

Arquitectura de computadoras

▼ clase 1 - repaso oc / pasaje de parámetros

▼ Modelo Von Neumann

El matemático John Von Neumann propuso la idea del programa-almacenado en memoria.

▼ Este modelo consta de 5 componentes principales

1. Unidad de entrada: provee las instrucciones y los datos
2. Unidad de memoria: donde se almacenan datos e instrucciones
3. Unidad aritmético-lógica: procesa los datos
4. Unidad de control: dirige la operación
5. Unidad de salida: se envían los resultados

▼ Aspectos más importantes

- Utilización del sistema binario
- Instrucciones y datos residen en memoria
- La memoria es direccionable por localidad sin importar el dato almacenado

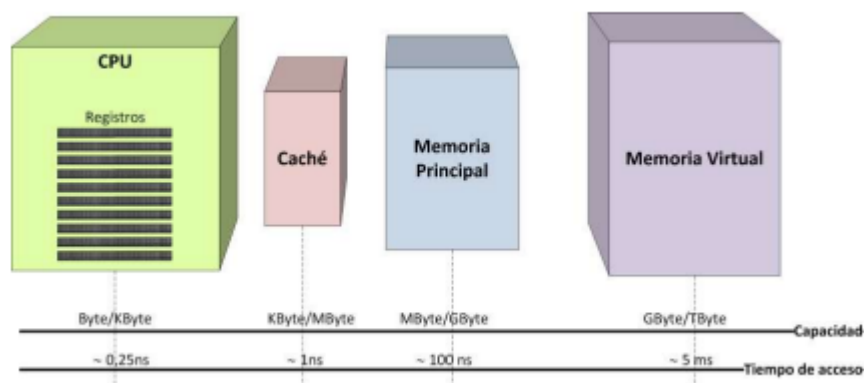
▼ Interconexión de un sistema de cómputo

Un sistema de cómputo está constituido por 3 subsistemas (CPU, Memoria, E/S)

▼ Se debe poder comunicar entre sí

Un bus es un medio de comunicación compartido, constituido por líneas (caminos de comunicación) capaces de transmitir señales binarias. Bus de datos, bus de direcciones, bus de control.

▼ Jerarquía de memoria



- Organizada en niveles.
- Niveles fabricados con tecnologías diferentes.
- A medida que nos alejamos de la CPU, cada nivel es más grande, lento y barato que el nivel anterior en la jerarquía.
- Propiedades a cumplir
 - Inclusión: los datos almacenados en un nivel deben estar almacenados en los niveles inferiores a él.
 - Coherencia: Las copias de la misma información en los distintos niveles deben contener los mismos valores.

▼ Por qué funciona la jerarquía?

El uso de la memoria caché (y la jerarquía de memoria en general) se sustenta en dos principios o propiedades que exhiben los programas:

- Principio de localidad espacial de referencia: cuando se accede a una palabra de memoria, es muy probable que

próximo acceso sea en la vecindad de la palabra anterior.

- Se sustenta en:
 - Ejecución secuencial del código.
 - Tendencia de los programadores a hacer próximas entre sí las variables relacionadas.
 - Acceso a estructuras tipo matriz o pila.
- Principio de localidad temporal de referencia: cuando se accede a una posición de memoria, es muy probable que en un lapso de tiempo corto, dicha posición de memoria sea accedida nuevamente.
 - Se sustenta en:
 - Formación de ciclos o bucles.
 - Subrutinas.
 - Pilas.

▼ Pasaje de parámetros a subrutinas

- El pasaje puede ser:
 - vía registros: el número de registros es la principal limitación. Es importante documentar los registros que se usan.
 - vía memoria: se usa un área definida en memoria (RAM). Difícil de estandarizar.
 - vía pila:
 - es el método más usado.
 - independiente de memoria y registros.
 - hay que comprender bien como funciona, ya que es usada por el usuario y por el sistema

▼ Pila

- Qué es?
 - Es una zona reservada de memoria con acceso LIFO (Last In, First Out).
- Funcionamiento

- Consta de dos operaciones básicas:
 - PUSH: Agregar un elemento en la parte superior de la pila.
 - POP: Desapilar el elemento en la parte superior de la pila.
- Se requiere un registro puntero de pila (SP)
 - contiene la dirección de la cabeza de la pila.
- Posibles pasos en un procedimiento
 1. Salvar el estado de BX.
 2. Salvar el estado de 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 parámetros. (opcional)
 8. Regresar correctamente del procedimiento.

▼ clase 2 - interrupciones

- Mecanismo por el cual se puede interrumpir el procesamiento normal de la CPU (ejecución secuencial de instrucciones en un programa).
- Pueden ser de origen interno o externo a la CPU.

▼ Por qué interrumpir?

1. Por un resultado de una ejecución de instrucción. (ej: overflow)
2. Por un temporizador interno del procesador.
3. Por una operación de entrada/salida.
4. Por un fallo de hardware. (ej. pérdida de energía)

▼ Qué hacer si interrumpen?

- Se transfiere el control a un programa llamado GESTOR que:
 - salve el estado del procesador.
 - responda a la causa que ocasionó la interrupción.
 - restaure el estado original del procesador.

- retorne a la ejecución normal del programa interrumpido.

▼ Jerarquía de interrupciones

- Si hay muchas fuentes que pueden solicitar interrupción, se establecen cuáles son las más importantes.
- Hay 2 tipos:
 - Enmascarables: pueden ser ignoradas.
 - No enmascarables: NO pueden ser ignoradas. Indican eventos peligrosos o de alta prioridad.

▼ Interrupciones por hardware

- Son las generadas por los dispositivos de E/S.
- Son las "verdaderas" interrupciones.
- No están relacionadas con el proceso en ejecución en ese momento.
- El sistema de cómputo tiene que manejar estos eventos externos "no planeados".
- Son conocidas como *interrupt request*.

▼ Traps

- Interrupciones por hardware creadas por el procesador en respuesta a eventos como:
 - Condiciones excepcionales: overflow en ALU de punto flotante.
 - Falla de programa: tratar de ejecutar una instrucción no definida.
 - Fallas de hardware: error de paridad de memoria.
 - Accesos a zonas de memoria protegidas.

▼ Interrupciones por software

- Es un llamado a una subrutina, solo que es por medio de una instrucción específica (no por medio de una señal externa como en las int. por hardware).
- Muchos procesadores tienen instrucciones que afectan al procesador de la misma manera que las interrupciones por

hardware.

- Son usadas para hacer llamadas a funciones del SO.
- No requieren conocer la dirección de la rutina en tiempo de ejecución.

▼ Ciclo de interrupción

- Se comprueba si se ha solicitado una interrupción
- Guarda su contexto
- Si no hay señal se capta la siguiente instrucción
- Si hay algún pedido de interrupción pendiente:
 - Se suspende la ejecución del programa en curso
 - Guarda su contexto
 - Carga el PC con la dirección de la rutina de gestión de interrupción (se inhiben otras interrupciones)
 - Retoma la ejecución del programa en el punto de interrupción

▼ Interrupciones múltiples

- Interrupciones inhabilitadas

Si ocurre una interrupción, se inhabilitan las demás y luego se gestiona la misma. El procesador puede y debe ignorar la señal de petición de interrupción en el momento de gestión. Por lo tanto las interrupciones tienen un orden secuencial estricto.

- Definir prioridades
 - Una interrupción de prioridad más alta puede interrumpir a un gestor de interrupción de prioridad menor.
 - Cuando se haya gestionado la interrupción de prioridad más alta, el procesador vuelve a las interrupciones previas.
 - Cuando todas las rutinas de gestión de interrupciones terminaron se retoma el programa del usuario.

▼ Reconocimiento de interrupciones

- Interrupciones multinivel

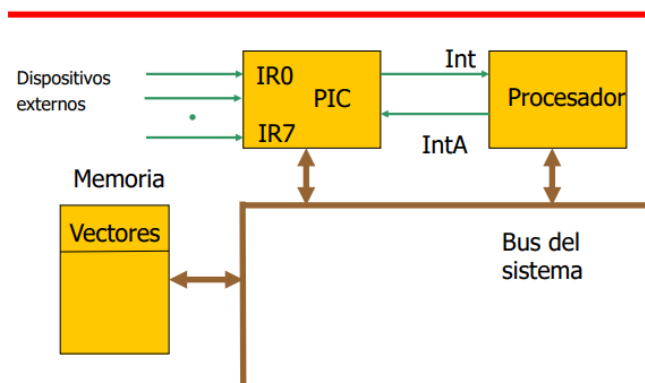
- Cada dispositivo que puede provocar una interrupción tiene una entrada física de interrupción conectada a la CPU (sencillo pero caro)
- Línea de interrupción única
 - Todos los dispositivos que pueden interrumpir conectados a una entrada.
 - Se debe "preguntar" a cada dispositivo si ha producido el pedido de interrupción (técnica polling)
- Interrupciones vectorizadas
 - El dispositivo que quiere interrumpir debe colocar en el bus de datos un vector.

▼ PIC

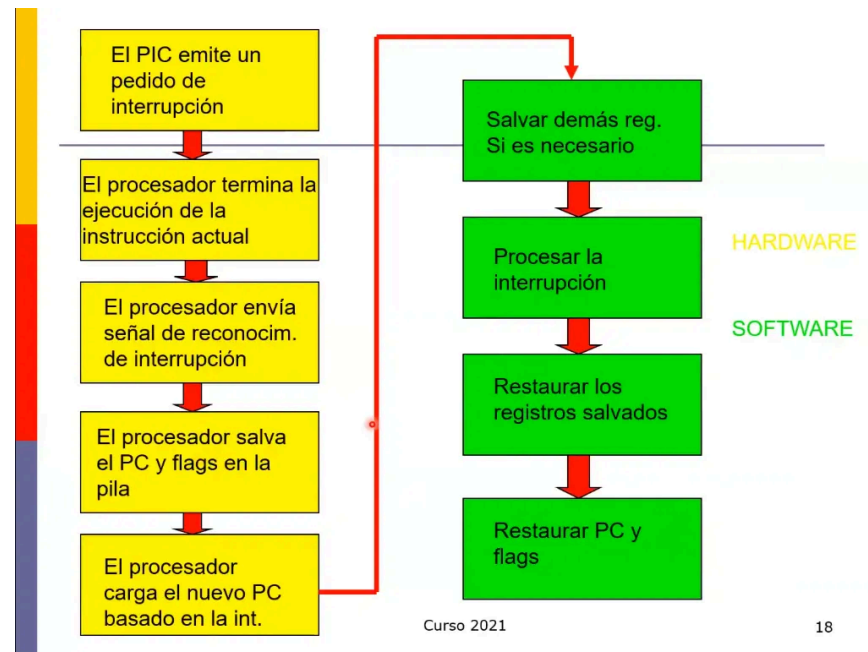
- Escenario de trabajo:
 - si el procesador tiene una única entrada de pedido de interrupciones.
 - si tenemos varios productores de interrupciones.
- ¿Cómo lo solucionamos?
 - Respuesta: con el PIC (Programmable Interrupt Controller)
- El PIC es un dispositivo que gestiona las señales de interrupción en un sistema de cómputo, determinando su prioridad y enviándolas al procesador para su ejecución en el momento adecuado.

▼ Conexionado

Conexionado



▼ Funcionamiento

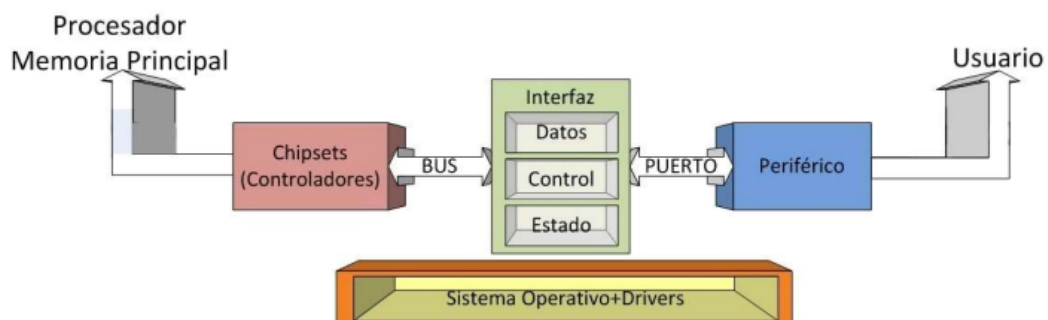


▼ clase 3 - entrada / salida

- El problema de Entrada/Salida es que hay varios periféricos con varios métodos de operación y diferentes transmisiones, velocidades, y formatos de datos.
- Todos estos métodos son más lentos que la CPU y la RAM
- Necesitan módulos de E/S.

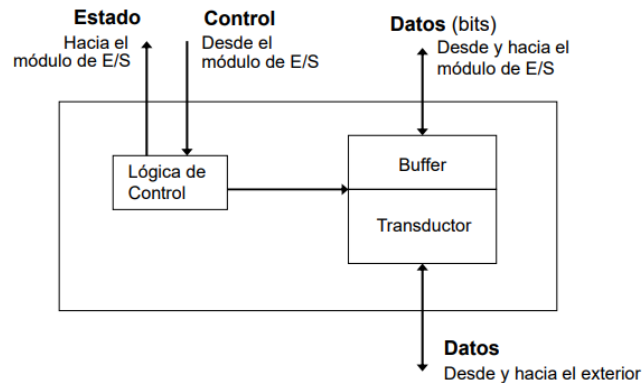
▼ Qué hace un módulo de E/S?

- Realiza la interfaz entre el procesador y la memoria con los periféricos.
- Pueden manejar uno o más periféricos.



▼ Dispositivo externo / periférico típico

Dispositivo externo tipo



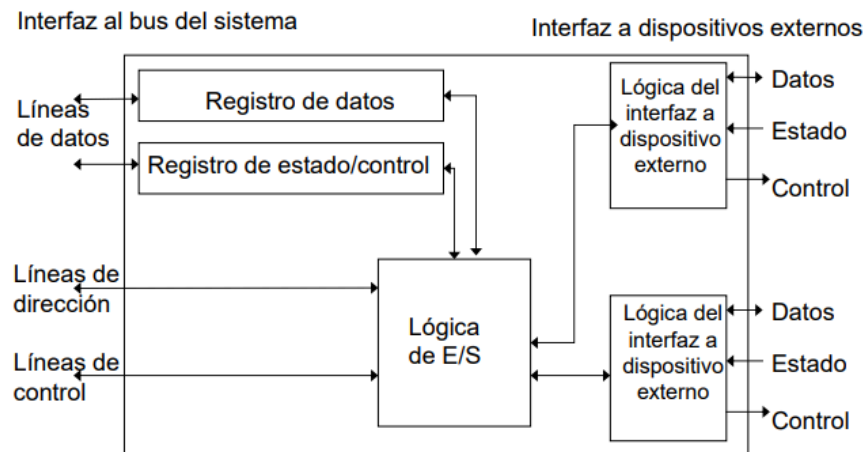
▼ Características de un puerto

- Realiza la interfaz entre el periférico y el módulo E/S
- Señales de control, estado y datos

▼ Funciones de un módulo de E/S

- Control y temporización de dispositivos externos (periférico).
- Interpretar órdenes de la CPU y transmitirlos al periférico.
- Comunicación con la CPU y memoria.
- Comunicación con el periférico.
- Controlar las transferencias de datos entre la CPU y el periférico.
- Informar a la CPU del estado del periférico.
- Almacenamiento temporal de datos.
- Detección de errores.

▼ DIAGRAMA MODULO E/S



▼ Operación de E/S

- Requiere:

- ▼ Direccionamiento

- E/S mapeada en memoria:
 - dispositivos de E/S y memoria comparten un único espacio de direcciones.
 - no hay órdenes específicas de E/S.
 - E/S aislada:
 - espacios de direcciones separados.
 - líneas especiales de E/S y memoria.
 - órdenes específicas de E/S.

- ▼ Transferencia de información

- Lectura o escritura

- ▼ Gestión de la transferencia

Mecanismos de sincronización y control de la transferencia de datos

- ▼ Técnicas de gestión de E/S

- ▼ E/S Programada con espera de respuesta

- La CPU tiene el control directo sobre la operación de E/S.
 - comprobación del estado del dispositivo

- envío de comandos de lectura/escritura
- transferencia de datos
- La CPU espera que el módulo de E/S termine la operación por lo tanto permanece ociosa durante un período de tiempo y eso no es deseable.
- ▼ E/S con interrupciones
 - La CPU no tiene que esperar que el módulo de E/S termine la operación.
- ▼ E/S con acceso directo a memoria (DMA)
 - ▼ Qué es el DMAC?
 - El controlador DMA es un dispositivo capaz de controlar una transferencia de datos entre un periférico y memoria sin intervención de la CPU.
 - El controlador DMA debe actuar como maestro del bus durante la transferencia DMA y debe ser capaz de:
 - Solicitar el uso del bus.
 - Especificar la dirección de memoria sobre la que se realiza la transferencia.
 - Generar las señales del control del bus.
 - ▼ Etapas de una transferencia DMA
 1. Inicialización de la transferencia: CPU envía los parámetros a la interfaz y al DMAC.
 2. Inicialización del interfaz: información sobre la transferencia y control.
 3. Inicialización DMAC: información sobre la transferencia y dirección inicial de memoria.
 4. Realización de la transferencia: el periférico le avisa al DMAC cuando está listo y el DMAC pide el control del bus y se realiza la transferencia.

5. Finalización de la transferencia: DMAC libera el bus y avisa al CPU la finalización de la transferencia.

▼ Problema que puede haber

- Puede empeorar el rendimiento de la CPU si el DMAC hace uso intensivo del bus.
- El problema se reduce con el uso de la memoria caché.
 - La mayor parte del tiempo el CPU lee de la memoria caché por lo tanto no usa mucho el bus de memoria.
 - El DMAC aprovecha estos intervalos para realizar las transferencias.
- En casos de computadoras sin caché
 - la CPU no utiliza el bus en todas las fases de ejecución así que el DMAC las aprovecha para las transferencias.

▼ Tipos de transferencias

▼ DMA por ráfaga (burst)

- El DMAC solicita el control del bus a la CPU.
- Cuando la CPU libera el bus, el DMAC no libera el bus hasta realizar la transferencia completa.
- Ventajas: la transferencia se realiza de forma rápida.
- Desventajas: puede empeorar el rendimiento del sistema.

▼ DMA por robo de ciclo (cycle-stealing)

- El DMAC solicita el control del bus a la CPU.
- La CPU libera el bus y el DMAC transfiere de a una palabra y se lo devuelve a la CPU.

- Esto se repite cuanto sea necesario hasta que termine la transferencia.
- Ventajas: no empeora el rendimiento del sistema
- Desventajas: la transferencia lleva más tiempo.

▼ Canales de E/S

- Representan una extensión al concepto DMA.
- Completo control de la transferencia de datos.
- Programa almacenado en memoria principal.
- La CPU inicia la transferencia de E/S.
- Tipos:
 - Selector:
 - Controla varios dispositivos de alta velocidad y uno a la vez.
 - Multiplexor:
 - Puede manejar E/S con varios dispositivos a la vez.

▼ clase 4 - memoria

▼ Memoria caché

- Cantidad pequeña de memoria rápida.
- Se ubica entre la memoria principal y la CPU.
- Puede estar en un chip o en módulo CPU.

▼ Funcionamiento de la caché.

1. La CPU solicita contenido de 1 dirección de memoria.
2. La cache ¿tiene ese dato?
 - a. Si es así, lo obtiene de la caché.
 - b. Si no es así, se lee el bloque que contiene esa dirección desde la memoria principal y copia en la caché.
3. La caché le entrega el dato a la CPU.

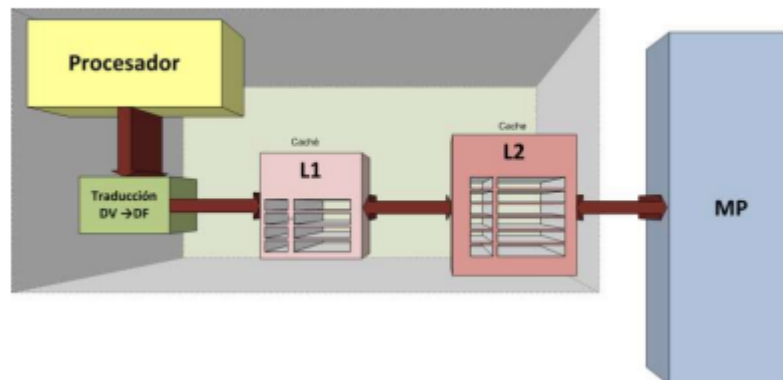
▼ Conceptos básicos

- Acierto: se encuentra en la CPU el dato solicitado.
- Fallo: no se encuentra en la caché el dato solicitado.

▼ Diseño de la caché

▼ Organización (tamaño y cantidad)

• Tamaño - Costo - Niveles.



- Tiene dos niveles L1 y L2. L1 es una memoria caché que se encuentra interna al procesador y trabaja a su misma frecuencia (por eso la rapidez de acceso) y la L2 que es ligeramente más lenta que la L1 pero tiene mayor capacidad.

▼ Política de ubicación

- Hay distintos tipos de función de correspondencia: directa, totalmente asociativa y asociativa por conjuntos.

▼ Ubicación de un bloque

- correspondencia directa: un bloque solo puede estar almacenado en un lugar de la caché.
- correspondencia totalmente asociativa: un bloque solo puede almacenarse en cualquier lugar de la caché.
- correspondencia asociativa por conjuntos: un bloque puede almacenarse en un conjunto restringido de lugares en la caché.
 - conjunto = grupo de líneas.

▼ Ventajas y desventajas

- correspondencia directa
 - ventajas
 - simple
 - poco costosa
 - desventajas
 - hay una posición concreta para cada bloque dado, lo que puede generar muchos desajustes.
- correspondencia totalmente asociativa:
 - ventajas
 - un bloque de MP puede colocarse en cualquier línea de la caché.
 - la etiqueta identifica sin equivocarse un bloque de memoria.
 - desventajas
 - se busca en TODAS las etiquetas = búsqueda costosa.
- correspondencia asociativa por conjuntos
 - combina lo mejor de las otras correspondencias.
 - la caché se divide en un grupo de conjuntos.
 - un bloque determinado corresponderá a cualquier línea de un conjunto determinado.

▼ Política de reemplazo o sustitución

- Algoritmos de sustitución
 - En correspondencia directa:
 - el que ocupa el lugar del nuevo
 - En correspondencias asociativas:
 - LRU (menos recientemente usado)
 - FIFO (más antiguo)
 - LFU (menos frecuentemente usado)

- Aleatoria

▼ Política de escritura

- En acierto:
 - Write-through:
 - se actualizan simultáneamente la posición de la caché y de la MP.
 - Write-back:
 - La información solo se actualiza en la caché.
- En fallo
 - Write allocate:
 - La información se lleva de la MP a la caché. Se sobrescribe en la caché. (habitual con write-back)
 - No write allocate:
 - El bloque no se lleva a la memoria caché. Se escribe directamente en la MP. (habitual con write-through)

▼ Buses

▼ Qué es un bus?

- Es un medio de comunicación compartido.
- Constituido por líneas (camino de comunicación) capaces de transmitir señales binarias.
- Interconecta a CPU, MEMORIA y E/S.

▼ Bus de datos

- Transporta datos e instrucciones.
- Número de líneas = número de bits que se pueden transferir al mismo tiempo (8,16,32,64)

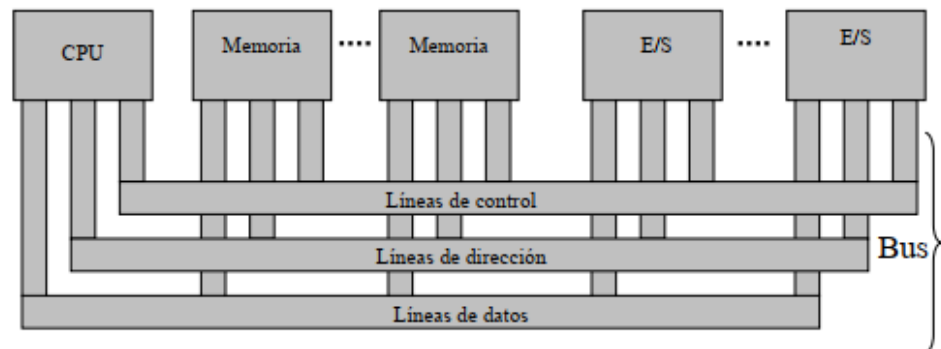
▼ Bus de direcciones

- Identifica el origen o el destino de los datos
- El ancho del bus determina la capacidad máxima de memoria del sistema

▼ Bus de control

- Información de control y temporizado
 - Señales de lectura/escritura de memoria y E/S
 - Señales de selección o habilitación
 - Señales de reloj (clock)
 - Señales de pedido de interrupción

Interconexión mediante un bus



▼ Problemas de un único bus

- Conectar muchos dispositivos a un bus produce retardos de propagación.
- Se utilizan varios buses para solucionar estos problemas.
 - jerarquía de buses.

▼ Tipos de buses

- Dedicados
 - uso de líneas separadas para direcciones y datos.
- Multiplexados
 - uso de las mismas líneas para direcciones y datos.

▼ Arbitraje del bus

- El control del bus puede necesitar más de un módulo.
 - ej CPU y DMA
- Sólo una unidad puede transmitir a través del bus en un instante dado.
- Dos métodos de arbitraje
 - centralizado
 - un único dispositivo hardware es responsable de asignar tiempos en el bus.
 - distribuido
 - cada módulo puede controlar el acceso al bus.
 - cada módulo dispone de lógica para controlar el acceso

▼ Temporización

- Forma de coordinar los eventos en un bus.
- sincrónica
 - utiliza un reloj (clock).
 - la presencia de un evento está determinada por este.

▼ clase 5 - segmentación de cauce

▼ Qué es?

- La segmentación de cauce (pipelining) es una forma efectiva de organizar el hardware de la CPU para realizar más de una operación al mismo tiempo.
- Descompone el proceso de ejecución de las instrucciones en etapas que permitan una ejecución simultánea.
- Explota el paralelismo entre instrucciones.
- La segmentación es invisible al programador.

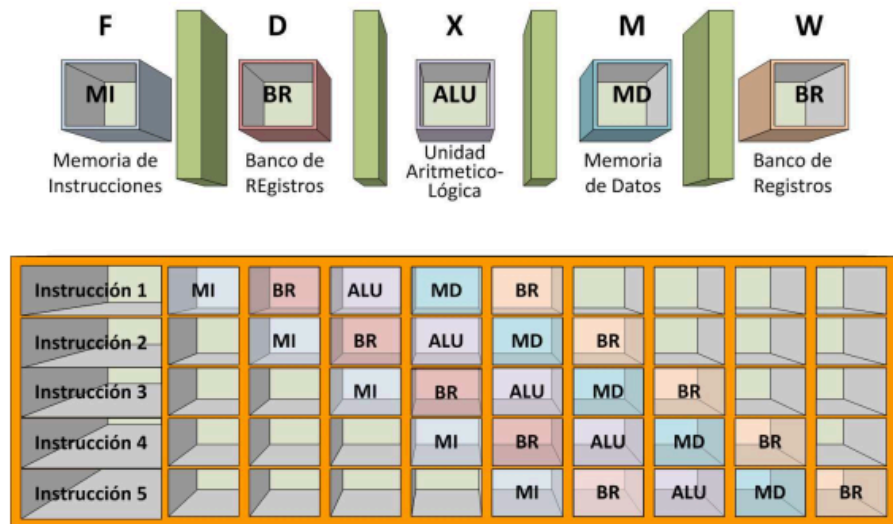
▼ Tareas a realizar por ciclo

1. Búsqueda
2. Decodificación
3. Ejecución

4. Acceso a memoria

5. Almacenamiento

Ejemplo de segmentación



▼ Análisis de la segmentación

- Suposiciones

- Todas las tareas duran el mismo tiempo.
- Las instrucciones siempre pasan por todas las etapas.
- Todas las etapas pueden ser manejadas en paralelo.

- Problemas

- No todas las instrucciones necesitan todas las etapas.
- No todas las etapas pueden ser manejadas en paralelo.
 - F y M acceden a memoria.
- No se tienen en cuenta los saltos de control.

▼ Atascos de un cauce (stall)

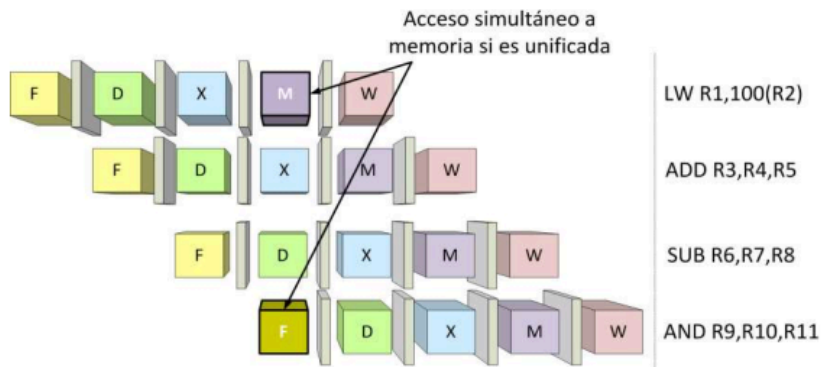
- Situaciones que impiden a la siguiente instrucción que se ejecute en el ciclo que le corresponde.

▼ Estructurales

- Provocados por conflictos por los recursos.

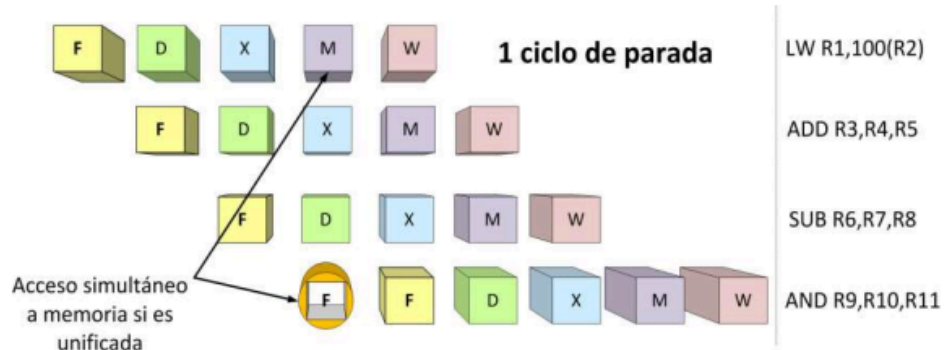
Riesgos estructurales

Dos o mas instrucciones necesitan utilizar el mismo recurso hardware en el mismo ciclo.



Riesgos estructurales (2)

Resolución ante el riesgo:



▼ Por dependencia de datos

- Situación en la que los operandos fuente o destino de una instrucción no están disponibles en el momento en que se necesitan en una etapa determinada del cauce.
- Tipos de dependencia de datos
 - Lectura después de escritura (RAW)
 - una instrucción genera un dato que lee otra posterior.
 - Escritura después de escritura (WAW)

- una instrucción escribe un dato después que otra posterior.
 - Escritura después de lectura (WAR)
 - una instrucción modifica un valor antes de que la otra lo tiene que leer, lo lea.
- ▼ Por dependencia de control
 - Ocurren cuando la ejecución de una instrucción depende de cómo se ejecute otra (especialmente en instrucciones de salto)
- ▼ clase 6 - soluciones a atascos
 - ▼ soluciones a riesgos estructurales
 - Duplicación de recursos hardware.
 - ej., más sumadores/restadores además de la ALU
 - Separación en memorias de instrucciones y datos.
 - Turnar el acceso al banco de registros.
 - escrituras en la primera mitad del ciclo de reloj, lecturas en la segunda.
 - ▼ soluciones a riesgos de datos
 - Técnica hardware
 - Adelantamiento de operandos (forwarding), pasando directamente los resultados a las instrucciones siguientes sin esperar escritura en memoria.
 - Técnica software
 - Instrucciones NOP.
 - Reordenación de código para maximizar la separación de instrucciones con dependencia de datos.
 - ▼ soluciones a riesgos de control
 - Los tipos de instrucciones de salto que generan riesgos de control son:
 - Salto incondicional: Se debe determinar la dirección de destino lo más rápido posible.

- Salto condicionales: Introducen un riesgo adicional porque dependen del resultado de una instrucción previa.
- Algunas soluciones:
 1. Modificaciones en la ruta de datos: adelantar la resolución de los saltos a la etapa de decodificación para minimizar paradas.
 2. Técnicas hardware
 - a. Técnicas estáticas: Asumir que el salto siempre o nunca se tomará.
 - b. Técnicas dinámicas: Usar un conmutador saltar/no saltar basado en la historia de ejecución (ej., tabla de historia de saltos - BTB)
 3. Técnica software: salto retardado, el programador introduce instrucciones que se ejecutarán en cualquier caso después de la instrucción de salto.
 4. Otras:
 - a. Flujos múltiples: ejecutar varias opciones en paralelo
 - b. precaptación del destino de salto
 - c. uso de buffer de bucles para mejorar eficiencia en iteraciones pequeñas.

▼ clase 7 - RISC

▼ Nota de la clase

En esta clase se compara las arquitecturas de procesador RISC vs CISC. Igualmente no existe una clara "barrera diferenciadora" ya que no existe un par de máquinas RISC y CISC directamente comparables.

"La mayoría de máquinas son una mezcla de ambas."

- RISC: Computadoras de repertorio reducido de instrucciones

▼ Características principales RISC

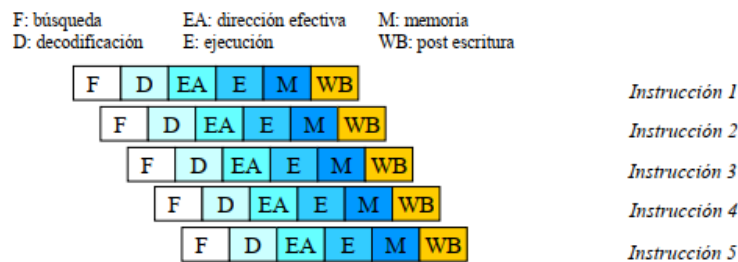
- Gran número de registros de uso general.
- Repertorio de instrucciones limitado y sencillo.
- Pone énfasis en la optimización de la segmentación de instrucciones.

- Una instrucción por ciclo de reloj.
- Operaciones registro a registro.
- MDD sencillos.
- Formatos de instrucción fijos y sencillos.
- Mayor tiempo de compilación
- CISC: Computadoras de repertorio complejo de instrucciones.
 - ▼ Características principales CISC
 - Menos registros, más operaciones en memoria.
 - Repertorio de instrucciones variado y complejo.
 - Ejecución en múltiples ciclos de reloj.
 - MDD avanzados.
 - Menos dependientes de compiladores optimizados.
 - Programas más pequeños
 - El programa ocupa menos memoria, pero la memoria hoy en día es muy barata.

▼ clase 8 - superescalares

- ▼ Procesadores supersegmentados
 - Divide el ciclo de reloj en sub-intervalos.
 - Mayor frecuencia del ciclo de reloj.
 - Se transmiten los datos a la mayor velocidad del ciclo de reloj.
 - El tiempo para las instrucciones individuales no varía.
 - Aumenta el grado de paralelismo.
 - Incrementa la aceleración percibida.

Procesador supersegmentado

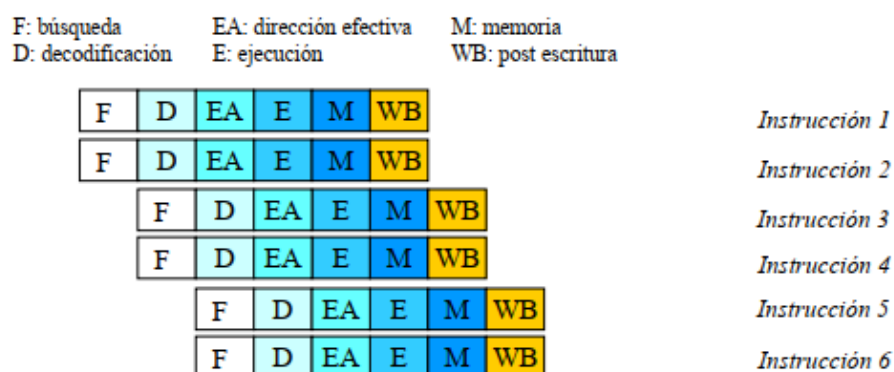


Ejecución super-segmentada de instrucciones

▼ Procesadores superescalares

- Se puede completar más de una instrucción simultáneamente.
 - Captar instrucciones al mismo tiempo.
 - Ejecutar sumas y multiplicaciones simultáneamente.
 - Ejecutar una store mientras se completa una operación en la ALU.
- Por esto aumenta el grado de paralelismo.

Enfoque Superescalar (2)



Ejemplos: MC68040, i80486, MC88110, i80860, PA-RISC, Sparc, R6000 ...

▼ Paralelismo

▼ De instrucciones

existe cuando instrucciones de una secuencia son independientes y pueden ejecutarse en paralelo

- limitaciones
 - Dependencia de datos verdadera.
 - Dependencia relativa al procedimiento.
 - Conflictos en los recursos.
 - Dependencia de salida.
 - Antidependencia.

▼ De la máquina

es una medida de la capacidad del procesador para aprovechar paralelismo de instrucciones

- Depende de:
 - núm. de instrucciones captadas por ciclo
 - núm. de unidades funcionales
 - mecanismos para localizar instrucciones independientes

▼ Sobre las instrucciones

- En la localización de instrucciones independientes, hay tres ordenaciones importantes:
 1. orden en que se captan las instrucciones.
 2. orden en que se ejecutan las instrucciones.
 3. orden en que las instrucciones actualizan los registros y posiciones de memoria.
- Políticas de emisión de instrucciones de procesadores superescalares:
 - Emisión en orden y finalización en orden: Se emiten las instrucciones en el orden exacto en que lo haría una ejecución secuencial, y se escriben los resultado en ese mismo orden.
 - Emisión en orden y finalización desordenada: Las instrucciones se pueden completar en cualquier orden una vez que han sido ejecutadas, siempre y cuando no altere el resultado final. La

emisión se para cuando hay cualquier tipo de conflicto o dependencia de datos.

- Emisión desordenada y finalización desordenada: Las instrucciones se emiten a la etapa de ejecución en el orden en que están listas para ser ejecutadas, y se pueden completar en cualquier orden, siempre y cuando no altere el resultado final.

▼ Renombre de registros

- Las dependencias de salida y antidependencias surgen por conflictos de almacenamiento.
 - Los registros no pueden reflejar la secuencia de valores dictada por el flujo del programa.
- Esto puede detener alguna etapa del cauce.
- Renombramiento de registros
 - Cuando la ejecución de una instrucción guarda un resultado en registro, se almacena en un registro nuevo.

▼ clase 9 - procesamiento paralelo

▼ Introducción al procesamiento paralelo

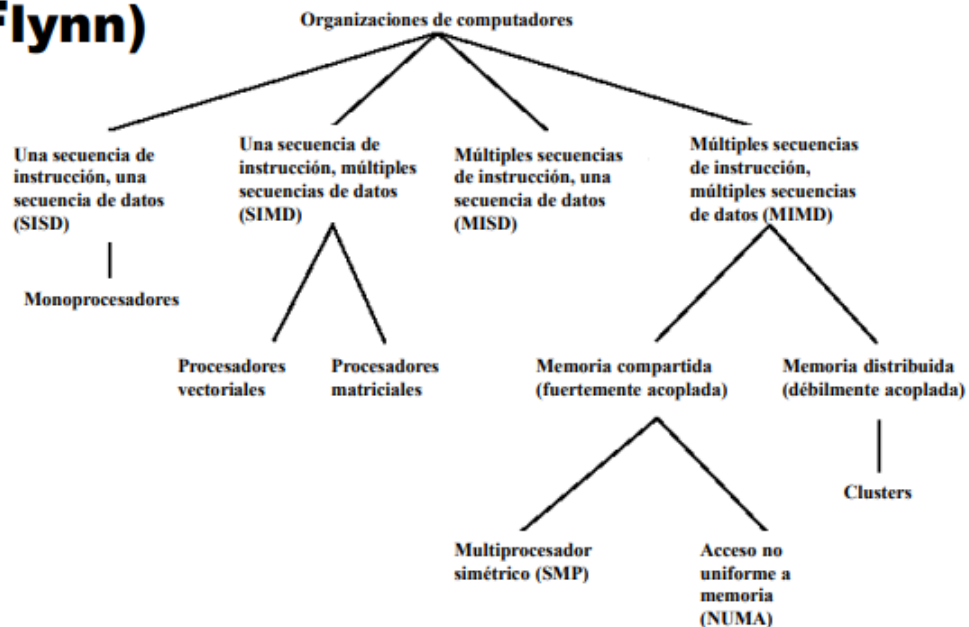
- La demanda de máquinas de mayor rendimiento seguirá existiendo.
- Dos opciones
 - Mejorar el rendimiento con un solo procesador.
 - Paralelismo a nivel instrucción.
 - Arquitecturas de sistemas con varios procesadores.
 - Paralelismo a nivel proceso.

▼ Taxonomía de Flynn

- Categoría de las computadoras
 - SISD: una secuencia de instrucciones y una secuencia de datos.
 - SIMD: una secuencia de instrucciones y muchas secuencias de datos.
 - MISD: muchas secuencias de instrucciones y una secuencia de datos.

- MIMD: muchas secuencias de instrucciones y muchas secuencias de datos.

Taxonomía de las arquitecturas (Flynn)



▼ SISD

- Un único procesador interpreta una única secuencia de instrucciones.
- Datos almacenados en una única memoria.
- Computadoras monoprocesador.

▼ SIMD

- Una única instrucción máquina controla paso a paso la ejecución simultánea de cierto número de elementos de proceso.
- Cada elemento de proceso tiene una memoria dedicada.
- Cada instrucción es ejecutada por cada procesador.
- Procesadores vectoriales y matriciales
 - Computadora con una única unidad de control y una matriz de elementos "computacionales"

▼ MISD

Nunca implementada

▼ MIMD

- Un conjunto de procesadores ejecuta secuencias de instrucciones diferentes en simultáneo.
- Conjuntos de datos diferentes.
- Se dividen según la forma de comunicarse

▼ Memoria compartida

▼ SMP

- Computadora autónoma.

▼ Características

- Tiene dos o más procesadores que:
 - comparten la MP y la E/S.
 - están interconectados.
 - tienen tiempo de acceso a memoria similar.
 - pueden desempeñar las mismas funciones.
- SO integrado:
 - proporciona interacción entre los procesadores y sus programas.

▼ Ventajas

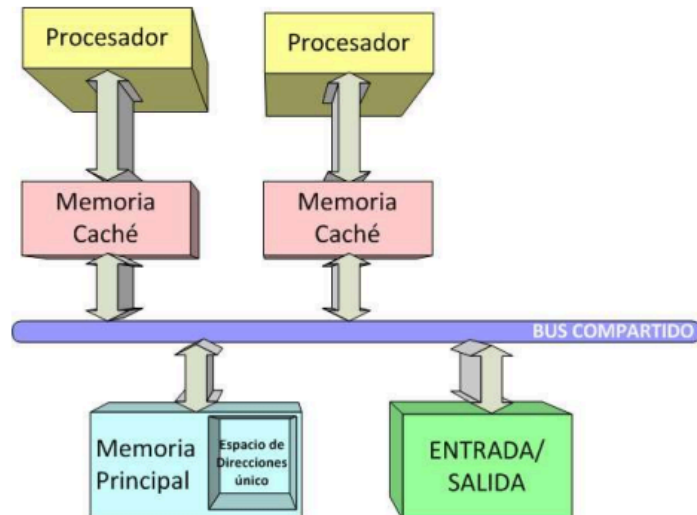
- mayores prestaciones (si se puede trabajar en paralelo)
- buena disponibilidad (un fallo en un procesador no detiene la pc)
- crecimiento incremental (se pueden agregar más procesadores)

▼ Desventajas

- bus compartido
 - la prestación está limitada por el tiempo de ciclo de bus.

- puede haber problemas con la caché.

Arquitectura de un SMP



▼ NUMA

- El Non-Uniform Memory Access (NUMA) es una arquitectura de memoria en sistemas multiprocesador donde el tiempo de acceso a la memoria varía según la ubicación del procesador que la solicita. En NUMA, cada procesador tiene su propia memoria local, a la que accede más rápido, y puede acceder a la memoria de otros procesadores, pero con mayor latencia.

▼ Memoria distribuida (CLUSTERS)

▼ Características

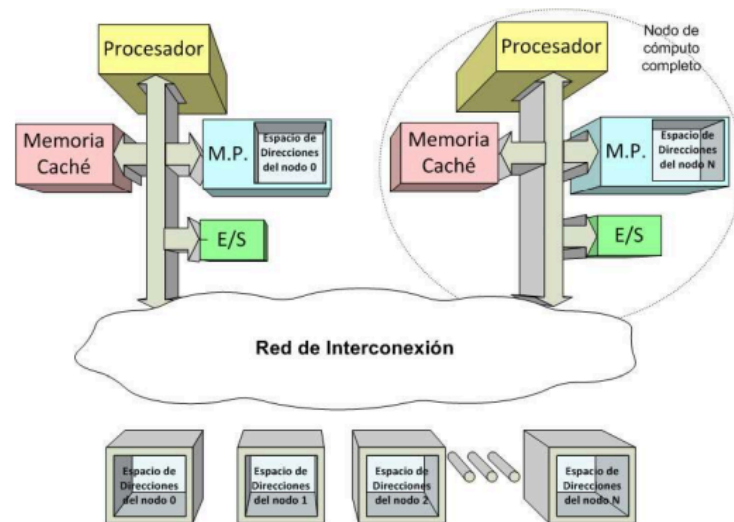
- Computadoras completas interconectadas que trabajan conjuntamente como un único recurso.
- Cada computadora se denomina "nodo".
- Aplicaciones propias de un servidor.
- Prestaciones y disponibilidad elevada.

▼ Beneficios

- Escalabilidad absoluta.
- Escalabilidad incremental.

- Alta disponibilidad.
- Mejor relación precio/prestaciones.

Arquitectura de un cluster



▼ SMP vs CLUSTER

- SMP
 - más fácil de administrar y configurar.
 - cercano a los sistemas de un solo procesador.
- CLUSTER
 - superior escalabilidad incremental y absoluta.
 - superior disponibilidad.