Finales resueltos:

Final 1:

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

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

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

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

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

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

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

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

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

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

**Tratamiento de multiples interrupciones:**

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

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

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

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

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

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

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

Funcion del DMA:

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

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

Hay dos tipos de transferencias DMA:

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

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

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

Hay 3 tipos de correspondencia de caché:

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

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

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

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

Este enfoque elimina por completo los conflictos de caché.

Politicas de escritura en la caché:

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

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

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

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

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

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

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

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

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

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

Entre los conflictos de dependencia se encuentran:

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

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

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

Algunas soluciones para estos conflictos pueden ser:

Reordenamiento de instrucciones, forwarding (adelantamiento de operandos),

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

1. Caracteristicas que posee un procesador superescalar

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

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

Final 2:

1. Describir el pasaje de argumentos a subrutinas:

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

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

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

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

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

1. Describir el PIC

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

IRR: Especifica las interrupciones que están pendientes de reconocimiento

ISR: Especifica las interrupciones que están siendo atendidas.

IMR: Especifica que interrupciones están enmascaradas y no enmascaradas.

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

1. Conceptos de la memoria cache:

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

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

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

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

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

1. Describir la técnica de renombramientos de registros

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

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

1. Caracteristicas de un bus

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

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

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

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

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

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

Hay dos tipos de buses:

Buses Sincronicos:

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

Buses Asincronicos:

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

1. Que es el MIMD de la Taxonomía de Flynn

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

Los sistemas MIMD pueden clasificarse en dos categorías:

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

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

Final 3:

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

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

Algunas formas de direccionamiento de e/s son:

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

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

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

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