Preguntas finales

Final Arqui 10/22

1. **¿A qué método de atención lo conocemos como “interrupciones Vectorizadas”?**

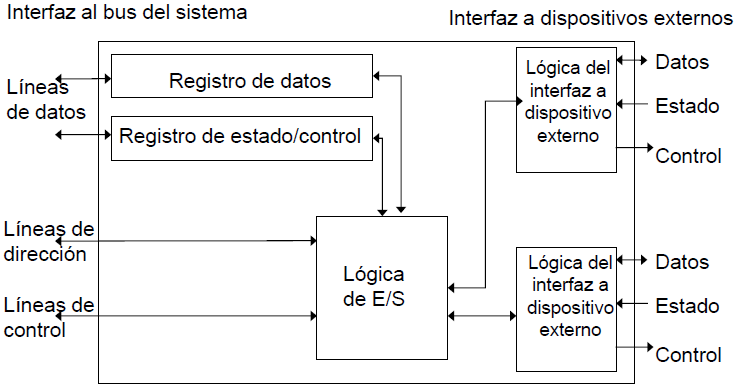
Las interrupciones vectorizadas surgieron ante a ineficiencia de la linea de interrupción única ya que el procesador se tomaba mucho tiempo para verificar que modulo realizo la petición de interrupción. Ahora con las interrupciones vectorizadas, cuando un módulo emita una interrupción, este enviaría un vector que funciona como identificador, 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 lo coloca el periférico directamente o el PIC, el cual se ocupa de todo.

1. **¿Cuándo, porqué, para que y como 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 para permitir que los programas realicen tareas en segundo plano mientras se espera que ocurra una interrupción de hardware. Son interrupciones enmascarables que en la mayoría de los casos tienen una manera de poder tratarlos y poder continuar con el procesamiento normal sin necesidad de tener que detener la ejecución del programa. Generalmente son utilizadas para hacer llamadas a funciones del SO. Esta característica permite que las subrutinas del sistema se carguen en cualquier lugar. No requieren conocer la dirección de la rutina en tiempo de ejecución.

1. **Esquematice y describa la estructura interna de un módulo de E/S. (Consultar cuál de las dos respuestas está bien)**

El modulo se conecta al resto del computador a través de un conjunto de líneas (líneas de datos, líneas de dirección y líneas de control). Los datos que se transfieren a y desde el modulo 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. Estos registros internos pueden ser leídos y escritos por la CPU y son el medio de intercambio de datos con los dispositivos externos. La lógica que hay en el módulo interactúa con el procesador a través de una serie de líneas de control. Estas líneas las utiliza el procesador para proporcionar las ordenes al módulo de E/S. Algunas de estas líneas de control pueden ser utilizadas por el módulo de E/S (por ejemplo, para las señales de arbitraje del bus). El módulo también debe ser capaz de reconocer y generar las direcciones asociadas a los dispositivos que controla. Cada módulo de E/S tiene una dirección única o, si controla más de un dispositivo externo, un conjunto único de direcciones. Por último, el módulo de E/S también posee lógica específica para la interfaz con cada uno de los dispositivos que controla.



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

Cuando la CPU, memoria principal e E/S comparten un bus común, hay dos modos de direccionamiento posibles: mapeado de memoria y aislado. Con E/S mapeado de memoria existe un único espacio de direcciones para las posiciones de memoria y los módulos E/S. La CPU trata los registros de datos y estado de los módulos como posiciones de memoria, y utiliza las mismas instrucciones máquina para acceder a memoria y E/S. Las líneas del bus de direcciones dan un número máximo de posiciones a direccionar, que estarán repartidas entre memoria y E/S en alguna proporción. En el otro caso, el bus de control dispone de líneas de lectura/escritura en memoria más otras en entrada/salida. En este caso, el comando especifica cuándo una dirección se refiere a memoria o E/S. Se puede utilizar todo el rango del bus de direcciones para direccionar memoria y E/S de forma independiente, de ahí el nombre de E/S aislada.

1. **Describa las funciones de correspondencia entre Memoria Principal y Cache.**

La función de correspondencia en la memoria caché determina cómo se asignan los bloques de memoria de la memoria principal a los bloques de la memoria caché. Hay tres técnicas principales de correspondencia:

1. **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.

2. **Correspondencia por asociación:** En este caso, 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.

3. **Correspondencia por conjuntos**: En este tipo, 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.

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

Las políticas de escritura en caché son importantes para garantizar la coherencia de los datos entre la memoria principal y la caché. Las políticas comunes son:

- En aciertos, escritura inmediata y post escritura. La escritura inmediata pone como prioridad la integridad de los datos, por lo que, si se almacena en cache, también se hará en memoria principal mientras que la post escritura pone como prioridad el menor acceso a memoria posible, mejorando así la performance y reduciendo la cantidad de accesos, almacenando los datos en cache y solo actualizando en memoria principal cuando se necesite hacer un reemplazo.

- En fallos, write allocate que en esta política la información se lleva de la memoria principal a la cache. Se sobrescribe en la cache y no-write allocate que en este caso el bloque no se lleva a la memoria cache. Se escribe directamente en la memoria principal.

1. **¿Qué características tienen los procesadores superescalares? Describa las causas que pueden retardar el funcionamiento de los mismos.**

Los procesadores superescalares son capaces de ejecutar múltiples instrucciones en paralelo e independientemente, lo que aumenta la velocidad de procesamiento. El objetivo principal es realizar la ejecución de varias instrucciones, es decir, finalizar más de una instrucción a la vez, por lo que hace que uno llegue a pensar que esto generará que se requieran más recursos a tal punto de duplicar partes de CPU o ALU para una ejecución paralela o almacenar en memoria distintos datos en un mismo ciclo. Sin embargo, esto puede verse afectado por causas como las dependencias de datos, las dependencias de control y las limitaciones de recursos, como la capacidad limitada de las unidades de ejecución y la memoria caché, lo que puede ralentizar su funcionamiento y afectar su rendimiento.

1. **¿Qué elementos característicos definen un bus?**

Los elementos característicos que definen un bus son:

* **Ancho de banda**: determina la cantidad de datos que se pueden transmitir simultáneamente en el bus.
* **Velocidad de reloj:** establece la frecuencia a la que se transfieren los datos en el bus.
* **Latencia**: representa el tiempo que tarda un mensaje en ser transmitido desde la fuente hasta el destino.

1. **¿Qué son los MIMD de las taxonomías de Flynn?**

Los MIMD (Multiple Instruction, Multiple Data) son una de las categorías de la taxonomía de Flynn que hace referencia al conjunto de procesadores que ejecutan conjuntos de instrucciones y utilizan conjuntos de datos diferentes por procesador, con acceso o a una memoria compartida o a una memoria distribuida. Esto significa que son capaces de procesar múltiples instrucciones al mismo tiempo y utilizar múltiples datos en paralelo. Se pueden sub-clasificar por las formas de comunicación y gestión de memoria en:

- Memoria compartida (Multiprocesadores Simétricos (SMP) y Sistema NUMA.

- Memoria Distribuida (Cluster).

**Final Arqui**

1. **¿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, por referencia. También cuando hablamos de pasajes de argumentos a subrutinas pueden ser:

- **Vía Registros:**

* El número de registros es la principal limitación.
* Es importante documentar que registros se usan.

**- Vía Memoria**:

* Se usa un área definida de memoria (RAM)
* Difícil de estandarizar.

- **Vía Pila:**

* Es el método más ampliamente usado.
* El “verdadero pasaje de parámetro”
* Independiente de memoria y registros.
* Hay que comprender bien cómo funciona porque la pila es usada por el usuario y por el sistema.

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

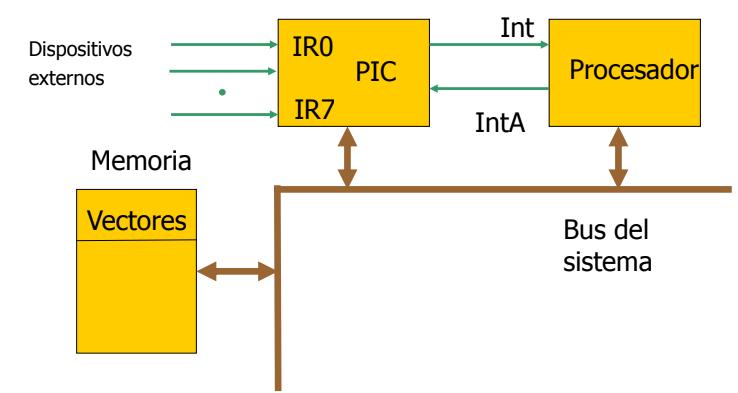
La principal diferencia entre la terminación de una subrutina y un gestor de interrupción es que una subrutina se llama y termina su ejecución de manera normal, mientras que un gestor de interrupción se ejecuta como respuesta a una interrupción externa y luego vuelve a la ejecución de la tarea principal. 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.

1. **Esquematice y describa la estructura interna de una Controlador Programable de Interrupciones.**

El dispositivo controlador programable de interrupciones (PIC) es un chip que sirve si el procesador tiene una única entrada de pedido de interrupciones o si tenemos varios productores de interrupciones.

Es un dispositivo usado para combinar varias fuentes de interrupciones sobre una o más líneas del CPU. Se maneja con prioridades (las de número más bajo se atienden primero) y tiene un conjunto de registros internos:

* IRR (Interrupt Request Register): Registro de petición de interrupción, indica con bit en 1 las interrupciones demandadas hasta el momento.
* ISR (In Service Register): Registro de interrupción en servicio, indica con bit en 1 cuál es la interrupción que está siendo atendida.
* IMR (Interrupt Mask Register): Registro de maścara de interrupciones, permite el enmascaramiento selectivo de cada una de las entradas de interrupción, indicando con bit en 1. Tras un reset los bits de este registro quedarán en 0. Indica cuáles deben ser ignoradas.
* EOI (End of Interruption): Fin de interrupción. Como consecuencia, se pone en 0 el bit del ISR correspondiente.
* INT0...INT7: 8 registros, donde carga el valor del vector de interrupción correspondiente



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

En la gestión E/S programada con espera de respuesta, es un intercambio de datos entre la CPU y el módulo. La CPU mantiene el control total de la operación de E/S que hay que realizar, como, por ejemplo, la comprobación de estado del dispositivo. Le ordena al módulo y dispositivo que debe ejecutarse y que hacer a través de una orden. Periódicamente, la CPU pregunta por los bits de estado del módulo que indican si la E/S finalizo, ya que este no tiene la capacidad de notificar cuando terminó. Mientras tanto la CPU se mantiene ociosa a la espera, lo cual resulta ineficiente.

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

Los elementos a tener en cuenta para el diseño de una memoria caché incluyen: Organización (tamaño, costo y niveles), tipo de función de correspondencia (Directa, Asociativa y por Conjuntos), la política de reemplazo (LRU (menos recientemente usado), FIFO (más antiguo), LFU (menos frecuentemente usado), Aleatorio), la política de escritura, la velocidad de acceso y la coherencia de caché.

1. **¿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.

1. **¿De qué depende el paralelismo de una máquina superescalar?**

El paralelismo de una máquina superescalar depende de la capacidad del procesador para ejecutar múltiples instrucciones en paralelo y de manera independiente, así como de la capacidad de identificar y resolver las dependencias de datos y de control entre las instrucciones. También depende de la cantidad y tipo de unidades funcionales disponibles en el procesador, así como de la eficiencia de la caché y la memoria.

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

Cuando se utilizan técnicas de desordenación los valores de los registros no pueden conocerse completamente en cada instante de tiempo. Las instrucciones entran en conflicto por el uso de registros y el procesador debe detener alguna etapa para resolverlo.

Las técnicas de software de optimización de registros empeoran la situación.

Los efectos de las dependencias de salida pueden disminuirse por esta técnica, que consiste en disponer de registros adicionales (internos, ocultos al programador) y asignarlos (por hardware) a instrucciones en conflicto. Existen R3a y R3b, por ejemplo.

1. **¿Qué elementos característicos definen un bus?**

Los elementos característicos que definen un bus son:

* **Ancho de banda**: determina la cantidad de datos que se pueden transmitir simultáneamente en el bus.
* **Velocidad de reloj:** establece la frecuencia a la que se transfieren los datos en el bus.
* **Latencia**: representa el tiempo que tarda un mensaje en ser transmitido desde la fuente hasta el destino.

**Final de Arquitectura de Computadoras – Mesa 04/23**

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

El controlador de DMA es un dispositivo controlar una transferencia de datos entre un periférico y memoria sin intervención de la CPU. El DMA se encarga de todo el procesamiento de E/S para que la CPU no esté al pendiente de la ejecución de la operación y/o gestión de interrupciones, lo cual en transferencias masivas de E/S provocaría que este se mantuviera ocupada en gran parte para la ejecución de este tipo de operaciones haciendo que su performance se vea perjudicada.

La técnica aplicada por el DMA beneficia a la gestión de este tipo de operaciones haciendo que este tome el control total de la transferencia de E/S por lo cual, este cuenta con un controlador (DMAC) el cual toma el control del bus del sistema de forma arbitraria para no tener conflictos con la CPU al momento de realizar la operación e indica las direcciones implicadas para el almacenamiento de los datos a transferir, así como también el tipo de operación que se va a realizar (IN/OUT).

Como única tarea por parte de la CPU es inicializar al DMA y al periférico indicando los parámetros de transferencia tales como la del tipo de transferencia que tienen que realizar, la cantidad de bits o palabras que deben transferir, la ubicación a memoria o de memoria que se ve implicada en la operación y parámetros de control o de n de canales en caso de que se cuenten con DMA con varios canales. Una vez que realiza esto, la CPU se desliga del procesamiento de E/S.

1. **SEGMENTACION DE CAUCE.**
2. **Definición**

La segmentación de cauce o Pipeline es una manera de organizar la ejecución de procesos entre el hardware para llevar a cabo su procesamiento simultaneo, es decir, varios procesos a la vez descomponiéndose en fases o cauces permitiendo así el paralelismo entre las instrucciones que ejecuta un procesador.

1. **Rendimiento.**
2. **Describa el problema y posibles soluciones ante riesgos por transferencia de control de programa.**

Los atascos son situaciones que impiden que las instrucciones se ejecuten en el ciclo que le corresponde. Pueden ser:

**Estructurales:** Este caso sucede cuando las dos o más instrucciones necesitan hacer uso de un mismo recurso de hardware en un cauce.

**Por dependencia de datos:** Este caso sucede cuando las instrucciones imponen una comunicación entre ellas mediante datos y esta no llega a estar disponible en el ciclo que se lo requiere por lo que genere ese atasco. Cuando las instrucciones requieren el uso de datos que algunos casos están siendo usados por otras instrucciones o están siendo accedidas por lo que esto repercute en la espera hasta su liberación. Este tipo de atascos se clasifican en 3 tipos:

*Dependencia de datos verdadera (RAW)*

*Dependencia de datos de salida (WAW)*

*Dependencia de control o antidependencia (WAR)*

**Por dependencia de control:** Estos tipos de atasco ocurren cuando una instrucción en una etapa del cauce necesita esperar a que se complete la ejecución de una instrucción anterior en otra etapa del cauce antes de poder continuar su propia ejecución.

**Las posibles soluciones a los atascos son:**

* *Para atascos estructurales:*

- **Duplicación de recursos:** Aumentar los recursos o unidades para evitar que dos instrucciones compitan por ellos.

- **Separación en memoria de instrucciones y datos:** Una técnica de diseño que podría dar solución a estos atascos podría ser la separación en distintos lugares de memoria las instrucciones y los datos.

- **Mecanismo de sincronización de acceso al banco de registros:** Por ultimo buscar un mecanismo de tal forma que haya un acceso sincronizado al banco de registros. Como ejemplo, seria qué en la primera mitad de los ciclos del reloj sean todas instrucciones y en la segunda mitad todas de lectura.

* *Para atascos de datos:* Técnica de detección de dependencia por software y por hardware

- **Por Software:** Para este tipo de atascos es necesario detectarlo a tiempo para poder prevenir que pase. Se suele contar con unidades que se encargan de esta detección o bien se cuenta con un compilador más complejo que puede realizar esta tarea.

Una posible solución consiste en realizar la **Inserción de Instrucciones (NOP)**, así poder generar un retardo de aquellas instrucciones que no deberían ejecutar su cauce porque generarían dependencia de datos.

Otra solución es la **Reordenación de las Instrucciones** para que de alguna manera se mantengan separadas en el cauce las que podrían generar conflictos teniendo en cuenta que su ejecución no se dé “fuera de orden”.

- **Por Hardware:** La solución para este caso es el **Adelantamiento de Operando** (forwarding). Donde las instrucciones cuentan con ellos apenas hayan sido calculados en su cauce, es decir, sin haber sido almacenados en registros o memoria.

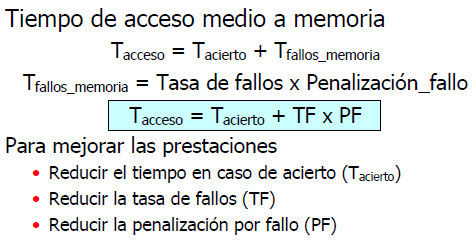
* *Para atascos de control:*

- **Predicción de Saltos:** Esta técnica intenta predecir si un salto condicional se tomara o no antes de que se calcule la dirección de salto. Si la predicción es correcta, se evita el atasco de control y se sigue ejecutando el programa. Si la predicción es incorrecta, se deshace el trabajo realizado y se reanuda la ejecución del programa desde la nueva dirección de salto.

- **Predicción de Saltos dinámicas:** Esta técnica utiliza la historia de los saltos anteriores para predecir el comportamiento de los saltos futuros.

- **Saltos Retardados:** Esta técnica permite que el procesador continúe ejecutando instrucciones después un salto incondicional, lo que reduce el impacto de los atascos de control.

1. **MEMORIA CACHE.**
2. **Si se pretende mejorar el tiempo de acceso medio a memoria cache. ¿Sobre qué parámetros será necesario trabajar y propone como medida para hacerlo?**

******

1. **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.**
2. **PROCESADORES SUPERESCALARES.**
3. **¿Cuáles son las características distintivas?**

Los procesadores superescalares son capaces de ejecutar múltiples instrucciones en paralelo e independientemente, lo que aumenta la velocidad de procesamiento. El objetivo principal es realizar la ejecución de varias instrucciones, es decir, finalizar más de una instrucción a la vez, por lo que hace que uno llegue a pensar que esto generará que se requieran más recursos a tal punto de duplicar partes de CPU o ALU para una ejecución paralela o almacenar en memoria distintos datos en un mismo ciclo

1. **Compare las políticas de emisión de instrucciones.**

La ***emisión es alineada*** si no pueden introducirse nuevas instrucciones en la Ventana de Instrucciones hasta que ésta no está totalmente vacía.

En la ***emisión no alineada***, mientras que exista un espacio libre en la ventana, se pueden ir introduciendo nuevas instrucciones para ser emitidas.

En la ***emisión ordenada*** se respeta el orden en que las instrucciones se han ido introduciendo en la Ventana de Instrucciones. Este orden es el mismo en que las instrucciones se han ido decodificando, y coincide con el orden de las instrucciones en el programa

En el caso de la ***emisión desordenada*** no existe bloqueo, ya que pueden emitirse todas las instrucciones que dispongan de sus operandos y de una unidad funcional del tipo apropiado.

1. **Elija una alternativa de emisión/finalización y justifique.**

Una alternativa de emisión/finalización puede ser el Renombramiento de registros, que es la técnica utilizada en procesadores superescalares para evitar las dependencias de datos. Cuando se utilizan técnicas de desordenación los valores de los registros no pueden conocerse completamente en cada instante de tiempo. Las instrucciones entran en conflicto por el uso de registros y el procesador debe detener alguna etapa para resolverlo. Las técnicas de software de optimización de registros empeoran la situación. Los efectos de las dependencias de salida pueden disminuirse por esta técnica, que consiste en disponer de registros adicionales (internos, ocultos al programador) y asignarlos (por hardware) a instrucciones en conflicto. Existen R3a y R3b, por ejemplo.

1. **PROCESAMIENTO PARALELO.**
2. **Describa las 4 variantes de arquitectura de la Taxonomía de Flynn.**

***SISD (Single Instruction, Single Data):*** Esta categoría define los procesadores que ejecutan una sola secuencia de instrucción y que procesan un solo conjunto de datos, lo cual, es común en Monoprocesadores. Es la arquitectura más simple, ineficiente si se quiere lograr paralelismo, eficiente ante tareas simples y poco procesamiento de datos.

***SIMD (Single Instruction, Multiple Data):*** Esta categoría define una arquitectura de un solo conjunto de instrucciones con la capacidad de manejo de varios conjuntos de datos de forma simultánea, es decir, un conjunto por procesador que tiene un sector de memoria dedicado. Esta categoría se relaciona con procesadores vectoriales y matriciales, donde cada uno, con su grado de complejidad, tienen las capacidades de manejar varias cantidades de datos por instrucción, ya sea en vectores o matrices de una manera eficiente y paralela.

***MISD (Multiple Instruction,******Single Data)***: Categoría menos usada donde se trata de representar una arquitectura donde se transmitan entre todos los procesadores distintos, secuencias de instrucciones donde a su vez utilicen un solo conjunto de datos.

***MIMD (Multiple Instruction, Multiple Data):*** Conjunto de procesadores que ejecutan conjuntos de instrucciones diferentes y utilizan conjuntos de datos diferentes por procesador, con acceso o a una memoria compartida o a una memoria distribuida.

1. **Compare los sistemas MP y clusters.**

* Ambos dan soporte a aplicaciones de alta demanda de recursos y también están disponibles comercialmente (SMP es más antiguo)
* SMP:

- Más fácil de administrar y configurar.

- Cercano a los sistemas de un solo procesador.

- La planificación (scheduling) es la diferencia principal.

- Menos espacio físico / Menor consumo de potencia.

* Cluster:

- Superior escalabilidad incremental y absoluta.

- Superior disponibilidad

- Redundancia.

**Final Arquitectura de Computadora – Mesa 06/23**

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

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

**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.**

**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?**

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

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

**a)- ¿Qué elementos característicos definen un bus?**

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