Clase 1.1: Repaso de Organización de computadoras.

Repertorio de instrucciones: Es el conjunto de instrucciones que un procesador puede ejecutar. Es el conjunto completo de instrucciones maquina que una computadora puede entender y llevar a cabo.

Ejemplo: operaciones: ADD (Suma) SUB(Resta). De operandos: ADD BX, dato1

Decisiones en el diseño de conjunto de instrucciones: Es un aspecto fundamental en la arquitectura de computadoras. Afecta a la implementación del procesador y a la forma en que los programadores lo controlan. Entre los aspectos mas importantes se encuentran:

- Repertorio de operaciones: Se debe decidir cuantas y que operaciones considerar, asi como la complejidad de cada operación.
- Tipos de datos: Consiste en definir los tipos de datos con los que se efectúan las operaciones, por ejemplo, direcciones,números,caracteres.
- Formato de instrucción: Se debe establecer la longitud de la instrucción (en bits), el número de direcciones, el tamaño de los campos.
- Registros: Es importante definir el numero de registros
- Direccionamiento: Se deben elegir los modos de direccionamiento mediante los cuales puede especificarse la dirección de un operando.

Cuales son los elementos de una instrucción?

Codigo de operación (codop): Especifica la operación a realizar. Por ejemplo suma, resta, e/s. Se especifica mediante un código binario, conocido como codop.

Referencia a operando fuente: Especifican los operandos o datos, que sirven como entrada para la instrucción. Puede requerir uno o mas operandos fuente.

Referencia al operando destino: Especifica en donde se almacena el resultado de la operación. No todas las operaciones producen un resultado, pero aquellas que lo generan, deben tener la referencia al operando destino.

Referencia a la siguiente instrucción: Le indica al procesador donde obtener la siguiente instrucción, después de completar la ejecución de la instrucción actual. Este dato, se almacena en el PC (Program Counter).

Clase 1.2: Pasaje de argumentos a subrutinas.

Subrutinas:

Una subrutina es una secuencia de instrucciones, que realiza una tarea específica, puede invocarse desde cualquier punto de un programa.

Requieren dos instrucciones: CALL (invocarla) y RETURN (retornar de la subrutina), ambas instrucciones son de bifurcación. Cuando se llama a una 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 más pequeñas.

Pasaje de argumentos a subrutinas:

Vía registros: En este método, los datos se almacenan en los registros antes de invocar a la subrutina y dentro de la subrutina se acceden a los datos a través de los registros. La ventaja de utilizar los registros es que son de rápido acceso. La desventaja es el número de registros que son limitados. Es importante documentar que registros se usan.

Vía memoria: Se usa un área definida de memoria (RAM). Los datos se almacenan en ubicaciones especificas de la memoria RAM. Es difícil de estandarizar. No es una buena solución por el peligro de que algún otro programa, o el S.O sobrescriba una dirección de memoria.

Vía Pila(stack): Es el método más usado, se considera el verdadero pasaje de parámetros. Es independiente de memoria y registros. En este método, los datos son apilados antes de invocar a la subrutina, y dentro de la subrutina, se desapilan para poder acceder a los datos. La única limitación es comprender el funcionamiento de la pila, porque es utilizada tanto por el usuario y por el sistema.

Procedimientos/Subrutinas:

Si tenemos parámetros: Antes de llamar a la subrutina, debemos apilar los parámetros, para luego dentro de la subrutina, acceder a los parámetros.

Ejemplo

```
Push Parametro 1
Push Parametro 2
Call  Nombre
```

: Con el CALL, hacemos el llamado a la subrutina

Al momento de llamar a la subrutina, se debe guardar la dirección de retorno de la subrutina, la cual se almacena en el registro PC (dirección de la próxima instrucción a ejecutar)

Las subrutinas, retornan con la instrucción RET, cuando se ejecuta la instrucción RET, debemos tener el SP en la dirección de retorno

Pasos al momento de trabajar con subrutinas/procedimientos:

1. Salvar el estado de BP (viejo BP)
2. Salvar estado de SP (BP=SP)
3. Reservar espacio para datos locales (opcional)
4. Salvar valores de otros registros (opcional)
5. Acceder a parámetros
6. Escribir sentencias a ejecutar
7. Retornar parámetro (opcional)
8. Regresar correctamente del procedimiento

Que es una pila?

La pila es una estructura de tipo LIFO (ultimo en entrar, primero en salir), 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 número de elementos en la pila es variable. El ultimo elemento apilado, está indicado por el SP (stack pointer)

Caracteristicas:

Solo se pueden añadir o eliminar elementos en la cabecera de la pila

Tenemos dos instrucciones PUSH (apila), POP (desapila).

Clase 2: Interrupciones

Las interrupciones, proporcionan una forma de mejorar la eficiencia del procesador.

Una interrupción es un mecanismo el cual permite interrumpir el procesamiento normal de la CPU. Una interrupción, permite que la CPU responda solo cuando se genere algún evento, o cuando se necesite la atención del procesador. Las interrupciones pueden ser de origen externo o interno a la CPU.

Mientras la CPU está ejecutando, si recibe un pedido de interrupción, y esa interrupción esta habilitada, se detiene la ejecución del programa en curso, la CPU salva su contexto, guarda el PC y el registro de estado y salta a una rutina de gestión de interrupción. Una vez que termina de atender la interrupción, se recupera el PC, y se sigue con la siguiente instrucción, siempre y cuando haya otra instrucción que ejecutar y que no haya ningún otra interrupción para atender.

Que hacer si interrumpe al procesador?

Implica transferir el control a un programa (el GESTOR de interrupción), el cual es el responsable de:

- Salvar el estado del procesador.
- Corregir o responder a la causa que genero la interrupción
- Restaurar el estado original del procesador
- Retornar a la ejecución normal del programa interrumpido.

Tipos de interrupciones:

TRAPS: 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.

Interrupciones por software: Llaman a funciones del sistema operativo, generadas mediante la instrucción INT, la cual hacen que se ejecute una subrutina del sistema operativo.

Interrupción por HARDWARE: Generadas por eventos externos al procesador, como dispositivos periféricos.Estas interrupciones son **asincrónicas**, no están vinculadas a la ejecución del programa actual, sino que este tipo de interrupciones pueden ocurrir en cualquier momento. Las interrupciones por hardware son necesarias para lidiar con la diferencia de velocidad entre la CPU y los dispositivos de e/s.

En las interrupciones por hardware, el que interrumpe es un dispositivo y la CPU pasa el control a una subrutina especial llamada manejador de interrupciones. En las interrupciones por software, el que interrumpe es el mismo programa, y la CPU pasa el control al sistema operativo, que se encarga de ejecutar una subrutina de sistema.

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.

Interrupciones enmascarables y no enmascarables:

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.

***Las interrupciones Enmascarables*:** Son aquellas que pueden ser ignoradas temporalmente o pueden descartarse, 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.

Ciclo de interrupción:

- Se comprueba si se ha solicitado alguna interrupción.
- Si no hay señal de interrupción, se capta la siguiente instrucción.
- Si hay algún pedido de interrupción pendiente:
    - Suspende la ejecución del programa en curso
    - Guarda su contexto (próxima instrucción a ejecutar y el estado del procesador)
    - Carga el PC con la dirección de comienzo de una rutina de gestión de interrupción.
    - Finalizada la rutina de gestión, el procesador retorna la ejecución hacia el punto de interrupción.

Reconocimiento de interrupciones:

- **Interrupciones multinivel:** Cada dispositivo que puede provocar una interrupción, tiene una entrada física de interrupción conectada a la CPU. Es sencillo pero muy caro.
- **Línea de interrupción única:** Existe una sola entrada física de pedido de interrupción, la cual comparten todos los dispositivos, en este método, se debe preguntar a cada dispositivo si ha producido el pedido de interrupción (mejor conocido como técnica de polling/encuesta)
- **Interrupciones vectorizadas:** 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.
- El dispositivo que quiere interrumpir, además de la señal de pedido de interrupción, debe colocar en el bus de datos un identificador.

Por que son necesarias las interrupciones?

Las interrupciones son necesarias, porque permite a la CPU ejecutar tareas normales, y responder a las interrupciones solo cuando ocurre un evento. Sin las interrupciones, el procesador tendría que emplear técnicas como el polling, donde revise constantemente el estado de los dispositivos, esto resulta ineficiente, porque consume ciclos de reloj, incluso cuando no hay eventos que manejar.

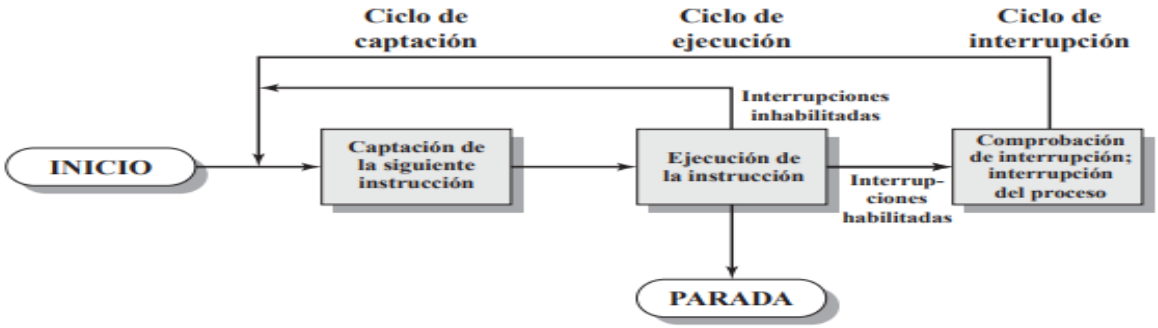
Cual es la relación entre las interrupciones y las operaciones de E/S?

E/S por interrupción: Antes de iniciar la operación, el controlador de E/S se configura para generar una interrupción cuando la operación E/S se complete. De esta manera, mientras el controlador de E/S se encarga de realizar la operación, el procesador puede continuar ejecutando otras tareas y no queda inactivo. Una vez que se complete la operación, se genera una interrupción, entonces el procesador interrumpe sus tareas y ejecuta una rutina de servicio de interrupciones. Después de manejar la interrupción, el procesador retorna 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**

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.

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.



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.

Que diferencia hay entre las llamadas a subrutinas y las interrupciones?

En ambos casos, al invocarlas, el PC (Program Counter), salta a una dirección de memoria, las interrupciones por software y las subrutinas, son sincrónicas, es decir, están relacionadas con la ejecución del programa actual. Aquellas interrupciones que son distintas, son las interrupciones de hardware, son asincrónicas, surgen en cualquier momento, y a parte involucran elementos distintos, particularmente el PIC, mediante el PIC podemos administrar los pedidos de interrupción de hardware.

Las interrupciones por hardware tienen dos etapas

- El PIC le avisa a la CPU el pedido se interrupción
- El PIC le avisa a la CPU el dispositivo que genero la interrupción.

Primeros 6 minutos explica esto

PIC (Controlador de interrupción programable):

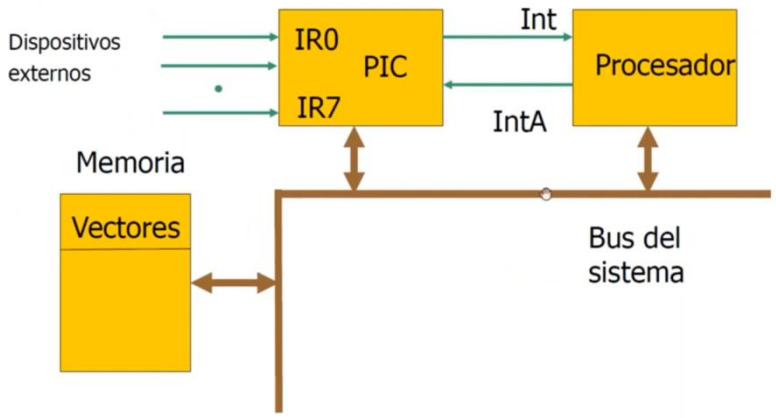
Se encarga de gestionar las interrupciones, permitiendo que múltiples dispositivos periféricos compartan líneas de interrupción del procesador. El PIC es el responsable de recibir señales de interrupción de varios periféricos y dirigir estas solicitudes a la CPU. El PIC es programable, se puede configurar 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.



Clase 3.1: Entrada / Salida

Problemas de entrada/salida:

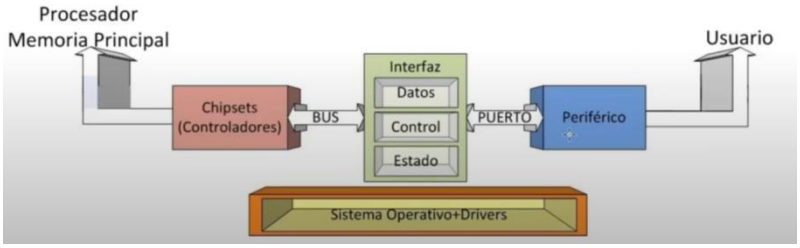
Gran variedad de periféricos con varios métodos de operación:

- Transmisión de diferentes cantidades de datos
- Diferentes velocidades

Todos los periféricos son más lentos que la CPU y la RAM, por lo tanto, necesitan de módulos de e/s.

Que es un modulo de entrada/salida?

Un modulo de entrada/salida, es un componente esencial en la computadora, actúa como intermediario, entre el procesador, la memoria, y los periféricos. El modulo de e/s está conectado al bus del sistema, y se encarga de gestionar el flujo de datos y control entre los dispositivos conectados a él. Permite al procesador que interactué con una amplia variedad de dispositivos. Es capaz de conocer y generar direcciones asociadas a los dispositivos que controla.



Como se conectan los periféricos?

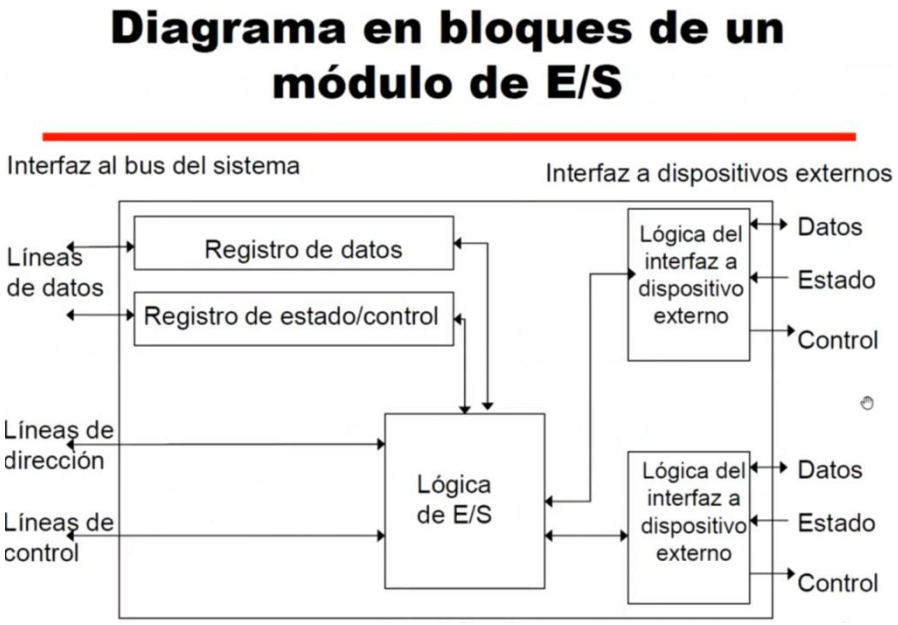
Generalmente a través de un puerto. Un puerto es una interfaz entre el periférico y el modulo de e/s. Existen dos tipos de puertos.

Puerto paralelo: Hay múltiples líneas que conectan el modulo de e/s y el periférico, lo que permite la transferencia simultanea de varios bits a través del bus de datos. Se utilizan generalmente para dispositivos de alta velocidad **Cada dispositivo de e/s tiene su propio espacio de direcciones**

Puerto serie: Hay solo una línea para transmitir los datos, por lo que los bits deben transmitirse uno a uno de forma secuencial. Este tipo de puertos, son mas adecuados para dispositivos de velocidad mas baja, como impresoras.

Cuales son las funciones de un modulo de entrada/salida?

Un módulo de entrada/salida (E/S) es responsable de controlar uno o varios periféricos, actúa como intermediario entre la CPU, la memoria y los periféricos. Su función principal es interpretar las órdenes de la CPU y transmitirlas a los periféricos mediante registros de control. Además, gestiona las transferencias de datos, lo que implica adaptar velocidades, convertir formatos y almacenar temporalmente información en un buffer. También supervisa el estado de los periféricos, informando a la CPU sobre su disponibilidad o posibles fallos, y es capaz de detectar y corregir errores para garantizar una comunicación eficiente y segura



Operación de entrada/salida

Es una transferencia de datos, desde o hacia un periférico.

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.

- **E/S mapeada en memoria:** Hay una cierta cantidad de direcciones de memoria, que van a ser especificas para entrada/salida, y no pueden ser utilizadas para espacios de memoria libre. De esta manera, no va a haber necesidad de manejar instrucciones distintas, por que sabemos que direcciones están reservadas a un puerto de e/s.

- **E/S aislada:** Tenemos espacios de direcciones separados, tenemos por un lado, espacio de memoria principal, y espacio de memoria asociadas a operaciones de entrada/salida.

Técnicas de E/S:

E/S programada con espera de respuesta: Se refiere a que el procesador emite una solicitud de operación de E/S. **El procesador espera activamente hasta que el dispositivo de 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 encarga de realizar la operación, el procesador puede continuar ejecutando otras tareas y no queda inactivo. Una vez que se **complete la operación, se genera una interrupción**, entonces el procesador interrumpe sus tareas y ejecuta una rutina de servicio de interrupciones. Después de manejar la interrupción, el procesador retorna 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**

E/S con 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.

Clase 3.2: Entrada/Salida

Cuál es la mejor técnica de entrada/salida? (suelen preguntarlo en finales)

Depende, depende del periférico que sea, de la cantidad de datos que vaya a transferir.

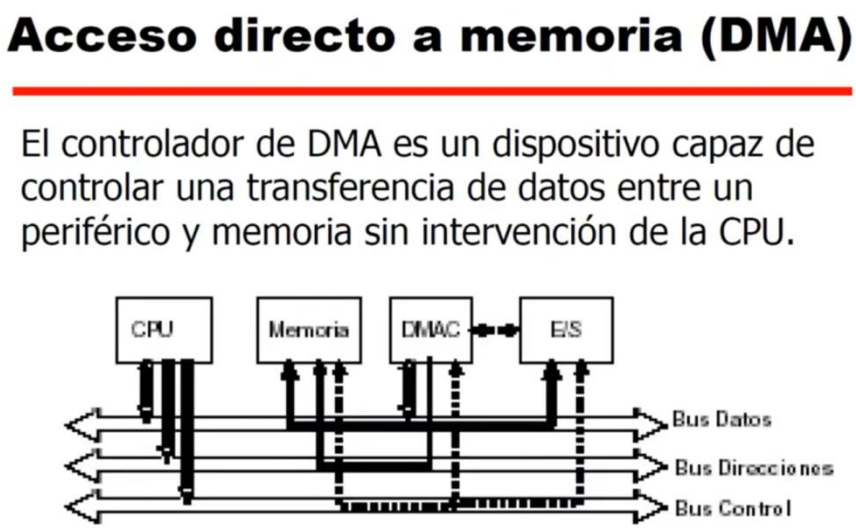
Por ejemplo, si se está utilizando un teclado, la técnica adecuada es e/s por interrupcion, de esta manera, cuando se presiona una tecla, el teclado genera una interrupcion y el procesador lee el código de la tecla. Con el mouse pasa lo mismo, es conveniente la e/s por interrupcion, ya que los movimientos del mouse y los clics generan interrupciones.

En general, para dispositivos de velocidad baja, como teclados y mouses, la técnica de e/s con interrupcion es eficiente. Pero para dispositivos de alta velocidad que requieran transferencias de grandes volúmenes de datos, el DMA es la mejor opción.

Si es un periférico rápido, la técnica de DMA va a ser la más adecuada.

Acceso directo a memoria (DMA)

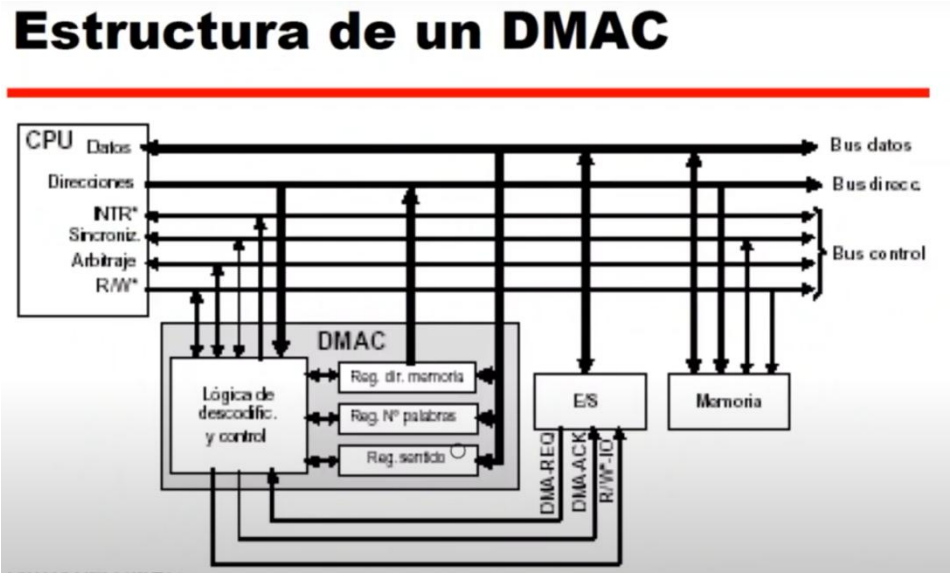
El objetivo de esta técnica es tener una transferencia de datos lo más rápido posible.
El controlador de DMA (DMAC) es un dispositivo capaz de controlar una transferencia de datos entre un periférico y la memoria sin intervención directa de la CPU. La CPU no interviene en el proceso de transferencia de datos, pero va a intervenir al principio y al final de la transferencia de datos. El DMA, al liberar a la CPU de la tarea de gestionar cada transferencia, se dice que el DMA reduce la carga del procesador.



Funcionamiento del DMA

El DMAC actúa como maestro del bus durante la transferencia DMA y debe ser capaz de:

- Solicitar el uso del bus mediante señales y la lógica de arbitraje necesarias
- Especificar la dirección de memoria sobre la que se realiza la transferencia
- Generar señales de control de bus

- 

<u>Etapas de una transferencia DMA:</u>

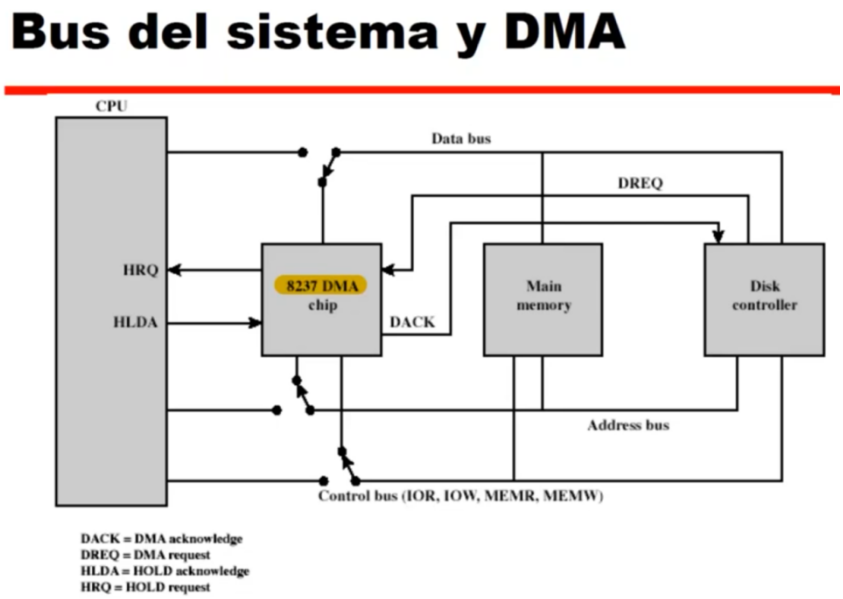
- **Inicialización de transferencia:** La CPU debe enviar los parámetros de transferencia hacia el DMAC y la interfaz del periférico.
    - Se debe configurar el DMAC:
        - Numero de bytes
        - Tipo de transferencia (Lectura/Escritura)
        - Direccion de memoria inicial para la transferencia
        - Numero de canal de DMA
- **Realizacion de la transferencia:**
    - Cuando el periférico esta listo para realizar la transferencia , se lo indica al DMAC.
    - El DMAC pide el contro l del bus y realiza la transferencia entre el periférico y la memoria.
- Finalizacion de la transferencia: El DMAC libera el bus y devuelve el control a la CPU. El DMAC, suele activar una señal de interrupcion para indicar a la CPU la finalización de la e/s.

Se puede degradar el rendimiento de la CPU si el DMAC hace uso intensivo del bus.

Tipos de transferencias DMA

Transferencia por ráfagas: El DMAC solicita el control del bus a la CPU. Cuando la CPU le concede el bus, el DMAC no libera el bus hasta haber finalizado la transferencia de todo el bloque de datos. **Ventajas**: La transferencia se realiza de forma rápida.
Desventajas: Durante el tiempo que dura la transferencia, la CPU no puede utilizar el bus de memoria, lo que puede degradar el rendimiento del sistema. **Es optimo para operaciones cortas.**

Transferencia por bloques: El DMA solicita el control del bus a la CPU. Cuando la CPU concede el bus al DMAC, se realiza la transferencia de una única palabra y después el DMAC libera el bus. El DMAC solicita el bus tantas veces como sea necesario hasta finalizar la transferencia del bloque completo. **Ventajas:** No se degrada el rendimiento del sistema. **Desventajas**: La transferencia tarda mas tiempo en llevarse a cabo.



Tipos de canales de E/S:

- **Selector:** Controla varios dispositivos de alta velocidad, pero solo uno a la vez. Por lo tanto el canal se dedica para la transferencia de datos de ese dispositivo. El canal selector, selecciona un dispositivo y efectúa la transferencia.
- **Multiplexor:** Comparte varias funciones o dispositivos. Transmite diferentes tipos de 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.

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.

Clase 4: Segmentación de instrucciones

Que es la segmentación de cauce?

La segmentación de cauce, es una técnica que está orientada a la optimización del uso del hardware de la CPU, permite realizar más de una tarea al mismo tiempo. El objetivo de utilizar la técnica de segmentación de cauce, es lograr una mayor cantidad de operaciones en el mismo tiempo. La segmentación de cauce, consiste en descomponer el proceso de ejecución de las instrucciones en fases o etapas que permitan una ejecución simultánea.

Esta técnica, explota el paralelismo entre las instrucciones de flujo secuencial.

Es una técnica propia por la arquitectura RISC.

Cual es el rendimiento de la segmentación de cauce?

El máximo rendimiento, CPI = 1, lo que significa que en todos los ciclos finaliza una instruccion

El incremento potencial de la segmentación de cauce, es proporcional al número de etapas del cauce.Logramos la productividad por que voy a tener en un ciclo de reloj, varias etapas de diferentes instrucciones, pero no necesariamente reduce el tiempo de ejecución de la instrucción.

Si K es el numero de etapas del cauce => Vel. Procesador segmentado Vel. secuencial * K

Tareas a realizar por ciclo

- **Búsqueda (F, *Fetch*)**
  - Se accede a memoria por la instrucción
  - Se incrementa el PC
- **Decodificación (D, *Decode*)**
  - Se decodifica la instrucción, obteniendo operación a realizar en la ruta de datos
  - Se accede al banco de registros por el/los operando/s (si es necesario)
  - Se calcula el valor del operando inmediato con extensión de signo (si hace falta)
- **Ejecución (X, *Execute*)**
  - Se ejecuta la operación en la ALU
- **Acceso a memoria (M, *Memory Access*)**
  - Si se requiere un acceso a memoria, se accede
- **Almacenamiento (W, *Writeback*)**
  - Si se requiere volcar un resultado a un registro, se accede al banco de registros

Conflictos en un cauce segmentado:

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

Copiar atascos raw, y etc…

Clase 5:Atascos y Soluciones a atascos

Soluciones a atascos estructurales: Duplicar recursos de hardware

Soluciones para riesgos RAW (dependencia de datos): Dos soluciones:

- Hardware: adelantamiento de operandos (Forwarding), consiste en pasar directamente el resultado obtenido a las instrucciones que lo necesitan como operando.
- Software: reordenar código, o instrucciones NOP.

Instrucciones de saltos / bifurcación / control de flujo:

Instrucciones de salto:

- Salto incondicional: La dirección de destino se debe determinar lo mas pronto posible dentro del cauce para reducir la penalización.
- Salto condicional: Introduce riesgo adicional por la dependencia entre la condición de salto y el resultado de una instrucción previa
  Técnicas para tratamientos de saltos:
  - Técnica de hardware: Predicción de saltos
    - Predecir que nunca se salta: Asume que el salto no se producirá. Siempre se capta la siguiente instrucción.
    - Predecir que siempre se salta: Asume que el salto se producirá. Siempre se capta la instrucción destino del salto.
  - Técnica de software: Salto retardado

Clase 6: RISC y CISC

RISC (Computadoras de repertorio reducido de instrucciones):

Características principales:
Esta arquitectura tiene un repertorio de instrucciones reducido. Se compone de instrucciones sencillas que se ejecutan en un ciclo de reloj. Posee un gran numero de registros. Las operaciones son de registro a registro. **Este tipo de procesadores hace énfasis en la segmentación de instrucciones, para ejecutar múltiples instrucciones en paralelo.**
Ventajas: Requiere hardware más simple. Tiene ciclos de instrucciones mas cortos.
Desventajas: Se necesitan más instrucciones para realizar tareas mas complejas.

CISC(Computadoras de repertorio amplio y complejo de instrucciones):

Características principales:

Esta arquitectura tiene un repertorio de instrucciones amplio. Se compone de instrucciones mas complejas, permite realizar operaciones mas avanzadas en una sola instrucción, cada instrucción puede requerir mas de un ciclo de reloj. Este tipo de arquitectura es usada en lenguajes de alto nivel, y **su enfoque se centra en la ejecución secuencial de instrucciones.**

Ventajas: Los programas tienden a ser mas cortos, ya que las instrucciones mas potentes permiten escribir programas con menos líneas de código.

Desventajas: Las instrucciones suelen tener ciclos de ejecución mas largos. Ademas, la decodificación y ejecución de estas instrucciones requieren un hardware mas costoso.

Clase 7: Memoria

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): Es un tipo de acceso a memoria en el que el procesador puede acceder a cualquier celda de memoria directamente, sin necesidad de recorrer secuencialmente las celdas de memoria previas. Este tipo de acceso es característico de la memoria volátil, como la RAM (memoria de acceso aleatorio), y es fundamental para el rendimiento de los sistemas informáticos modernos, ya que permite un acceso rápido y directo a los datos.

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. Cuenta con un método de acceso aleatorio.

Hay dos tipos de memoria RAM : SRAM y 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, pero con un costo de acceso mucho mayor al costo de acceso de una memoria principal (RAM).

La SRAM es una memoria aleatoria, rápida, es usada en caches y en sistemas donde la velocidad es importante.

La DRAM es una memoria aleatoria, es más lenta, más barata, y se utiliza en la memoria principal de las computadoras debido a su alta capacidad y alto costo.

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 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 de tiempo más largo. 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 ubicación de la memoria principal al mismo tiempo que la palabra en caché.

Para entender las políticas de escritura: Debemos entender que es un acierto y fallo de caché

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 más 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 tráfico 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. **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.

Clase 8: Procesadores:

Una unidad segmentada es una secuencia de etapas con la propiedad de que nuevas operaciones pueden iniciarse mientras otras se están procesando.

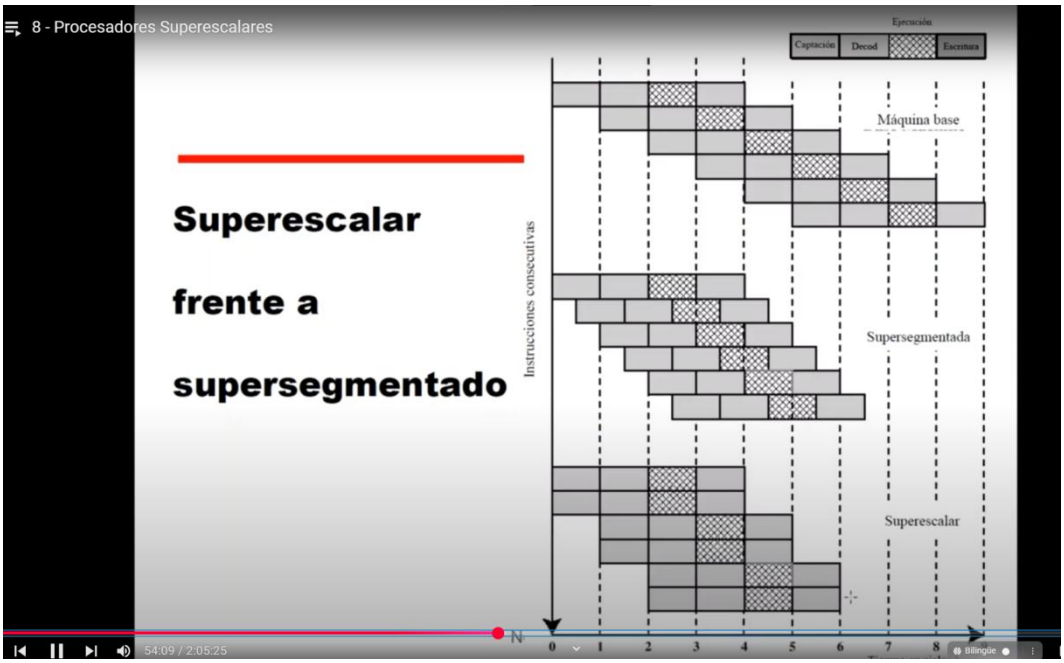
Procesadores supersegmentados:

Muchas operaciones no necesitan todo un ciclo de reloj. En los procesadores supersegmentados, se divide las etapas del cauce segmentado, en sub-etapas más pequeñas (y por lo tanto mas rápidas), además se transmiten los datos a la mayor velocidad del ciclo de reloj, por lo que, aumenta el grado de paralelismo, e incrementa la aceleración percibida.

Procesadores superescalares: El enfoque de este procesador, es llevar a cabo (completar) más de una instrucción de forma simultánea. Conlleva la duplicación de algunas o todas las partes de la CPU. Caracteristicas:

- Captar multiples instrucciones al mismo tiempo
- Ejecutar operaciones aritmeticas simultáneamente
- Ejecutar carga/almacenamiento, mientras se lleva a cabo alguna operación en ALU.

Aumenta el grado de paralelismo y la aceleración de la maquina, ya que se ejecutan mas instrucciones en paralelo.



Clase 9: Procesamiento paralelo anexo buses

Que es la segmentación de cauce?

La segmentación de cauce, es una técnica que está orientada a la optimización del uso del hardware de la CPU, permite realizar más de una tarea al mismo tiempo. El objetivo de utilizar la técnica de segmentación de cauce, es lograr una mayor cantidad de operaciones en el mismo tiempo. La segmentación de cauce, consiste en descomponer el proceso de ejecución de las instrucciones en fases o etapas que permitan una ejecución simultánea.

Esta técnica, explota el paralelismo entre las instrucciones de flujo secuencial.

Es una técnica propia por la arquitectura RISC.

Cual es el rendimiento de la segmentación de cauce?

El máximo rendimiento, es completar una instrucción cada ciclo de reloj.

El incremento potencial de la segmentación de cauce, es proporcional al número de etapas del cauce. Incrementa la productividad. Logramos la productividad por que voy a tener en un ciclo de reloj, varias etapas de diferentes instrucciones, pero no necesariamente reduce el tiempo de ejecución de la instrucción.

Si K es el numero de etapas del cauce => Vel. Procesador segmentado Vel. secuencial * K

Preguntas de finales:

Que es la segmentación de instrucciones ?

La segmentación de instrucciones es una técnica que mejora el rendimiento de los procesadores mediante el paralelismo en la ejecución de instrucciones. En la técnica de segmentación de instrucción, el ciclo de instrucción se divide en varias etapas que se ejecutan de forma paralela, cada etapa es independiente, de esta manera, no es necesario que se termine una instrucción para poder ejecutar la siguiente, sino que las instrucciones se van ejecutando a medida que se liberan unidades de CPU.

Esta técnica, podría verse limitada por la presencia de saltos y dependencias entre instrucciones.

Conclusión:

En un procesador sin segmentación: estas etapas se ejecutan secuencialmente.

Con 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.

Que es el 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

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 y de mecanismos para localizar instrucciones independientes.

Supersegmentacion

Es una técnica que aumenta el número de etapas en el pipeline. Con más etapas, puede haber más instrucciones en el pipeline al mismo tiempo, incrementando el paralelismo. La supersegmentacion divide las etapas del cauce en sub-etapas mas pequeñas y rapidas.

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 más 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.

Describir el funcionamiento de un sistema basado en microprocesador

Un sistema basado en microprocesador funciona ejecutando un programa almacenado en la memoria, que consiste en una secuencia de instrucciones. La CPU es la encargada de llevar a cabo estas instrucciones. El funcionamiento básico implica la repetición de un ciclo de captación y ejecución de instrucciones, mejor conocido como ciclo de instrucción.

Ciclo de instrucción:

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

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

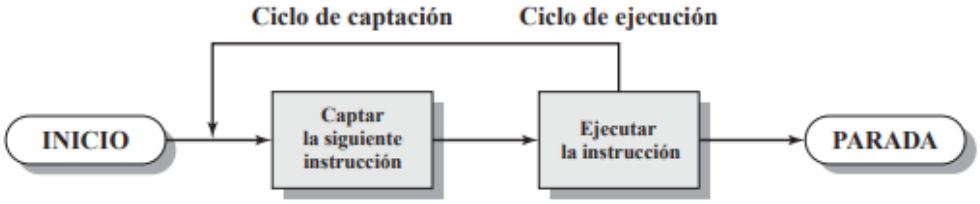


Figura 3.3. Ciclo de instrucción básico.

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).

Captación de instrucción: El procesador capta (lee) la siguiente instrucción desde la memoria. Esta instrucción se almacena en el registro de instrucción (IR).

Ejecución de la instrucción: El procesador lleva a cabo la operación especificada por la instrucción. Esta operación puede ser aritmética/lógica, de transferencia de datos, operaciones de control, operación de e/s, etc.

Además, en el ciclo de instrucción podemos encontrar la etapa de decodificación de la instrucción, que se encuentra entre la primer y segunda etapa, es decir, en el medio de ellas, luego de captar la siguiente instrucción y antes de ejecutarla.

Decodificación de la instrucción: El procesador interpreta el código de operación de la instrucción para determinar la operación a realizar.

Buses

Que es un bus, tipos de buses, aspectos claves para el diseño de buses.

Un bus, es un camino de comunicación entre dos o más dispositivos. Se trata de un medio de transmisión compartido. Al bus se conectan varios dispositivos y cualquier señal transmitida por uno de los dispositivos, estará disponible para cualquier dispositivo conectado al bus.
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.

Es importante tener un método de arbitraje, para controlar el permiso del bus a los dispositivos.

Existen varios **tipos de buses** dentro de una computadora

Bus de datos: Proporcionan un camino de comunicación para transmitir datos entre los módulos del sistema. Cuenta con una cantidad de líneas, conocidas como anchura del bus. Una línea puede transmitir de a un bit a la vez, por ende, el número de líneas determina cuantos bits se pueden transferir al mismo tiempo.

Bus de dirección: Se utilizan para designar fuente o destino del dato situado en el bus de datos.

Bus de control: Se utilizan para controlar el acceso y el uso de las líneas de datos y dirección.

Jerarquía de buses

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). Otros tipos de buses pueden conectar dispositivos de alta velocidad 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.

Si se conecta un gran número 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 más anchos)

Cuáles son los elementos/aspectos claves 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 líneas del bus: Pueden ser de dos tipos (Dedicadas o Multiplexadas)

Línea dedicada: Una línea de bus dedicada, está permanentemente asignada a una función o a un dispositivo en particular. Puede usarse para transferir datos, direcciones o señales de control, pero solo una función y dispositivo a la vez.

Línea multiplexada: Comparte varias funciones o dispositivos. Transmite diferentes tipos de 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.

Métodos de arbitraje

Son esenciales para controlar el acceso y el permiso al bus, generalmente en un bus se conectan varios dispositivos, y como un dispositivo puede transmitir con éxito en un momento dado, se necesita un método de arbitraje.

Hay dos tipos de arbitraje

Arbitraje centralizado: Existe un único dispositivo de hardware, llamado controlador o arbitro del bus, el cual es el responsable de asignar tiempo en el bus.

Arbitraje distribuido: No hay un controlador, sino que cada modulo dispone de la lógica necesaria para controlar el acceso al bus. Los módulos deben actuar conjuntamente para asignarse el acceso al bus.

Temporización

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

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.

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 afecta a las prestaciones del sistema. Cuanto más ancho es el bus de datos, mayor es el número de bits que se transmiten a la vez.

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

Proporciona un conjunto de funciones de uso general. **Utiliza temporización síncrona, un esquema de arbitraje centralizado, y utiliza líneas multiplexadas para compartir datos y direcciones en el mismo conjunto de líneas.**

Características del Bus PCI:

Alto ancho de banda: Ofrece altas tasas de transferencia de datos, lo que lo hace adecuado para subsistemas de e/s de alta velocidad, como por ejemplo adaptadores de pantalla grafica, controladores de interfaz de red.

Fácil implementación: Se implementa con pocos circuitos integrados. Está diseñado para ajustarse económicamente a los requisitos de e/s de los sistemas actuales.

Repertorio de instrucciones: Características y funciones

El repertorio de instrucciones, define las operaciones que el procesador puede llevar a cabo.

Los elementos esenciales de las instrucciones de las computadoras son:

Código de operación (codop): Especifica la operación que se va a realizar, por ejemplo operaciones aritméticas, lógicas, transferencia de datos entre registros, etc.

Referencia a operandos origen: Indica los operandos que sirven como entrada para la instrucción.

Referencia a operandos Destino: Específica donde se almacena el resultado de la operación. No todas las operaciones producen un resultado, pero aquellas operaciones que lo producen, deben indicar el destino.

Referencia a la siguiente instrucción: Indica al procesador donde encontrar la siguiente instrucción a ejecutar después de completar la instrucción actual. Esta referencia se encuentra en el PC (Program Counter)

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.

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:

- **Registro de datos:** Almacenan temporalmente los datos que se transfieren entre el dispositivo y el procesador.

- **Registro de control y estado:** 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.

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.

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.**

Procesadores

Procesador Superescalar: Se caracteriza por el uso de múltiples cauces de instrucciones independientes, cada cauce consta con múltiples 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.**

En los procesadores superescalares, **cada instrucción requiere al menos un ciclo de reloj para ejecutarse.**

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 más 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.

Resumen:

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. En la supersegmentacion, 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.

Conclusión 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.**

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): Es un tipo de arquitectura de procesamiento en paralelo. En un sistema SMP, Todos los procesadores tienen acceso equitativo a los recursos del sistema. Esto significa que los procesos pueden acceder y compartir datos entre sí de forma sencilla. Son especialmente adecuados para procesamiento de imágenes, análisis de datos, 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.

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.

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): Múltiples 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. Se encuentran las arquitecturas SMP Y NUMA

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. Por ejemplo los Clusters, abarcan esta categoría.

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

El uso de DMA, puede afectar la coherencia de datos en un sistema jerárquico 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 vector de interrupción?

El vector de interrupción, actúa como un puntero a la ubicación de memoria donde se encuentra la rutina de servicio de interrupción especifica para ese tipo de interrupción.

Cada tipo de interrupción tiene asignado un id que se usa para acceder a la tabla de vectores de interrupción. La tabla contiene 256 sectores de 32 bits.

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 interrupción, y luego el procesador salta a esa dirección de memoria y comienza a ejecutar la rutina de manejo de interrupción.

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.

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.**

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

(parecida a la pregunta anterior, pero ahora nos pide la finalización también.)

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.

Que son las bifurcaciones (SALTOS), y cuales existen?

(tomada en final de marzo 2024)

La instrucción de bifurcación, también llamada instrucción de salto, tiene como uno de sus operandos, la dirección de la siguiente instrucción a ejecutar.

Las instrucciones de salto más frecuentes son las de salto condicional, es decir, si se cumple una condición, se efectúa la bifurcación (o el salto), se actualiza el PC (program counter que contiene la dirección de la próxima instrucción a ejecutar), se actualiza

con la dirección especificada en el operando. En caso contrario de que no se ejecute la bifurcación, se ejecuta la siguiente instrucción.

Hay dos tipos de saltos:

Saltos condicionales: Cambia el flujo de ejecución solo si se cumple una condición especifica. En MIPS, las instrucciones de salto condicional, evalúan el valor de los registros y saltan a una dirección especifica solo si la condición es verdadera.

Saltos incondicionales: Este tipo de instrucciones, cambia siempre el flujo de ejecución de un programa a una dirección especifica, sin evaluar ninguna condición.

Describa las características que diferencian los SMP respecto a los CLUSTERS?

Los SMP se caracterizan por su arquitectura de memoria compartida, donde multiples procesadores comparten el mismo espacio de memoria y recursos brindando equidad en el acceso a los recursos sin jerarquía. La comunicación entre los procesadores es directa, a través de la memoria compartida.

Los clusters, se componen de nodos independientes interconectados mediante una red, cada uno con su propia memoria y recursos. La comunicación entre nodos se realiza via red, lo que puede generar latencia.

La principal ventaja de un SMP es que resulta mas fácil de gestionar y configurar que un cluster, ademas necesita menos espacio físico y consume menos energía.

Mencione las principales diferencias entre un bus PCI y SCSI.

Bus PCI (Peripheral Component Interconnect Interconexión de Componente Periférico): Es un bus de ancho de banda elevado, independiente del procesador, que se puede utilizar como bus de periféricos o bus para una arquitectura de entreplanta.

Bus SCSI (Small Computer System Interface - Pequeña interfaz del sistema de cómputo): Es una interfaz estándar para la transferencia de datos entre distintos dispositivos del bus de la computadora. Se utiliza para comunicar dispositivos rápidos, como discos CD-ROM, dispositivos de audio y dispositivos de almacenamiento externo de datos. Requiere un controlador de interfaz.

¿Qué características posee un procesador supersegmentado frente a un superescalar?

La supersegmentación aprovecha el hecho de que muchas etapas del cauce realizan tareas que requieren menos de medio ciclo de reloj. De este modo, se dobla la velocidad de reloj interna, lo que permite la realización de dos tareas en un ciclo de reloj externo. Por otro lado, un procesador superescalar puede ejecutar instrucciones en diferentes cauces de manera independiente y concurrente.

Características de los Cluster.

Se puede definir un Cluster como un grupo de computadores completos interconectados que trabajan conjuntamente como un único recurso de cómputo, creándose la ilusión de que se trata de una sola máquina. Cada computador del Cluster se denomina nodo. Sus principales características son:

- **Escalabilidad absoluta:** Es posible configurar Clusters grandes que incluso superan las prestaciones de los computadores independientes más potentes.

- **Escalabilidad incremental:** Un Cluster se configura de forma que sea posible añadir nuevos sistemas a él en ampliaciones sucesivas.

- **Alta disponibilidad:** Puesto que cada nodo es un computador autónomo, el fallo de uno de los nodos no significa la pérdida del servicio.

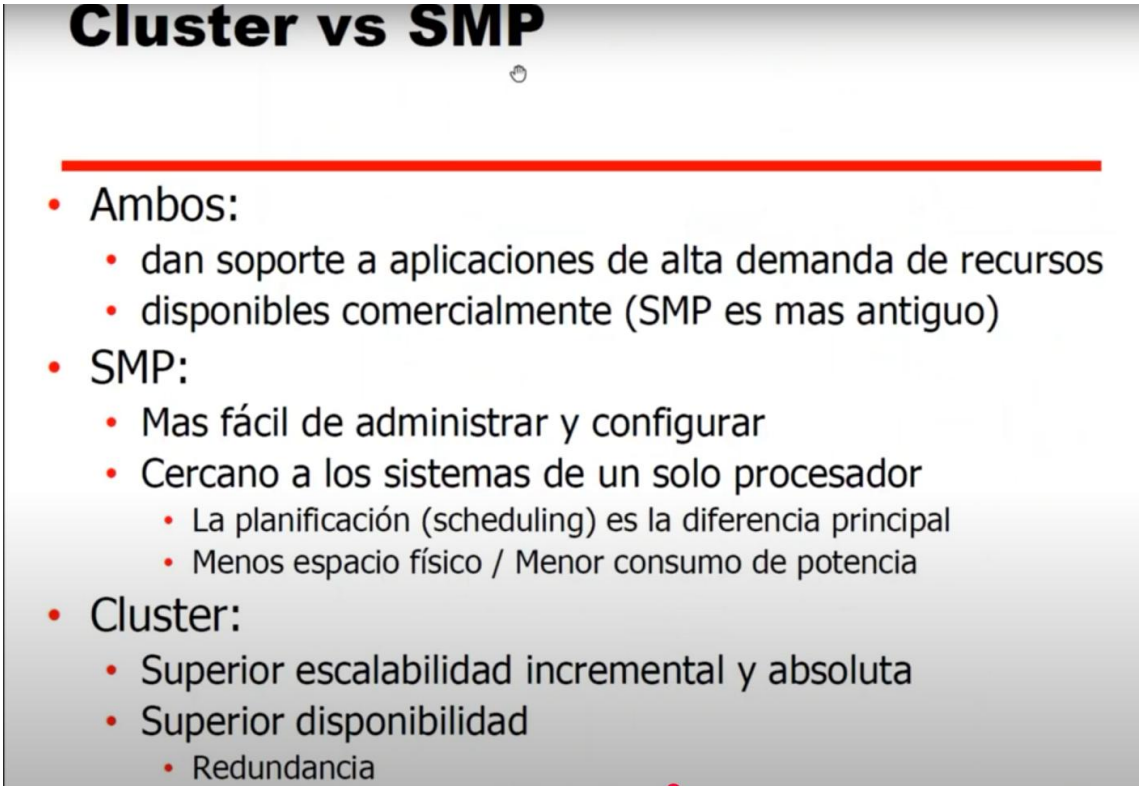
- **Mejor relación precio-prestaciones:** al utilizar elementos estandarizados, es posible configurar un Cluster con mayor o igual potencia de cómputo que un computador independiente mayor, a mucho menos costo.

MIMD MEMORIA COMPARTIDA:

MIMD MEMORIA DISTRIBUIDA

Problemas de un único bus:

Conectar un gran numero d



Términos UMA, NUMA, CC-NUMA

Todos los procesadores tienen acceso a toda la memoria
- Usan 'load' y 'store'

- UMA - Uniform memory access
  - Igual tiempo de acceso a todas las regiones de memoria
  - Igual tiempo de acceso a memoria para los diferentes procesadores
- NUMA - Nonuniform memory access
  - EL tiempo de acceso de un procesador difiere dependiendo de la región de memoria que accede
  - Diferentes procesadores acceden a diferentes regiones de memoria a diferentes velocidades
- CC-NUMA - cache coherente NUMA
  - Es un NUMA que mantiene coherencia de cache entre las cache de los distintos procesadores