**ARQUITECTURA DE COMPUTADORAS - Preguntas de Finales con respuestas.**

**Final XX/XX/XXXX**

**1. a) ¿Qué métodos para pasaje de argumentos podemos utilizar en una computadora?**

Los métodos para el pasaje de argumentos en una computadora pueden ser: por valor (se pasa el valor de una variable a un procedimiento, este valor no se ve modificado por fuera), y por referencia (se pasa la dirección de memoria del argumento, permitiendo que el valor original de la variable pueda cambiar). También cuando hablamos de pasaje de argumentos a subrutinas pueden ser:

- Vía registros: los parámetros se pasan a través de los registros (limitados) de la CPU.

- Vía memoria: los parámetros se transfieren a través de un área definida de memoria (RAM).

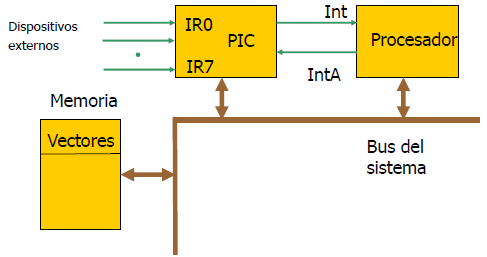
- Vía pila: los datos se pasan a través de la pila, y es independiente de la memoria y registros.

**b) ¿Cuáles son las diferencias en la terminación de una subrutina y un gestor de interrupción?**

La principal diferencia entre la terminación de una subrutina y un gestor de interrupción es que una subrutina se llama y termina su ejecución de manera normal (con RET), mientras que un gestor de interrupción se ejecuta como respuesta a una interrupción externa y luego vuelve a la ejecución de la tarea principal (con IRET). En otras palabras, una subrutina es una función que se llama explícitamente y devuelve un valor a la llamada, mientras que un gestor de interrupciones se ejecuta de manera asincrónica en respuesta a un evento externo y no devuelve un valor a la llamada.

**2. a) Esquematice y describa la estructura interna de un Controlador Programable de Interrupciones.**

El PIC (Controlador de Interrupciones Programable) es un dispositivo interno con líneas de interrupción por hardware, donde se conectan varias dispositivos que pueden interrumpir a la CPU. El PIC permite utilizar varios dispositivos, multiplexando los pedidos de todos ellos a la única línea de interrupción del procesador.



El PIC está conformado por varios registros y por una lógica de control y otra de gestión de prioridad, todo conectado por un bus interno del mismo, la comunicación entre el PIC y el procesador es por medio de señales y comparten el bus del sistema. Los registros internos son:

- ISR (Interrupción en servicio): sus bits indican si se está atendiendo la interrupción de algún dispositivo.

- IRR (Interrupciones pedidas): sus bits indican que dispositivos están solicitando una interrupción.

- IMR (Máscara de interrupciones): sus bits indican qué líneas están habilitadas. Permite el enmascaramiento de interrupciones.

- EOI (Fin de la interrupción): sus bits le indican al PIC que la interrupción ya terminó y fue atendida.

- INT0 a INT7: 8 registros, donde se cargan los ids de las interrupciones correspondientes.

**b) Describa cómo funciona la gestión de E/S programada con espera de respuesta.**

En la gestión de la transferencia de E/S programa con espera de respuesta la CPU interviene directamente en la transferencia de cada unidad de información con el módulo, o sea que tiene el control directo sobre la operación de E/S. Es capaz de comprobar el estado del dispositivo, enviarle comandos de escritura o lectura y realizar la transferencia de todos los datos. La CPU le ordena tanto al módulo como al dispositivo que deben ser ejecutados y realizar alguna tarea a través de una orden. El procesador es el responsable de comprobar periódicamente el estado del módulo de E/S hasta que la operación haya terminado (permanece ociosa).

**3. a) ¿Cuáles son los elementos a tener en cuenta para el diseño de una memoria Cache?**

* Organización. - Tamaño de caché: la caché debe ser suficientemente grande para contener la mayor cantidad posible de información. Limitado por las superficies disponibles de chip y de tarjeta que contiene el procesador.

- Tamaño de línea: los bloques más grandes reducen el número de bloques que caben en la caché y hace que cada palabra adicional esté mas lejos de la requerida.

- Número de cachés: la memoria caché puede ser una sola o estar dividida en varias unidades (cache on-chip como externa, de dos niveles). Además una caché puede estar unificada o partida (una parte de la caché dedicada a instrucciones y otra a datos).

* Función de correspondencia: determina cómo se organiza la cache, es decir la forma en la que se van a asignar los bloques de la memoria principal en la memoria caché. Las tres técnicas son: correspondencia directa, correspondencia asociativa y correspondencia asociativa por conjuntos.
* Políticas de reemplazo: cuando se ubica un bloque traído desde memoria se requiere reemplazar un bloque existente. Para las técnicas asociativas se precisan algoritmos de sustitución y son 4: (LRU: utilizado menos recientemente, FIFO: primero en entrar-primero en salir, LFU: utilizado menos frecuentemente y aleatoria).
* Políticas de escritura: desde el punto de vista de la coherencia de datos, se debe evitar inconsistencia de información entre las memorias principal y cache, durante los procesos de escrituras. Las políticas de escritura en aciertos son: escritura inmediata y post-escritura, y en fallos: no-write allocate y write allocate.
* Velocidad de acceso y coherencia de la caché.

**b) ¿La coherencia de datos de un sistema jerárquico de memoria se ve afectado por el uso de DMA?**

Sí, el uso de DMA (Acceso Directo a Memoria) puede afectar la coherencia de datos en un sistema jerárquico de memoria. Esto se debe a que el DMA permite que dispositivos periféricos accedan directamente a la memoria principal sin pasar por la jerarquía de caché, lo que puede dar lugar a inconsistencias de datos en la caché y la memoria principal. Para mantener la coherencia de datos, se requieren técnicas especiales como invalidación de caché y protocolos de coherencia de caché para garantizar que los datos se sincronicen correctamente.

**4. a) ¿De qué depende el paralelismo de una máquina superescalar?**

El paralelismo es la capacidad para ejecutar varias tareas en el mismo intervalo de tiempo. El paralelismo de una máquina superescalar depende tanto a nivel de instrucciones como a nivel de la máquina. A nivel de instrucciones depende de la frecuencia de dependencias de datos verdaderas y dependencias relativas al procedimiento que haya en el código, a su vez dichos factores dependen de la arquitectura del repetorio de instrucciones y de la aplicación. Y de la máquina: depende del número de instrucciones que pueden captarse y ejecutarse al mismo tiempo, y de la velocidad del mecanismo que usa el procesador para localizar instrucciones independientes.

**b) ¿Cuál es el objetivo de usar la técnica de Renombre de Registros en un procesador superescalar?**

El objetivo principal de la técnica de Renombre de Registros es permitir que múltiples instrucciones que están listas para ejecutarse puedan hacerlo simultáneamente, incluso si dependen de los resultados de instrucciones anteriores, aumentando la capacidad de paralelismo de instrucciones y mejorando el rendimiento del procesador. Esta técnica consiste en disponer de registros adicionales (internos, ocultos al programador) y asignarlos (por hardware) a instrucciones en conflicto.

**5. a) ¿Qué elementos característicos definen un bus?**

Respondidas en final 10/2022

**b) ¿Qué son los MIMD de la tanoxomía de Flynn?**

**Final 10/2022**

**1. a) ¿A que método de atención lo conocemos como de “interrupciones vectorizadas”?**

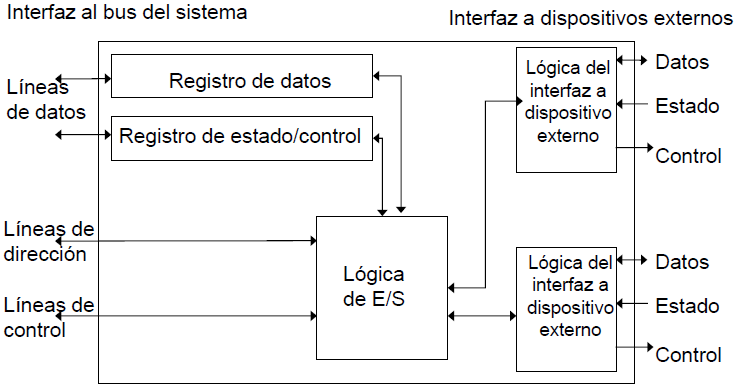
Las interrupciones vectorizadas son un método de atención utilizado en sistemas para mejorar la eficiencia en el manejo de interrupciones. Con las interrupciones vectorizadas, cuando un módulo emite una interrupción, este envía un vector que funciona como identificador al bus, para que el procesador o el controlador de interrupciones sepa cómo gestionarlo sin necesidad de preguntarle a cada módulo, es decir, que con este vector sabría cómo atender adecuadamente para tal interrupción buscando la rutina correspondiente en el vector de interrupciones e identificaría el periférico que lo emitió. Este vector es colacado directamente por el periférico o por el PIC, que se ocupa de todo.

**b) ¿Cuándo, por qué, para qué y cómo se utiliza una de las denominadas interrupciones por software?**

Las interrupciones por software son utilizadas para simular interrupciones de hardware en situaciones en las que no hay un dispositivo de E/S físico presente. Se pueden utilizar también para permitir que los programas realicen tareas en segundo plano mientras se espera que ocurra una interrupción por hardware, o para hacer llamadas a funciones del sistema operativo, proporcionando rapidez y eficiencia en el flujo de ejecución del sistema. Las interrupciones por software se generan mediante la ejecución de una instrucción específica en el programa, que interrumpe la secuencia normal de ejecución y pasa el control al manejador de interrupciones correspondiente.

**2. a) Esquematice y describa la estructura interna de un módulo de E/S.**

El módulo se conecta al resto del computador a través de un conjunto de líneas. Los datos que se transfieren a y desde el módulo se almacenan temporalmente en uno o más registros de datos. Además, puede haber uno o más registros de estado que proporcionan información del estado presente. Un registro de estado también puede funcionar como un registro de control, para recibir información de control del procesador. La lógica que hay en el módulo interactúa con el procesador a través de una serie de líneas de control. Estas son las que utiliza el procesador para proporcionar las órdenes al módulo de E/S. El módulo también debe ser capaz de reconocer y generar las direcciones asociadas a los dispositivos que controla, teniendo el módulo una dirección única o si controla a más de un dispositivo externo, un conjunto único de direcciones. Por último, el módulo de E/S posee la lógica específica para la interfaz con cada uno de los dispositivos que controla.



**b) Describa los posibles modos de ubicación de los módulos de E/S.**

* Espacio de E/S compartido con memoria (memory-mapped).
  + En esta técnica los registros de los dispositivos de E/S y memoria comparten un único espacio de direcciones.
  + Los registros de E/S se comportan idéntico a una memoria de lectura/escritura.
  + No hay instrucciones específicas para E/S, se usan las mismas instrucciones de movimiento de datos a memoria. Permite una variedad de órdenes de acceso a memoria (programación eficiente).
* Espacio de E/S separada de la memoria (aislada).
  + En esta técnica los registros de los dispositivos de E/S y la memoria están en diferentes espacios de direcciones.
  + Dado que el bus de direcciones compartido por la memoria y el subsistema de E/S, se requieren señales de control adicionales para identificar a donde está accediendo la CPU.
  + Hay un conjunto limitado de instrucciones específicas de E/S, distintas de las instrucciones de acceso a la memoria. Cuando se ejecutan estas instrucciones específicas, en el bus de control se identifica el acceso al mapa de direcciones de E/S, para el resto de instrucciones se identifica el acceso a memoria.

**3. a) Describa las funciones de correspondencia entre Memoria Principal y Cache.**

La elección de la función de correspondencia determina cómo se organiza la cache, es decir la forma en la que se van a asignar los bloques de la memoria principal en la memoria caché. Hay tres ténicas principales de correspondencia:

- Correspondencia directa: un bloque de memoria se asigna a una sola linea de cache, por lo tanto, cuando este bloque se actualice en memoria, la misma linea de cache asignada también será actualizada, lo que facilita la recuperación de datos, pero puede provocar una alta tasa de conflictos.

- Correspondencia asociativa: un bloque de memoria puede ser asignado o mapeado en cualquier linea de cache, por lo que para este caso se trata de manera distinta la metodología de localización del bloque almacenado, los bloques ahora cuentan con una etiqueta que sirve para poder ser ubicados en cache, lo que hace que sea más eficiente, pero podría conllevar costos por hardware.

- Correspondencia asociativa por conjuntos: , la cache se divide en bloques, por lo que el mapeo en memoria pasa a ser de a conjuntos de líneas (bloques) donde se le da el mismo tratamiento de etiquetado para identificar el bloque en cache. Esta combina lo mejor de las otras correspondencias.

**b) Analice las políticas de escritura desde el punto de vista de la coherencia de datos.**

Desde el punto de vista de la coherencia de datos, se debe evitar inconsistencias de información entre la memoria principal y cache, durante los procesos de escrituras. Es decir, que aún escribiéndose el dato en la caché, el correspondiente bloque de memoria principal debe ser actualizado en algún momento. Las políticas comunes son:

- En aciertos, escritura inmediata y post escritura. La escritura inmediata se basa en que todas las operaciones de escritura se hacen, tanto en la caché como en memoria pirncipal, mientras que en la post escritura, la información sólo se actualiza en la caché y se escribe la memoria principal cuando se reemplaza el bloque.

- En fallos, write allocate y no-write allocate. En la primera, el bloque requerido primero se copia en la caché, y luego se escribe en la caché. Y en la segunda, el bloque no se lleva a la memoria caché y se escribe directamente en la memoria principal.

**4. a) ¿Qué características tienen los procesadores superescalares?**

Los procesadores superescalares son aquellos procesadores donde las instrucciones comunes pueden iniciar su ejecución simultáneamente y ejecutarse de manera independiente. Algunas características de estos son: el procesador por medio de estrategias y mecanismos es capaz de ejecutar dos o más instrucciones en cada etapa, en cada ciclo se inician más de una instrucción, por ende, es necesario duplicar parte de la CPU y ALU, además cuentan con múltiples unidades funcionales segmentadas y jerarquías de memoria capaces de atender múltiples referencias de memoria. Por último, poseen una lógica para determinar depedendencias verdaderas entre valores de registros y tienen estrategias que permiten la captación de múltiples instrucciones al mismo tiempo.

**b) Describa las causas que pueden retardar el funcionamiento de los mismos.**

Existen limitaciones fundamentales del paralelismo: dependencia de datos verdadera, dependencia relativa al procedimiento y de control, conflictos en los recursos, dependencia de salida y la antidependencia. También el ancho de banda de memoria, como los saltos y llamadas a subrutinas pueden llegar a retardar.

**5. a) ¿Qué elementos característicos definen un bus?**

Hay algunos elementos de diseño que sirven para definir y caracterizar a los buses:

- Tipo: las líneas de bus se pueden dividir en dos tipos genéricos: dedicadas y multiplexadas.

- Método de arbitraje: se refiere al proceso utilizado para gestionar y controlar el acceso de múltiples dispositivos al bus (centralizado o distribuido).

- Temporización: hace referencia a la coordinación de las operaciones de transferencia de datos en el bus en relación con el tiempo (síncrono o asíncrono) y la velocidad del reloj.

- Anchura del bus: número de líneas o cables en el bus que se utilizan para transmitir datos y referencias simultáneamente.

- Tipo de transferencia de datos: se refiere al movimiento de información desde una ubicación de almacenamiento hacia otro componente del sistema (lectura o escritura) en bloques y a su latencia.

**b) ¿Qué son los MIMD de la taxonomía de Flynn?**

Los MIMD (múltiples secuencias de instrucciones y múltiples secuencias de datos) son una de las categorías de computadoras nombradas en la taxonomía de Flynn que hace referencia al conjunto de procesadores que ejecutan simultáneamente secuencias de instrucciones diferentes con conjuntos de datos diferentes. Esto significa que son capaces de procesar múltiples instrucciones al mismo tiempo y utilizar múltiples datos en paralelo. Hay múltiples unidades de control, y cada una proporciona una secuencia de instrucciones separada a su propio elemento de proceso.

El MIMD puede ser un multiprocesador de memoria compartida (SMP o sistemas NUMA), o un multicomputador de memoria distribuida (clusters).

**Final 04/2023**

**1. E/S. a- Describa las características funcionales del acceso directo a memoria (DMA).**

El acceso directo a memoria es una técnica de transferencia de datos entre el periférico y memoria sin intervención directa del CPU. Es llevada a cabo por un módulo de DMA (DMAC), encargado de llevar a cabo la transferencia. El DMAC es capaz de imitar al procesador y puede recibir el control del sistema. Este módulo puede tomar el control del bus e indicar las dirreciones implicadas para el almacenamiento de los datos a transferir, como también el tipo de operación que se va a realizar (IN/OUT). El procesador solo se encarga de preparar y verificar el resultado de la transferencia.

El DMA puede transferir bloques de datos completos de manera eficiente (útil para transferencias de datos de gran tamaño) y de forma asíncrona. Los controladores DMA pueden operar en dos modos: por ráfaga (el DMAC no libera el bus solicitado a la CPU hasta no haber finalizado la transferencia de todo el bloque de datos completo) o por robo de ciclo (donde el DMAC toma el control del bus, transfiere el dato y luego devuelve el control del bus, repetiendo este ciclo tantas veces como sea necesario). Por último, el DMA puede acceder a diferentes regiones de memoria del sistema utilizando direcciones de memoria física.

**2. Segmentación del cauce. a- Definición.**

La segmentación de cauce es una forma efectiva de organizar el hardware de la CPU para realizar más de una operación al mis tiempo. Consiste en descomponer el procesedo de ejecución de las instrucciones en fases o etapas que permitan una ejecución simultánea (explotando el paralelismo entre las instrucciones de un flujo secuencial).

**b- Rendimiento.**

- Aprovechamiento del paralelismo: la segmentación del cauce divide el proceso de ejecución de instrucciones en varias etapas, permitiendo que múltiples instrucciones se ejecuten simultáneamente en diferentes etapas del cauce, aprovechando así el paralelismo a nivel de instrucción.

- Reducción del tiempo de ejecución y mejora de las prestaciones: no se mejora la velocidad de la instrucción, sino que la principal ventaja es el incremento de la productividad (throughput), es decir la cantidad de instrucciones resueltas en un período de tiempo determinado.

- Mejora del rendimiento en sistemas superescalares: la segmentación del cauce es beneficiosa para los procesadores superescalares, donde se pueden aprovechar aún más las capacidades del paralelismo del procesador, lo que resuelta en un rendimiento mejorado.

**c- Describa el problema y posibles soluciones ante riesgos por transferencia de control del programa.**

Los atascos son situaciones que impiden que las instrucciones se ejecuten en el ciclo que le corresponden. En el caso de las instrucciones de salto, estos generan conflictos por dependecia de control del programa. Estos saltos pueden ser incondicionales o condicionales, y el problema de estos es que se necesita esperar a que se complete la ejecución de una instrucción anterior en otra etapa del cauce antes de poder continuar su propia ejecución.

La forma más elemental de resolver el conflicto es agregando tiempos muertos hasta que desaparece el programa, pero esto resulta muy ineficiente.

Las técnicas para mitigar el impacto de las instrucciones pueden ser de dos tipos:

- Por hardware: consiste en resolver los conflictos a nivel de hardware mediante predicción de saltos. Encontramos técnicas estáticas y dinámicas. Sobre las técnicas estáticas no tienen en cuenta información previa de la ejecución del programa y se presume una condición (salta o no salta) y se ejecuta en base a esa predicción (nunca se salta o siempre se salta).

Por otra parte las técnicas dinámicas si tienen en cuenta la historia previa del programa en ejecución y son:

* Conmutador Saltar/no saltar: se conmuta de un estado al otro cuando ocurrieron dos predicciones fallidas.
* Tabla de historia de saltos (BHT): se usa una pequeña caché y cada vez que ocurre un salto se busca en esta si la instrucción es de ramificación. Si se tiene que saltar, se capta la dirección de salto y si no, se actualiza la tabla. En caso de no estar la instrucción, se la carga como nueva entrada.
* Predicción según el código de operación: se basa en la suposición que algunas instrucciones de salto tienen mayor probabilidad de saltar o no saltar.
* Varios cauces: tener cauces distintos para ejecutar simultáneamente los cauce, uno que sea para la opción de saltar y otro para no saltar.
* Precaptación del destino de salto: se precapta la instrucción destino del sato, además de las siguientes instrucciones a la bifurcación.
* Buffer de bucles: se usa una memoria muy rápida, que contiene las últimas instrucciones recientemente buscadas.

- Por software: intentan resolver los conflictos por software a nivel del compilador. Se basan en tratar de realizar trabajo útil mientras el salto se resuelve. Algunas máquinas cpatan y ejecutan siempre la instrucción siguiente a una instrucción de ramificación, la idea es que el compilador pueda tratar de insertar (reordenando las instrucciones) en esos huecos instrucciones útiles que no dependan del salto.

**3. Memoria caché. a- Si se pretende mejorar el tiempo de acceso medio a memoria cache. ¿Sobre qué parámetros será necesario trabajar y propone como medida para hacerlo?**

El tiempo de acceso medio a memoria es igual a la suma de la tasa de aciertos y la tasa de fallos en memoria caché. Esta tasa de fallos es el resultado de la multiplicación entre la tasa de fallos de la caché y el tiempo de acceso (penalización) a la memoria principal cuando ocurre esta situación. Para mejorar las prestaciones es necesario realizar tres acciones sobre tres parámetros mencionados: reducir el tiempo en caso de acierto (tasa de aciertos), reducir la tasa de fallos de la caché y reducir la penalización por fallo.

**b- Compare las correspondencias entra la MP y la caché. ¿Cómo se determina, en cada caso, la hilera/fila de ubicación de un bloque de palabras provenientes de la MP? Ejemplifique.**

- Correspondencia directa: consiste en hacer corresponder cada bloque de memoria principal a sólo una línea posible de la caché.

* Se implementa utilizando la dirección. Cada dirección de memoria principal puede verse como dividida en tres campos:
  + Tag: es el número de grupo en la memoria principal.
  + Line: es el número de bloque dentro del grupo.
  + Word: es el número de palabra dentro del bloque.
* El uso de una parte de la dirección como número de línea proporciona una correspondencia única o asignación única de cada bloque de memoria principal en la cache.
* Si la comparación da un acierto, el dato se busca en la caché. Si la comparación da una falla, el dato se trae de la memoria principal.
* Esta técnica es simple y poco costo de implementar.
* Su principal desventaja es que hay una posición concreta de caché para cada bloque dado. Si un programa accede a palabras de dos bloques diferentes en la misma línea de caché, se intercambiarán continuamente, resultando en baja tasa de aciertos y grandes pérdidas de caché.
* Numero de línea de cache = Numero de bloque referenciado MOD número de líneas de cache.

- Correspondencia asociativa: permite que cada bloque de memoria principal puede cargarse en cualquier línea de caché.

* La lógica de control interpreta una dirección de memoria como una etiqueta (tag) y un campo de palabra (word).
* El campo de etiqueta identifica unívocamente un bloque de memoria principal.
* Para determinar si un bloque está en la caché, su lógica de control (memoria del tipo asociativa) debe examinar simultáneamente todas las etiquetas de líneas para buscar una coincidencia.
* Si la comparación da un acierto, el dato se busca en la caché. Si la comparación da una falla, el dato se trae de la memoria principal.
* Hay flexibilidad para que cualquier bloque sea reemplazado cuando se va a escribir uno nuevo en la cache.
* La principal desventaja de este tipo de correspondencia es la compleja circuitería necesaria para examinar en paralelo las etiquetas de todas las líneas de caché.
* Un bloque puede almacenarse en cualquier lugar de la caché.

- Correspondencia asociativa por conjuntos: un bloque puede almacenarse en un conjunto restringido de lugares en la caché.

* Un bloque de memoria principal puede colocarse en bloques determinados de la cache. La caché se divide en un número de conjuntos N.
* Cada conjunto N contiene un número de líneas o ranuras.
* Un bloque determinado corresponderá a alguna línea o ranura de un conjunto determinado.
* La etiqueta de una dirección de memoria es mucho más corta, y se compara sólo con las etiquetas dentro de un mismo conjunto.
* Combina lo mejor de las otras correspondencias (asociativa y directa).
* Numero de línea de cache = Numero de bloque referenciado MOD número de conjuntos/vías.

**4. Procesadores superescalares. a- ¿Cuáles son las características distintivas?**

Respondida en final 10/2022

**b- Compare las políticas de emisión de instrucciones.**

Las políticas de emisión de instrucciones en procesadores superescalares son protocolos usados para el envío de las instrucciones a las unidades funcionales. Existen tres políticas:

- Emisión en orden y finalización en orden: consiste en emitir instrucciones en el orden exacto en lo que lo haría una ejecución secuencial (emisión en orden) y escribir los resultados en ese mismo orden (finalización en orden). Es más simple de implementar en hardware, pero puede resultar en una menor utilización de las unidades de ejecución y un menor grado de paralelismo de instrucciones, ya que las instrucciones deben ejecutarse secuencialmente.

- Emisión en orden y finalización desordenada: las instrucciones se emiten exactamente como está en el programa, pero los resultados no necesariamente se completan en el mismo orden (desordenada). Las instrucciones entran a las unidades de ejecución a medida que se van decodificando y haya unidades de ejecución disponibles. El gran problema es que la finalización desordenada requiere de una lógica más compleja que la política anterior, ya que debe evitar la nueva dependencia (de salida, WAR) y mantener la lógica del programa.

- Emisión desordenada y finalización desordenada: tanto la emisión como la finalización de las instrucciones pueden ocurrir en cualquier orden. Se utiliza una ventana de instrucciones donde se coloca la instrucción del programa, para que después sea ejecutada sin orden. Surge la antidependencia (WAW), a pesar de que el procesador tenga capacidad de anticipación.

**c- Elija una alternativa de emisión/finalización y justifique.**

Una alternativa de emisión/finalización puedede ser el renombramiento de registros, que es la técnica utilizada en los procesadores superescalares para evitar las dependecias de datos. Las técnicas de software de optimización de registros empeoran los conflictos en los registros usados en las técnicas de desordenación. Los efectos de este tipo de conflictos de almacenamiento se pueden reducir con la técnica mencionada al principio. Consiste en que el hardware del procesador asigna dinámicamente los registros, que están asociados con los valores que necesitan las instrucciones en diversos instantes de tiempo. Cuando se crea un nuevo valor de registro, se asigna un nuevo registro para dicho valor. Las instrucciones siguientes que referencien a ese registro, deben renombarse para referenciar el registro físico que tiene el dato correcto. Por ende, las referencias a un mismo registro original en diferentes instrucciones, pueden referirse a distintos registros reales, con distintos valores.

**5. Procesamiento paralelo. a- Describa las 4 variantes de arquitectura de la Taxonomía de Flynn.**

- Una secuencia de instrucciones y una secuencia de datos (SISD): un único procesador interpreta una única secuencia de instrucciones, para operar con los datos almacenados en una única memoria. Se dispone de una unidad de control que proporciona una secuencia de instrucciones a una unidad de proceso. La unidad de proceso actúa sobre una única secuencia de datos captados desde la unidad de memoria. Ejemplo: monoprocesadores.

- Una secuencia de instrucciones y múltiples secuencias de datos (SIMD): una única instrucción máquina controla paso a paso, la ejecución simultánea y sincronizada de un cierto números de elementos de proceso. Existe una sola unidad de control que proporciona una única secuencia de instrucciones a cada elemento del proceso. Cada elemento de proceso puede tener su propia memoria dedicada, o puede haber una memoria compartida. Ejemplos: procesadores vectoriales y matriciales.

- Múltiples secuencias de instrucciones y una secuencia de datos (MISD): se transmite una secuencia de datos a un conjunto de procesadores, cada uno de los cuales ejecuta una secuencia de instrucciones diferente.

- Múltiples secuencias de instrucciones y múltiples secuencias de datos (MIMD): un conjunto de procesadores ejecuta simultáneamente secuencias de instrucciones diferentes con conjuntos de datos diferentes. Hay múltiples unidades de control, y cada una proporciona una secuencia de instrucciones separada a su propio elemento del proceso. Puede ser MIMD de memoria compartida (SMP o NUMA), o de memoria distribuida (clusters).

**b- Compare los sistemas MP y clusters.**

SMP:

* Son más faciles de gestionar y configurar que un cluster.
* Son más cercanos al modelo de computador de un solo procesador, disponible en casi todas las aplicaciones.
* El principal cambio que se necesita para pasar de un computador monoprocesador a un SMP se refiere al funcionamiento del planificador.
* Necesitan menos espacio físico y consumen menos energía que un cluster comparable.
* Son plataformas estables y bien establecidas.

Clusters:

* Son superiores a los SMP en términos de escalabilidad absoluta e incremental.
* Son superiores también en términos de disponibilidad, ya que todos los componentes del sistema puede hacer altamente redundantes.
* Los nodos cuentan cada uno con memoria privada, y las aplicaciones no ven la memoria global.

**Final 06/2023**

**1. a) ¿Cómo es el funcionamiento del método de atención de interrupciones que conocemos como “interrupciones vectorizadas”?**

1

**b) ¿Cuándo y cómo se utiliza alguna de las denominadas interrupciones por Software?**

**2. a) Esquematice y describa la estructura interna de un módulo de E/S.**

**b) Describa cómo funciona la gestión de E/S programada con espera de respuesta.**

**3. a) ¿Cuáles son los elementos a tener en cuenta para el diseño de una memoria cache?**

**b) ¿La coherencia de datos de un sistema jerárquico de memoria se ve afectado por el uso de DMA?**

2

**4. a) ¿De qué depende el paralelismo de una maquina superescalar?**

**b) ¿Cuál es el objetivo de usar renombre de registros de un procesador superescalar?**

**5. a) ¿Qué elementos característicos definen un bus?**

1

**b) ¿Qué son los MIMD de la Taxonomía de Flynn?**

(1) Respondidas en final 10/2022

(2) Respondidas en final XX/XX/XXXX

**Final 05/07/2023**

**1. a) Explique el mecanismo de interrupción.**

El mecanismo de interrupción funciona de la siguiente manera:

1. Detectar el pedido de interrupción: el procesador examina, en cada ciclo de instrucción la presencia de interrupciones. Para poder implementar la tarea de detección de interrupciones (además de las fases de captura y ejecución), en el ciclo de instrucción se agrega la fase de gestión de interrupciones.

La etapa de gestión de interrupciones determina la presencia o ausencia de pedido de las mismas, la presencia de un pedido de interrupción se manifiesta mediante una o varias señales discretas conocidas como banderas que la CPU examina, que se encuentran en algún registro del entorno.

1. Detener la tarea que se estaba ejecutando: en caso de no haber pedido pendiente (Flag inactivo) se inicia el ciclo de captación de la siguiente instrucción y no se interrumpe la ejecución del programa. En caso contrario, el procesador responde suspendiendo la operación del programa que estaba ejecutando.
2. Salvar el estado de la tarea que se estaba ejecutando: si hay algún pedido de interrupción pendiente, el procesador guarda en la pila del sistema, el “estado del proceso”. Puede o guardar la dirección de la próxima instrucción y algún registro importante, o guardar directamente todos los registros del procesador.

El objetivo de esta operación es el de restablecer el estado del procesador al terminar el servicio de interrupción.

1. Obtener la dirección de comienzo del servicio de la interrupción y bifurcar a dicho servicio: se obtiene la dirección donde comienza la rutina de la interrupción y carga el PC con este valor, bifurcando al servicio de la interrupción.

Se dispone de un área de memoria reservada, conocida como área de vectores de interrupciones, donde están todas estas direcciones.

1. Ejecutar el servicio de interrupción: el procesador comienza a ejecutar la rutina de manejo de interrupciones correspondiente. Una vez que la rutina se ha ejecutado, el control es devuelto al punto en el programa donde se detuvo inicalmente.
2. Retornar y restaurar el estado en que estaba la tarea interrumpida: una vez que se manejó la interrupción, el procesador restaura el estado previo guardado, incluyendo los registros y la posición del programa, permitiendo que el programa original se reanude como si la interrupción nunca hubiera ocurrido.
3. Continuar con la ejecución normal de la tarea interrumpida: cuando la rutina de gestión de interrupción se completa, el procesador puede proseguir la ejecución del programa en el punto en el que se interrumpió.

**b) Describa el tratamiento a realizar cuando hay múltiples fuentes de interrupciones.**

Los procesadores son capaces de manejar varias interrupciones de distintas características y orígenes. Dado el origen diverso de las mismas, hay algunas que son más importantes que otras. Las interrupciones más importantes deben tener mayor “prioridad” que las menos importantes. Cuanto mayor sea su prioridad, mayor es la urgencia para ser atendida (pudiendo interrumpir a las de menor prioridad), incluso si hay una interrupción en curso.

- Procesamiento de interrupciones de igual prioridad: las interrupciones se atienden según el orden de llegada. Cuando llega una interrupción y es atendida, se inhabilita el resto de las interrupciones de igual o menor nivel de prioridad (si llega una nueva, quedará pendiente). El procesador ejecutará el servicio de la interrupción atendida y una vez finalizado el servicio, se habilitarán nuevamente las interrupciones.

- Procesamiento de interrupciones de distinta prioridad: una interrupción de prioridad más alta puede interrumpir en cualquier momento a una que es menor prioritaria. Cuando se ha gestionado la interrupción de prioridad más alta, el procesador vuelve a las interrupciones previas. Terminadas todas las rutinas de gestión de interrupciones se retoma el programa del usuario.

Respondida en final 04/2023

**2. a) Describa las funciones de correspondencia entre memoria principal y caché.**

Respondida en final 10/2022

**b) Analice las políticas de escritura desde el punto de vista de la coherencia de datos.**

Respondida en final 04/2023

**3. a) ¿Qué entiende por segmentación de cauce?**

**b) ¿Qué ventajas proporciona su implementación?**

La implementación de la segmentación de cauce proporciona varias ventajas, entre ellas:

- Mejora el rendimiento: la ejecución en paralelo acelera la velocidad de ejecución de las instrucciones.

- Aprovechamiento de recursos: mientras una unidad de ejecución realiza una operación, las etapas anteriores y posteriores pueden estar ocupadas con otras instrucciones, lo que permite un uso más completo de las unidades funcionales.

- Mayor paralelismo: al permitir que múltiples instrucciones se ejecuten al mismo tiempo, la segmentación de cauce aumenta el nivel de paralelismo a nivel de instrucciones en un procesador.

- Reducción del ciclo de reloj por instrucción: reduce el tiempo necesario para ejecutar una instrucción completa al dividirla en etapas más pequeñas.

- Mejora de la predicción de saltos: puede combinarse con técnicas de predicción de saltos para reducir las penalizaciones por fallos de predicción de saltos, lo que resulta en una ejecución más eficiente en las instrucciones.

**4. a) Describa tres diferentes causas que pueden retardar un cauce de instrucciones segmentado.**

**b) ¿Qué retardo produce cada una?**

- Estructurales: son conflictos provocados por el uso de los recursos del sistema. Los recursos típicamente son: memoria, ALU, registros. En el caso de que la memoria fuera una sola habría un conflicto cuando dos etapas quieren acceder a memoria. La solución es dividir la memoria para reducir los conflictos por accesos a memoria.

- Por dependencia de datos: son conflictos originados entre dos o más instrucciones que comparten un mismo dato dentro del cauce. Una etapa podría depender del contenido de un registro que podría verse alterado por una instrucción previa que aún esté en el cauce.

Los tres tipos de dependencias de datos son:

* Lectura después de Escritura (RAW, dependencia verdadera): una instrucción escribe un dato que otra lee posteriormente.
* Escritura después de Escritura (WAW, dependencia en salida): una instrucción escribe un dato que otra escribe posteriormente, sólo ocurre si se permite que las instrucciones se adelanten unas a otras.
* Escritura después de Lectura (WAR, anti dependencia): una instrucción lee un dato que otra escribe posteriormente.

- Por dependencia de control: son conflictos que ocurren cuando la ejecución de una instrucción depende de cómo se ejecute otra. Son riesgos que pueden ocurrir cuando se va a ejecutar una instrucción de salto condicional. Una instrucción tiene que calcular el nuevo valor que modifica el valor del PC. La próxima instrucción no puede comenzar hasta que no se resuelva el dato.

Respondida en final 10/2022

**5. a) ¿Qué elementos característicos describen un bus?**

Respondida en final XX/XX/XXXX

**b) ¿De qué depende el paralelismo de una máquina superescalar?**

**Final 23/9/2023 y 23/10/2023**

Respondida en final 05/07/2023

**1. Interrupciones. a- Explique el mecanismo de interrupción.**

**b- Describa cómo se realiza el reconocimiento de interrupciones vectorizadas mediante el PIC.**

Cuando un dispositivo quiere interrumpir a través de la linea N, el PIC realiza las siguientes acciones:

1. Verifica que la línea esté habilitada (que no esté enmascarada).
2. Verifica que no haya otros dispositivos con mayor prioridad que también quieran interrumpir.
3. Si las dos verificaciones previas fueron exitosas, le solicita atención a la CPU con la única señal de pedido de interrupción IntR (Interrupt request) y cuando este esté lista para atendar la interrupción, le avisa al PIC mediante la señal IntA (Interrupt aknowledge).
4. Al mismo tiempo el PIC genera en el bus de datos el número de la interrupción (vector) correspondiente a la línea N a ser atendida.
5. Cuando el procesador recibe la señal de interrupción del PIC, suspende temporalmente la ejecución del programa y procede a atender la interrupción.
6. El procesador utiliza el número de vector de interrupción recibido del PIC para acceder a una tabla de vectores de interrupción específica.
7. Una vez se ha localizado el manejador de interrupciones adecuado, el procesador ejecuta el código contenido en esa dirección de memoria.

Respondida en final 10/2022

**2. E/S. a- ¿Cómo es la estructura de un módulo de E/S?.**

Respondida en final 04/2023

**b- Describa las características funcionales del acceso directo a memoria - DMA.**

**3. Memoria. a- ¿Por qué funciona un sistema de memoria basado en jerarquía?**

La jerarquía de memoria funciona eficazmente debido a la explotación del principio de localidad de las referencias y a la combinación de diferentes tipos de memoria con características complementarias en términos de velocidad, capacidad y costo.

El principio de localidad de las referencias establece que el procesador en períodos de tiempos cortos, trabaja principalmente con agrupaciones (clusters) fijas de referencias a memoria. De esta manera es posible organizar los datos a través de la jerarquía, colocando los datos más frecuentemente utilizados en los niveles superior de la jerarquía, de tal manera que el porcentaje de accesos a cada nivel siguiente más bajo, sea sustancialmente menor que al nivel anterior. Este principio puede aplicarse a través de más de dos niveles de memoria, como sugiere la jerarquía.

Una vez explicado este principio, podemos dividirlo en dos subprincipios:

- Localidad temporal de las referencias: se refiere a la tendencia de un programa a acceder repetidamente a los mismos datos en un corto período de tiempo. En otras palabras, si un dato ha sido accedido recientemente, es probable que se vuelva a acceder en el futuro cercano.

- Localidad espacial de las referencias: se refiere a que es altamente probable que los próximos elementos de memoria referenciados estén en las proximidades de los últimos referenciados.

Además, para que la memoria se comporte como una jerarquía debe cumplir dos propiedades:

- Inclusión: los datos almacenados en un nivel han de estar también almacenados en los niveles inferiores a él.

- Coherencia: las copias de la misma información en los distintos niveles deben contener los mismos valores.

Respondida en final 05/07/2023

**b- Analice brevemente todos los elementos a tener en cuenta para el diseño de una memoria Cache.**

Respondida en final 04/2023

**4. Instrucciones. a- ¿Qué es la segmentación del cauce de instrucciones?**

Respondida en final 04/2023

**b- ¿Cuánto mejora el rendimiento?**

Respondida en final 06/2023

**c- ¿Qué es un riesgo WAR en un cauce segmentado?**

Respondida en final XX/XX/XXXX

**5. Paralelismo. a- ¿Qué es y de qué depende el paralelismo de una máquina?**

Respondida en final 10/2022

**b- ¿Qué características tiene la implementación de un procesador superescalar?**