se desactivan las interrupciones mientras se está procesando una interrupción. Una vez deshabilitadas, el procesador ignora cualquier nueva señal de interrupción hasta que se vuelvan a habilitar. Si ocurre una interrupción mientras está desactivada, se mantiene pendiente y se examina nuevamente cuando las interrupciones se reactiven.

- **Ventajas:** Esta solución es simple y efectiva, ya que las interrupciones se manejan de manera secuencial.
- **Desventajas:** No tiene en cuenta la prioridad de las interrupciones, lo que puede generar un retraso en el manejo de interrupciones importantes si no se gestionan adecuadamente.

Segunda alternativa: Definir prioridades a las interrupciones
Este enfoque permite que las interrupciones de mayor prioridad interrumpan a aquellas de menor prioridad. Así, una interrupción con mayor urgencia puede interrumpir el procesamiento de una interrupción de menor prioridad, asegurando que se atiendan las tareas más críticas.

- **Ventajas:** Las interrupciones de mayor prioridad son atendidas rápidamente, lo que es crucial en sistemas donde las tareas de alta prioridad deben ser resueltas con urgencia.
- **Desventajas:** Las interrupciones de menor prioridad pueden sufrir inanición, ya que si constantemente se presentan interrupciones de mayor prioridad, estas nunca se procesan.

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, se pueden inhabilitar mediante un mecanismo del procesador, lo que permite al SO controlar cuando se atienden y cuando no. Generalmente las interrupciones enmascarables no son interrupciones de alta prioridad, ni interrupciones criticas que deben ser atendidas de inmediato.

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. Son atendidas por el procesador, indican eventos críticos como condiciones de fallo del sistema.

ENTRADA / SALIDA

Un modulo de E/S, también conocido como controlador de E/S, actúa como intermediario entre el procesador, la memoria principal y los dispositivos externos.

La función principal de un modulo de e/s es facilitar la interacción 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 y detectar posibles errores para mantener la integridad y confiabilidad del sistema. El modulo de e/s es capaz de reconocer y generar direcciones asociadas a los periféricos que controla.

Funciones de un modulo de e/s:

- Control y temporización:
    - Coordinan el flujo de datos entre el procesador, la memoria y los dispositivos periféricos. Debido a las diferencias de velocidad entre estos componentes, el modulo de e/s debe sincronizar las operaciones para garantizar una transferencia de datos eficiente.
- Comunicación con el procesador:
    - Decodifican las ordenes enviadas al procesador.
    - Reportan el estado del dispositivo al procesador.
    - Manejan las interrupciones generadas por los dispositivos.
- Comunicación con los dispositivos periféricos:
    - Envían y reciben datos entre el dispositivo y el modulo de e/s

Diseño de un modulo de e/s

Un modulo de e/s consta de los siguientes componentes:

- Registros de datos:
    - Almacenan temporalmente los datos que se transfieren entre el dispositivo y el procesador.
- Registro de estado / control:
    - Proporcionan información sobre el estado actual del modulo y del dispositivo.
- Logica de interfaz con el bus del sistema
    - Permite la comunicación con el procesador y la memoria principal a través del bus del sistema.

Conclusión de modulos de E/S: Los módulos de e/s son esenciales para el funcionamiento de un sistema de computo, permitiendo la interacción con el mundo exterior y la utilización de una amplia variedad de dispositivos periféricos.

Estructura de un modulo de e/s:

La estructura de un modulo de e/s, esta compuesta por:

- **Interfaz al bus del sistema**: El modulo se conecta al resto del sistema, a través de un conjunto de líneas:
    - Lineas de dirección: Se utilizan para identificar el modulo de e/s y los dispositivos que controla.
    - Lineas de datos: Proporcionan un camino para la transferencia bidireccional de datos entre el modulo de e/s y el procesador o la memoria. El ancho del bus, determina cuantos bits se pueden transferir simultáneamente.
    - Lineas de control: Se utilizan para coordinar las operaciones entre el procesador y el modulo de e/s.
- **Registros:**
    - Registros de datos: Almacenan temporalmente datos que se transfieren entre el periférico y el resto del sistema.
    - Registro de estado: Proporcionan información sobre el estado actual del modulo de e/s.
    - Registro de control: Se utiliza para configurar el modo de operación del modulo de e/s.
- **Interfaz a dispositivos externos:** Contiene la lógica y los circuitos específicos necesarios para la comunicación con cada uno de los dispositivos externos que controla el modulo.

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**

Acceso Directo a Memoria (DMA): El DMA imita al procesador. 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.

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.

Bus:

Un bus es un camino de comunicación entre dos o mas dispositivos.

La característica clave de un bus es que varios dispositivos se conectan a el, y cualquier señal transmitida por uno de ellos, está disponible para todos los demás. Si dos dispositivos transmiten durante el mismo periodo de tiempo, sus señales pueden solaparse y distorsionarse, solo un dispositivo puede transmitir con éxito en un momento dado.

Aspectos clave del diseño de buses:

Los aspectos clave del diseño de buses son el arbitraje (si el permiso para enviar señales a través del bus es de forma distribuida o centralizada), la temporización (si las señales del bus se sincronizan mediante un reloj o se envían asincrónicamente) y la anchura (numero de líneas de dirección y datos)

Estructura del bus:

Un bus está compuesto por múltiples líneas, cada una capaz de transmitir señales binarias (1 o 0). Estas líneas se agrupan en tres categorías funcionales:

●**Líneas de datos:** Transmiten los datos entre los módulos del sistema. El número de líneas de datos se denomina "anchura del bus de datos" cada línea solo puede transportar un bit a la vez , por lo que determina cuántos bits se pueden transferir simultáneamente.

●**Líneas de direcciones:** Indican la fuente o el destino de los datos en el bus de datos. La anchura del bus de direcciones determina la capacidad máxima de memoria direccionable.

●**Líneas de control:** Controlan el acceso y uso del bus, incluyendo señales para arbitraje, temporización y estado del dispositivo.

Jerarquia de buses

Si se conecta un gran numero de dispositivos al bus, las prestaciones pueden disminuir, hay dos causas principales:

1) A mas dispositivos conectados al bus, mayor es el retardo de propagación
2) El bus puede convertirse en un cuello de botella a medida que las peticiones de transferencia acumuladas se aproximan a la capacidad del bus. (este problema se puede resolver incrementando la velocidad a la que el bus puede transferir datos, se podría usar buses mas anchos)

Se suele emplear una jerarquía de buses para mejorar el rendimiento.

La mayoría de las computadoras utilizan varios buses, normalmente organizados jerárquicamente. Un bus principal,llamado bus del sistema, conecta los componentes principales (CPU,memoria,e/s). Otro tipos de buses pueden conectar dispositivos de alta velocidad (como discos duros) al procesador o la memoria, formando una jerarquía de buses.

La jerarquía de buses permite optimizar el rendimiento al proporcionar caminos de comunicación especializados para diferentes tipos de dispositivos y velocidades de transferencia.

Funcionamiento del bus

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

3) Obtener el uso del bus.
4) Transferir el dato a través del bus.

Si un modulo desea pedirle un dato a otro modulo debe:

1) Obtener el uso del bus
2) Transferir la petición al otro modulo mediante las líneas de control y dirección apropiadas.

Elementos para el diseño de un bus

Tipo	**Anchura del bus**
Dedicado	Dirección
Multiplexado	Datos
Método de arbitraje	**Tipo de transferencia de datos**
Centralizado	Lectura
Distribuido	Escritura
Temporización	Lectura-modificación-escritura
Síncrono	Lectura-después de-escritura
Asíncrono	Bloque

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.

Metodos de arbitraje

En la mayoría de los sistemas, más de un modulo necesita el control del bus. Dado que en un instante dado, solo un dispositivo puede transmitir con éxito en el bus, se requiere algún método de arbitraje.

Los métodos se clasifican en centralizados o distribuidos.

Arbitraje centralizado: Un único dispositivo denominado controlador del bus, o arbitro, es el responsable de asignar tiempos en el bus.

Arbitraje distribuido: No existe un controlador, sino que cada modulo dispone de la lógica necesaria para controlar el acceso, y los modulos actúan conjuntamente para compartir el bus.

Temporización:

El termino temporización hace referencia a la forma en la que se coordinan los eventos en ell bus. Los buses utilizan temporización síncrona o asíncrona.

Temporización síncrona: La presencia de un evento en el bus, esta determinada por un reloj. El bus incluye una línea de reloj a través de la que se transmite una secuencia en la que se alternan intervalos regulares de igual duración a uno y a cero.

Temporización asíncrona: La presencia de un evento en el bus es consecuencia y depende de que se produzca un evento previo.

Anchura del bus

La anchura del bus de datos afecta a las prestaciones del sistema. Cuanto mas ancho es el bus de datos, mayor es el numero de bits que se transmiten a la vez. La anchura del bus de direcciones afecta a la capacidad del sistema.

Cuanto más ancho es el bus de direcciones => Mayor es el rango de posiciones a las que se puede hacer referencia.

●**Número de líneas de dirección:** Determina la cantidad de memoria que puede direccionar el sistema.
●**Número de líneas de datos:** Define cuántos bits se pueden transferir simultáneamente, afectando la velocidad de transferencia.

Tipo de Transferencia de Datos:

●**Lectura:** Transferencia de datos del esclavo al maestro.
●**Escritura:** Transferencia de datos del maestro al esclavo.
●**Lectura-modificación-escritura:** Una operación indivisible para leer, modificar y escribir datos en una ubicación de memoria.
●**Lectura-después-de-escritura:** Una operación indivisible para escribir un dato y luego leerlo para verificar.
●**Transferencia de bloque:** Un ciclo de dirección seguido de varios ciclos de datos para transferir un bloque de información.

Bus PCI

El bus PCI es **independiente del procesador**, de **ancho de banda elevado** que **permite conectar y comunicar periféricos con la placa base**. El bus PCI **proporciona una interfaz de alta velocidad**, permitiendo la transferencia de datos rápida entre los periféricos y el procesador o la memoria.

El PCI está diseñado para permitir una cierta variedad de configuraciones basadas en microprocesadores, incluyendo sistemas tanto de uno como de varios procesadores. Proporciona un conjunto de funciones de uso general. **Utiliza temporización síncrona y un esquema de arbitraje centralizado.**

El bus PCI, utiliza una variedad de órdenes para realizar transferencias de datos. El maestro del bus inicia la transferencia, el arbitro del bus gestiona el acceso al bus y las diferentes fases de transferencia aseguran la correcta transmisión de datos entre los procesadores.

Memoria

Métodos de acceso:

Acceso Secuencial: La memoria se organiza en unidades de datos llamadas registros. El acceso a estos registros debe realizarse de manera secuencial, con una secuencia lineal especifica.

Acceso Directo: El acceso se lleva a cabo mediante un acceso directo a una vecindad dada, seguido de una búsqueda secuencial.

Acceso Aleatorio (Random): Cada posición direccionable de memoria tiene un único mecanismo de acceso cableado físicamente.

Jerarquia de memoria

Las restricciones del diseño de la memoria de una computadora, se puede resumir en tres cuestiones:

¿cuánta capacidad? ¿cómo de rápida? ¿de qué costo?

• A menor tiempo de acceso, mayor coste por bit.

• A mayor capacidad, menor coste por bit.

• A mayor capacidad, mayor tiempo de acceso

Memoria Principal (RAM): La memoria RAM almacena temporalmente datos e instrucciones que la CPU necesita en tiempo real. Este tipo de memoria son volátiles, lo que significa que pierde su contenido cuando se apaga la computadora.

Hay dos tipos de memoria RAM => SRAM , DRAM:

Tanto la SRAM (RAM estatica) como la DRAM (RAM dinamica), son tipos de memoria de acceso aleatorio (RAM). Los discos, por otro lado, son dispositivos de almacenamiento externo que proporcionan una capacidad mucho mayor que la memoria principal.

Memoria Secundaria: Se utiliza para guardar datos de forma no volátil. Es decir, retiene su información, incluso cuando la computadora se encuentra apagada. **Un ejemplo de almacenamiento secundario es el disco.**

Disco: Tiene un costo de acceso mucho mayor que la RAM, por ende, acceder a un disco es costoso por que los datos se graban y luego se recuperan del disco a través de una bobina llamada cabezal, por genera un costo extra para posicionar el cabezal de lectura/escritura.

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.

Objetivo de la jerarquía de memoria

El objetivo de la jerarquía de memoria es abordar las limitaciones de diseño de la memoria de una computadora, que son la capacidad, la velocidad y el coste.

Es deseable tener una memoria con una gran capacidad,alta velocidad, y bajo coste. Sin embargo es difícil y costoso construir una memoria que cumpla con estos tres requisitos al mismo tiempo.

La jerarquía de memoria resuelve este problema utilizando múltiples niveles de memoria con diferentes características de coste,capacidad y velocidad.

A medida que se desciende en la jerarquía:

- Disminuye el costo por bit
- Aumenta la capacidad
- Aumenta el tiempo de acceso.

Conclusión: La jerarquía de memoria permite obtener un buen equilibrio entre coste, capacidad y velocidad, aprovechando el principio de localidad para minimizar el tiempo de acceso medio a los datos e instrucciones.

Memoria Caché

El objetivo de la memoria caché, es lograr que la velocidad de la memoria sea lo mas rápida posible. La caché contiene una copia de partes de la memoria principal. Cuando el procesador intenta leer una palabra de memoria, se hace una comprobación para determinar si la palabra está en la caché. Si es así, se entrega la palabra al procesador. Si no, un bloque de memoria principal, se transfiere a la cache y, después, la palabra es entregada al procesador

La caché se organiza en bloques (también puede llamarse líneas de caché). Un bloque representa la cantidad de datos transferidos entre la memoria principal y la caché en una única operación.

Acceso a los datos en memoria caché:

La caché se basa en dos principios de localidad:

1) **Principio de localidad referencial**: Establece que los programas tiendes a acceder repetidamente a las mismas posiciones de memoria, o los mismos datos durante un periodo corto de tiempo.
    - Si un dato o instrucción se utiliza una vez, es probable que se vuelva a utilizar en un futuro .
2) **Principio de localidad espacial**: Si un dato o instrucción se utiliza, es probable que los datos o instrucciones cercanas (en direcciones de memoria), también se utilicen en un futuro cercano.

Elementos de diseño de caché

Tamaño de caché	**Política de escritura**
Función de correspondencia	Escritura inmediata
Directa	Postescritura
Asociativa	Escritura única
Asociativa por conjuntos	**Tamaño de línea**
Algoritmo de sustitución	**Número de cachés**
Utilizado menos recientemente (LRU)	Uno o dos niveles
Primero en entrar-primero en salir (FIFO)	Unificada o partida
Utilizado menos frecuentemente (LFU)	
Aleatorio	

Formas de correspondencia de la caché

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.

Politica de escritura en cache:

Cuando se produce una operación de escritura, surge la cuestión de si actualizar la ubicaciob de la memoria principal al mismo tiempo que la palabra en caché.

Para entender las políticas de escritura:

Acierto de caché: Ocurre cuando el procesador intenta leer una palabra de la memoria y la palabra se encuentra en la caché. En este caso, la palabra se entrega al procesador desde la caché. Lo que es mucho mas rápido que acceder a memoria principal.

Fallo de caché: Ocurre cuando el procesador intenta leer una palabra de la memoria y la palabra no se encuentra en la caché. En este caso, se produce un fallo de caché, y se debe acceder a memoria principal para obtener la palabra.

Politicas de escritura en acierto de caché: Determina si los datos se escriben solo en la caché, o tanto en la caché como en la memoria principal.

Escritura inmediata: La información se escribe tanto en la caché como en la memoria principal. Esta es la técnica más sencilla .Garantiza que la memoria principal esté siempre actualizada, pero puede generar un mayor tráfico en el bus de memoria.

Postescritura: 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 fallo de caché: Determina si el bloque de memoria principal que contiene la palabra a escribir se carga en la caché o no.

Escritura y asignación: Ante un fallo de escritura, el bloque de memoria principal se carga primero en la caché y luego se realiza la escritura en la caché.

No escritura y asignación: Escribe los datos directamente en la memoria principal sin cargar el bloque en la caché. Se utiliza para reducir el trafico en el bus de memoria.

Tamaño de línea:

El tamaño de línea se refiere a la cantidad de datos que se transfieren de la memoria principal a la caché en cada acceso. Cuando se recupera y ubica en caché un bloque de datos, se recuperan no sólo la palabra deseada sino además algunas palabras adyacentes

• Bloques más grandes reducen el número de bloques que caben en la caché. Dado que cada bloque captado se escribe sobre contenidos anteriores de la caché, un número reducido de bloques da lugar a que se sobrescriban datos poco después de haber sido captados.

• A medida que un bloque se hace más grande, cada palabra adicional está más lejos de la requerida y por tanto es más improbable que sea necesaria a corto plazo

Numero de cachés:

Las caches multinivel son una técnica para mejorar el rendimiento al reducir el tiempo promedio de acceso a la memoria. Se trata de una jerarquía de cachés. Donde cada nivel es mas grande y lento que el anterior, pero mas rápido que la memoria principal.

Cuando el procesador necesita acceder a un dato, primero busca en la caché nivel1 (L1), la mas pequeña y rápida, si el dato se encuentra, se produce un acierto de caché, el acceso es muy rápido. Si el dato no se encuentra, se busca en la caché de nivel2 (L2) que es mas grande y lenta que la L1, pero mas rápida que la memoria principal, si el dato no se encuentra en L2, se produce un fallo y se accede a la memoria principal. Una **ventaja** de usar esta técnica es que reduce el tiempo promedio de acceso a la memoria.

Que es una subrutina?

Una subrutina es una secuencia de instrucciones, que realiza una tarea especifica, puede ser llamada desde diferentes partes de un programa. Una subrutina puede contener llamadas a otras subrutinas. Las subrutinas pueden recibir parámetros.

Requieren dos instrucciones: CALL (invocarla) y RETURN (retornar de la subrutina), ambas instrucciones son de bifurcación.

Cuando se llama a unas subrutina, el procesador guarda la dirección de retorno, que es utilizada para volver al programa principal una vez que se haya terminado de ejecutar la subrutina.

Ventajas de las subrutinas:

Permiten reutilizar código, reduce el tamaño de los programas almacenados en memoria, además las subrutinas aportan modularidad, al descomponer problemas grandes en partes mas pequeñas.

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 múltiples 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. Además, permite habilitar o deshabilitar (enmascarar) las interrupciones. Cuando una interrupción esta enmascarada, se ignorará. Además 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.

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 vía 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.

Segmentación de Cauce y segmentación de instrucciones:

(Definición del libro): Los procesadores utilizan la segmentación de cauce de instrucciones para acelerar la ejecución. LA segmentación de cause permite dividir el ciclo de instrucción en varias etapas separadas que operan secuencialmente. Cada etapa puede estar trabajando en una instrucción diferente al mismo tiempo. Las instrucciones se van ejecutando a medida que se liberan unidades.

La **segmentación de cauce** es una técnica de mejora las prestaciones a nivel de diseño de hardware 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. **IMPORTANTE: EN LA SEGMENTACION DE CAUCE, CADA ETAPA REQUIERE AL MENOS UN CICLO DE RELOJ COMPLETO.**

Al estar ejecutando múltiples instrucciones en paralelo, pueden producirse conflictos/atascos.

Los conflictos o atascos son situaciones que impiden que la siguiente instrucción se ejecute en el ciclo que le corresponda.

Conflicto	Solución
Dependencia de datos: Ocurren cuando dos o mas instrucciones comparten un mismo dato, y una instrucción necesita el resultado de otra.	- Reordenamiento de instrucciones - forwarding (adelantamiento de operandos).
Estructurales: Ocurren cuando dos o más partes del hardware compiten por el mismo recurso. Es decir, esta provocado por el uso de recursos, como memoria, alu, y registros.	- Duplicar los recursos de hardware que generan conflictos. - Segmentar los recursos - Realizar turnos para acceder a los recursos del hardware que generan conflictos.
Dependencia de control: Surgen cuando la ejecución de una instrucción depende de cómo se ejecute otra. Ejemplo: 1 salto y 2 posubkes caminos.	- Tecnicas de software: Salto retardado - Tecnica de hardware: Prediccion de saltos

Segmentacion de instrucciones:

La segmentación de instrucciones es una técnica empleada en los procesadores modernos, para mejorar el rendimiento mediante el paralelismo en la ejecución de instrucciones. En la segmentación de instrucciones, el ciclo de instrucción se divide en varias etapas que se ejecutan de forma solapada, similar a un pipeline por donde pasan las instrucciones.

Conclusión:

En un procesador sin segmentación, estas etapas se ejecutan secuencialmente. Con la segmentación, mientras una instrucción se está ejecutando, la siguiente instrucción ya se esta captando. Esto permite reducir el tiempo total necesario para ejecutar una secuencia de instrucciones.

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.

Procesadores

Procesador Superescalar: Se caracteriza por el uso de multiples cauces de instrucciones independientes, cada cauce consta con multiples etapas, lo que permite procesar varias instrucciones a la vez. Este tipo de procesadores, explota el paralelismo a nivel de instrucción. **Ejemplo: El Pentium es un ejemplo de procesador superescalar.**

Para mejorar las prestaciones en procesadores superescalares, se pueden utilizar técnicas como predicción de saltos y renombramiento de registros.

Procesador Supersegmentado: Divide las etapas del cauce en partes mas pequeñas, permitiendo ejecutar dos o mas tareas en cada ciclo de reloj. Esto permite que mas instrucciones se encuentren en el cauce al mismo tiempo, aumentando el paralelismo. En la segmentación de cauce, cada etapa requiere un ciclo de reloj completo, en la supersegmentacion, cada etapa se divide en subetapas que pueden ejecutarse en fracciones de ciclo. Al duplicar la velocidad del reloj interno, se pueden realizar dos tareas en un ciclo de reloj externo, lo que aumenta ademas, el nivel de velocidad de procesamiento.

Conclusion y diferencias entre superescalar y supersegmentado.

Superescalar: Diseñado para explotar el paralelismo a nivel de instrucción, ejecutando varias instrucciones al mismo tiempo en diferentes unidades funcionales. Es menos complejo de implementar. Cada instrucción requiere al menos un ciclo de reloj. **Su objetivo principal es maximizar la cantidad de instrucciones procesadas en paralelo.**

Supersegmentado: Ademas de explotar el paralelismo, se enfoca en aumentar la velocidad de procesamiento dividiendo cada instrucción en mas etapas de pipeline. Es mas complejo de implementar. Permite realizar dos tareas en un ciclo de reloj. **Su objetivo principal es procesar mas instrucciones por unidad de tiempo.**

Procesadores CISC y RISC

CISC (Conjunto de instrucciones complejo): Buscan simplificar la programación al proporcionar un repertorio amplio y complejo de instrucciones, incluyendo operaciones de alto nivel que pueden realizar tareas complejas en un solo ciclo de instrucción. Posee gran variedad de modos de direccionamiento. Es usado en lenguajes de alto nivel.

Ventajas: Programas más compactos, al tener instrucciones más potentes se pueden escribir programas más cortos y con menos líneas de código. Además tiene menor necesidad de acceso a memoria, ya que las instrucciones complejas pueden realizar múltiples operaciones sin necesidad de acceder a la memoria principal con tanta frecuencia. **Intel Pentium es un ejemplo de procesador con arquitectura CISC.**

Desventajas: Ciclos de instrucciones más largos, las instrucciones más complejas pueden requerir mas de un ciclo de reloj para poder ejecutarse. Además, la decodificación y ejecución de las instrucciones requiere un hardware mas complejo y costoso.

RISC (Conjunto de instrucciones reducido): Este tipo de procesadores priorizan la simplicidad y velocidad. Su repertorio de instrucciones es pequeño/reducido, y se compone de **instrucciones sencillas que se ejecutan en un ciclo de reloj.** Posee un número limitado de modos de direccionamiento, pero con la diferencia que posee una gran cantidad de registros de propósito general. **Este tipo de procesadores, hace énfasis en segmentación de instrucciones para ejecutar múltiples instrucciones en paralelo.**

Ventajas: Requiere hardware mas simple, y mas económico. Tiene ciclos de instrucciones mas cortos, ya que solo requieren un ciclo de reloj.

Desventajas: Se necesitan mas instrucciones para realizar tareas mas complejas, lo que puede resultar en programas largos. Ademas, al tener instrucciones mas simples, requiere un mayor numero de accesos a la memoria principal.

Repertorio de instrucciones: Características y funciones

Los elementos esenciales de las instrucciones de las computadoras son:

- El código de operación (que especifica la operación a realizar)
- Las referencias a operandos origen y destino (que especifican las entradas y salidas para la operacion)
- La referencia a la siguiente instrucción (que la llevamos en el PC)

Codigo de operación: Especifican las operaciones que pueden ser: Operaciones aritméticas y lógicas, transferencia de datos entre dos regis tros, entre registros y memoria, o entre dos posiciones de memoria; entrada/salida (E/S); y control

Diseño del repertorio de instrucciones:

El repertorio de instrucciones define muchas de las funciones realizadas por el procesador. El repertorio de instrucciones es el medio que tiene el programador para controlar el procesador. Se debe considerar las necesidades del programados al momento de diseñar el repertorio de instrucciones.

Los aspectos fundamentales para el diseño de repertorio de instrucciones son:

• **El repertorio de operaciones:** cuántas y qué operaciones considerar, y cuán complejas deben ser.

• **Los tipos de datos:** los distintos tipos de datos con los que se efectúan operaciones.

• **Los formatos de instrucciones**: longitud de la instrucción (en bits), número de direcciones, tamaño de los distintos campos, etc.

• **Los registros:** número de registros del procesador que pueden ser referenciados por las ins trucciones, y su uso.

• **El direccionamiento**: el modo o modos de direccionamiento mediante los cuales puede especificarse la dirección de un operando.

Control de flujo

El control de flujo se refiere a las instrucciones que cambian la secuencia de ejecución normal de un programa. Normalmente, las instrucciones se ejecutan secuencialmente en la memoria. Sin embargo, las instrucciones de control de flujo permiten al programa saltar a diferentes partes del código, lo que es esencial para implementar bucles, condicionales y llamadas a procedimientos

Mecanismos de control de flujo:

- **Instrucciones de bifurcación (saltos):** Las instrucciones de bifurcación, también conocidas como saltos, hacen que el procesador ejecute una instrucción en una dirección diferente a la siguiente en la secuencia. Un operando de la instrucción de bifurcación especifica la dirección de la siguiente instrucción a ejecutar.
  Hay dos tipos principales de bifurcaciones:
  - **Saltos Incondicionales:** El salto siempre se produce, independientemente de cualquier condición. **EJEMPLO INSTRUCCIÓN JMP**
  - **Saltos Condicionales:** El salto se produce solo si se cumple una condición específica. **EJEMPLO INSTRUCCIÓN JZ (SALTA SI ES=0)**

- **Saltos Implícitos:** Estas instrucciones implican un salto a una dirección específica sin necesidad de un operando de dirección. Un ejemplo común es la instrucción "incrementar y saltar si es cero" (ISZ).

- **Llamadas a Procedimientos**: Estas instrucciones transfieren el control a un procedimiento (subrutina) y guardan la dirección de retorno para que el programa pueda volver al punto de llamada después de la ejecución del procedimiento.

Modos de direccionamiento:

El modo de direccionamiento, se refiere al método utilizado para especificar la ubicación de un operando. Un modo de direccionamiento es un esquema especifico para calcular la dirección efectiva de un operando a partir de la información contenida en la instrucción y el estado actual del procesador.

Algunos modos de direccionamientos mas comunes son:

●**Inmediato**: El operando está incluido directamente en la instrucción. Este modo es rápido y eficiente para constantes o valores pequeños, pero limita el tamaño del operando.

●**Directo**: La dirección del operando está especificada explícitamente en la instrucción. Es un modo simple pero limita el rango de direcciones que se pueden alcanzar.

●**Indirecto**: La dirección especificada en la instrucción no contiene el operando, sino la dirección de memoria donde se encuentra la dirección del operando. Permite acceder a un rango más amplio de direcciones, pero requiere un acceso adicional a memoria.

●**De Registros:** El operando se encuentra en un registro del procesador. Es el modo más rápido ya que los registros son internos al procesador, pero el número de registros es limitado.

●**Indirecto con Registro:** La dirección del operando se encuentra en un registro del procesador. Similar al indirecto, pero con la dirección almacenada en un registro, lo que reduce un acceso a memoria.

●**Con Desplazamiento:** La dirección efectiva se calcula sumando un desplazamiento a un valor base, que puede ser una dirección o el contenido de un registro. Ofrece gran flexibilidad y se utiliza en varios modos de direccionamiento más específicos, como el relativo, el de registro base y el indexado.

○**Relativo**: El desplazamiento se suma a la dirección de la instrucción actual, lo que permite acceder a datos cercanos a la instrucción.

○**De Registro Base:** El desplazamiento se suma al contenido de un registro base, lo que permite acceder a datos dentro de un bloque de memoria o estructura de datos.

○**Indexado:** El desplazamiento se suma al contenido de un registro índice, lo que permite acceder a elementos de un array o tabla.

●**De Pila:** El operando se encuentra en la cabecera de la pila, una estructura de datos LIFO utilizada para gestionar llamadas a procedimientos y almacenar datos temporalmente. Es útil para la gestión de subrutinas y para el paso de parámetros.

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.

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.

Clusters:

Los clusters, tienen arquitectura MIMD con memoria distribuida, son grupos de computadoras completas interconectadas que trabajan en conjunto como un único recurso de cómputo, cada computadora es un nodo.. En otras palabras, un cluster es un conjunto de computadoras independientes que colaboran para realizar una tarea en común, dando la apariencia de ser una única máquina más potente.

Características y beneficios clave de los clusters:

●Escalabilidad: Los clusters pueden ser muy grandes, incluso superando el rendimiento de las computadoras individuales más potentes. Se pueden añadir nuevas computadoras a un cluster de forma incremental a medida que sea necesario, lo que permite un crecimiento flexible.

●**Alta disponibilidad:** Si una computadora en un cluster falla, las demás pueden seguir funcionando, lo que minimiza el tiempo de inactividad.

●Relación precio-rendimiento: Los clusters pueden ofrecer un rendimiento similar al de las supercomputadoras a un costo mucho menor.

Arquitecturas de memoria compartida – distribuida

Acceso no uniforme a memoria con coherencia de cache (CC – NUMA): Cada nodo es un SMP, con sus propios procesadores, mecanismo de e/s y memoria principal compartidas. Desde el punto de vista de los procesadores, comparten un único espacio de direcciones.

ARQUITECTURAS PARALELO Y TAXONOMIA DE FLYNN

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

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. Adema sen 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 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.

Procesamiento multihebra(multithreading)

Técnica que consiste en dividir la secuencia de instrucciones en secuencias mas pequeñas, que pueden ejecutarse en paralelo, para permitir paralelismo elevado sin incrementar la complejidad de los circuitos ni el consumo.

Algunas preguntas de finales:

Que elementos componen una maquina con arquitectura Von Neumann? Describir la función de cada uno. (tomada en final de 27/11/2024)

Una maquina con arquitectura Von Neumann se compone de los siguientes elementos:

CPU (Unidad central de procesamiento): La CPU es el cerebro de la computadora. Se encarga de ejecutar las instrucciones de los programas, realizar operaciones, controlar el flujo de datos.

Memoria principal: Almacena tanto los programas como los datos que la cpu necesita para ejecutarlos. Es volátil. Lo que significa que su contenido se pierde al apagar la computadora. El acceso a la memoria principal es aleatorio.

Dispositivos de e/s: Permiten a la computadora interactuar con el mundo exterior. Los dispositivos de e/s, como teclado, mouse, permiten al usuario introducir datos en la computadora. Los dispositivos de salida como el monitor y impresora, muestran los resultados del procesamiento al usuario.

Bus del sistema: Es un camino de comunicación entre dos o mas dispositivos. Conjunto de líneas que conectan diferentes componentes de la computadora, permitiendo la transferencia de datos, direcciones e instrucciones entre la CPU,la memoria principal y los dispositivos de e/s.

Esquematice y describa la estructura interna de un controlador programable de interrupciones (PIC) (tomada en final de 27/11/2024)

(falta el esquema, hay que buscarlo, el esquema lo suelen pedir seguido.)

Cuales son los elementos de diseño de bus?

Tipos de buses: Las lineas del bus se pueden dividir en dos tipos: dedicadas y multiplexadas. Una linea dedicada esta permanentemente asignada a una función o a un subconjunto físico de componentes del computador. Ej.: lineas separadas para direcciones y datos.

Sin embargo, la información de dirección y datos podría transmitirse a través del mismo conjunto de lineas si se utiliza una linea de control. A este método se lo llama multiplexado en el tiempo. La ventaja es que se ahorra espacio y costes. La desventaja es que se necesita una circuitería más compleja.

Método de arbitraje Puesto que, en un instante dado, sólo una unidad puede transmitir a través del bus, se requiere algún método de arbitraje. Los diversos métodos se pueden clasificar aproximadamente como centralizados o distribuidos.
En un esquema centralizado, un único dispositivo hardware, denominado controlador del bus o árbitro, es responsable de asignar tiempos en el bus. El dispositivo puede estar en un módulo separado o ser parte del procesador.

En un esquema distribuido, no existe un controlador central. En su lugar, cada módulo dispone de lógica para controlar el acceso, y los módulos actúan conjuntamente para compartir el bus.

En ambos métodos de arbitraje, el propósito es designar un dispositivo, el procesador o un módulo de E/S como maestro del bus. El maestro podría entonces iniciar una transferencia de datos (lectura o escritura) con otro dispositivo, que actúa como esclavo en este intercambio concreto.

Temporización El término temporización hace referencia a la forma en la que se coordinan los eventos en el bus.

Con **temporización síncrona**, la presencia de un evento en el bus está determinada por un reloj. El bus incluye una línea de reloj a través de la que se transmite una secuencia en la que se alternan intervalos regulares de igual duración a uno y a cero. Un único intervalo a uno seguido de otro a cero se conoce como ciclo de reloj o ciclo de bus y define un intervalo de tiempo unidad (“time slot”). Todos los dispositivos del bus pueden leer la línea de reloj y todos los eventos empiezan al principio del ciclo de reloj.

Con la **temporización asíncrona**, la presencia de un evento en el bus es consecuencia y depende de que se produzca un evento previo.

La temporización síncrona es más fácil de implementar y comprobar. Sin embargo es menos flexible que la asíncrona, ya que todos los dispositivos deben utilizar la misma frecuencia de reloj. **Con la temporización asíncrona pueden compartir el bus una mezcla de dispositivos lentos y rápidos.**

De que depende el paralelismo de una máquina superescalar? (tomada en final de 27/11/2024)

El paralelismo a nivel de maquina es una medida de la capacidad del procesador para aprovechar el paralelismo en las instrucciones. El paralelismo de la maquina depende del numero de instrucciones que se pueden captar y ejecutar al mismo tiempo y de la velocidad. Tanto el paralelismo en las instrucciones como el paralelismo de la maquina son factores importantes para mejorar las prestaciones.

La coherencia de datos de un sistema jerárquico se ve afectada por el uso de DMA? (tomada en final de 27/11/2024)

Si, el uso de DMA, puede afectar la coherencia de datos en un sistema jerarquico de memoria, especialmente cuando se utiliza una cache.

El problema surge porque el DMA permite a los dispositivos periféricos acceder a la maquina principal directamente sin pasar por la cache del procesador. Esto puede llevar a situaciones donde:

- Un dispositivo DMA modifica un dato en la memoria principal, pero la copia de ese dato en la cache del procesador permanece desactualizada.
- El procesador escribe un dato en la cache, pero el dispositivo DMA lee una versión antigua del dado desde la memoria principal, ya que la cache no se ha actualizado todavía.

Estas inconsistencias pueden resultar en comportamientos inesperados y errores en el programa. Por lo tanto, la coherencia de datos, se PODRIA ver afectada por el uso del DMA.,

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.

Cuales son los elementos de una instrucción de maquina?

Cada instrucción de maquina, contiene la información necesaria para que el procesador la ejecute. Los elementos principales de una instrucción son:

- **Codigo de operación (codop):** Indica la operación que se va a realizar (por ejemplo suma,resta,mover datos,etc.)
- **Referencias a operandos:** Especifican las ubicaciones de los datos (operandos) sobre los que se va a operar.

Describa el diseño de repertorio de las instrucciones.

El diseño de repertorio de instrucciones consta de 5 aspectos principales:

Repertorio de operaciones: Determinar el numero y la complejidad de las operaciones que el procesador puede realizar.

Tipo de datos: Definir los tipos de datos que el procesador puede manejar directamente, como enteros, números en punto flotante, etc.

Formato de las instrucciones: Establecer la longitud de las instrucciones, el numero de direcciones, el tamaño de los campos y la forma en que se codifica el modo de direccionamiento.

Registros: Decidir el numero de registros visibles al usuario.

Direccionamiento: Elegir los modos de direccionamiento que se utilizaran para especificar la ubicación de los operandos en memoria.

Cuales son los elementos de una instrucción de maquina?

- **Código de operación:** especifica la operación a realizar (suma, E/S, etc.). La operación se indica mediante un código binario denominado código de operación o, abreviadamente, codop.
- **Referencia a operandos fuente u origen:** la operación puede implicar a uno o más operan dos origen, es decir operandos que son entradas para la instrucción.
- **Referencia al operando de destino o resultado:** la operación puede producir un resultado.
- **Referencia a la siguiente instrucción:** dice al procesador de dónde captar la siguiente ins trucción tras completarse la ejecución de la instrucción actual

Cual es la diferencia en la invocación entre un procedimiento y una subrutina?

La diferencia clave entre la invocación de un procedimiento y una subrutina radica en cómo se maneja la transferencia de control y el intercambio de datos.

Invocación de un Procedimiento:

Los procedimientos **son invocados explícitamente por el programado**r, lo que significa que el programador decide cuándo y cómo llamar al procedimiento.

En un procedimiento, **se pueden pasar parámetros (datos) al procedimiento y este puede devolver un valor al programa principal.**

La pila se utiliza a menudo para manejar la transferencia de control y el paso de parámetros en las llamadas a procedimientos.

Invocación de una Subrutina:

Generalmente son encargadas de manejar fallos/errores durante la ejecución de un programa.

Las subrutinas pueden ser llamadas tanto por el programador como implícitamente por el sistema operativo o el hardware.

En una subrutina, el intercambio de datos con el programa principal puede ser más limitado o incluso inexistente. El uso de la pila para la gestión de subrutinas depende de la implementación específica.

En resumen, **un procedimiento es una unidad de código modular invocada a propósito por el programador** para realizar una tarea específica. **Puede recibir parámetros y devolver un valor, y la pila se utiliza comúnmente para manejar su ejecución**. **Una subrutina, generalmente es invocada para manejar ciertos fallos/errores durante la ejecución de un programa.**

Que es una pila?

Una pila es un conjunto ordenado de elementos ,en el que solo uno de ellos es accesible en un instante dado. El punto de acceso se denomina cabecera de la pila. El numero de elementos en la pila es **variable.** Solo se pueden añadir o eliminar elementos en la cabecera de la pila, por esta razón, se dice que la pila sigue una estructura LIFO.

- Características de una pila:
    - Acceso restringido: Solo se puede acceder al elemento en la cabecera de la pila.
    - Operaciones básicas: PUSH (Apilar/ agrega un elemento a la pila) POP (Desapilar/elimina el elemento de la cabecera de la pila.)
    - La estructura pila, es de longitud variable.

Que es un procesador RISC? Comparar con los CISC

Mejor explicado anteriormente en el resumen

RISC(Conjunto reducido de instrucciones): Se caracteriza por un conjunto de instrucciones pequeño y simple. Tienen una gran cantidad de registros de propósito general, por lo que reduce la necesidad de acceder a memoria principal. Este tipo de procesadores, están optimizados para la segmentación de instrucciones.

CISC (Conjunto complejo de instrucciones): Buscan optimizar la ejecución de las instrucciones individuales para logar un mayor rendimiento general. Tienen instrucciones de longitud variable y una variedad de modos de direccionamiento. Suelen tener menos registros que los RISC

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.

En que se basa el paralelismo a nivel de maquina?

El paralelismo a nivel de máquina se mide por la cantidad de instrucciones que un procesador puede captar y ejecutar simultáneamente. Cuanto mayor sea el paralelismo de la máquina, mayor será su capacidad para aprovechar el paralelismo a nivel de instrucción y, por lo tanto, mayor será su rendimiento.

Factores que influyen en el paralelismo a nivel de máquina:

●Número de cauces paralelos: Un procesador con más cauces paralelos puede ejecutar más instrucciones al mismo tiempo.

●Velocidad y sofisticación de los mecanismos de detección de instrucciones independientes: El procesador debe ser capaz de identificar rápidamente qué instrucciones pueden ejecutarse en paralelo sin violar las dependencias entre ellas.

●Técnicas de hardware como la duplicación de recursos, la emisión desordenada y el renombramiento de registros:

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 atención. 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 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.

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.

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.

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 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.

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.

Cuales son los elementos de una instrucción?

Los elementos de una instrucción son el **código de operación**, las **referencias a operandos origen y destino**, y la **referencia a la siguiente instrucción**,

Codigo de operación: Especifica la operación a realizar, como una suma,resta,mover datos, etc. Se indica mediante un código binario, denominado código de operación (también conocido como codop)

Referencia a operandos origen: Indican las ubicaciones de los datos de entrada para la operación.

Referencia a operandos destino: Especifica la ubicación donde se almacenará el resultado de la operación.

Referencia a la siguiente instrucción: Le dice al procesador de donde captar la siguiente instrucción tras completarse la ejecución de la instrucción actual. Esta referencia se almacena en el PC (Program counter)

Cuales son los elementos para el <u>diseño</u> de instrucciones?

Los elementos para el diseño de instrucciones son **el repertorio de operaciones, tipos de datos, formatos de instrucciones, registros, modos de direccionamiento.**

Repertorio de operaciones: Esencial para determinar que operaciones serán soportadas por el procesador y cuantas se incluirán. Esto abarca desde operaciones aritméticas y lógicas hasta operaciones mas complejas.

Tipos de datos: El procesador debe ser capaz de manejar distintos tipos de datos. La elección de los tipos de datos a soportar influye en el diseño de la ALU.

Formato de instrucciones: Se defina la longitud de las instrucciones en bits, la cantidad de direcciones que pueden especificar, el tamaño de los campos dentro de la instrucción, y la codificación del modo de direccionamiento.

Registros: Son pequeñas unidades de memoria que almacenan datos e instrucciones temporalmente. Se debe determinar la cantidad de registros visibles al usuario, sus funciones y como se utilizaran en las instrucciones. **Una mayor cantidad de registros puede mejorar el rendimiento al reducir los accesos a memoria, pero aumenta la cmplejidad del harsware.**

Direccionamiento: Define como se especifica la ubicación de un operando en memoria. Existen varios modos de direccionamiento (inmediato,directo,indirecto,de registros,etc.)

.

Finales resueltos (Los saque de otro resumen)

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.

2) **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.

3) 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.

4) 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.)

5) 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.

2) 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.

3) 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.
4) 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.

5) 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.

6) 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.