

1)

A)

Por valor (se pasa el valor del parámetro)

Por referencia (se pasa la dirección del parámetro)

Por registro (se utilizan registros para almacenar parámetros y luego utilizarlos en subrutinas)

Por pila (se apilan parámetros para luego utilizarlos mediante push y pop en las subrutinas) (es un espacio de memoria reservado)

B)

Una subrutina es una porción de código que se llama y retorna explícitamente (no concurrentemente a diferencia de las interrupciones) y se ejecuta secuencialmente dentro del programa principal, un gestor de interrupción es una porción de código que se ejecuta en respuesta a una interrupción y puede ocurrir de manera asíncrona con respecto al flujo principal del programa, por lo tanto el flujo del programa principal se reanuda desde donde se suspendió debido a la interrupción.

2)

A)

El PIC puede manejar 8 peticiones de interrupción independientes, numeradas de 0 a 7 (INT0 a INT7). El número de interrupción es la prioridad, teniendo prioridad los números más bajos.

El PIC tiene conectados los siguientes dispositivos a sus líneas de petición de interrupción:

INT0 Conectado a la tecla F-10.

INT1 TIMER, Contador de Eventos.

INT2 HANDSHAKE.

INT3 Controlador de DMA.

Memoria E/S	
20h	EOI
21h	IMR
22h	IRR
23h	ISR
24h	INT0
	INT1
	INT2
	INT3
	INT4
	INT5
	INT6
	INT7

Todos los registros están compuestos por 8 bits.

En el IMR cada bit representa una interrupción (del 0 al 7), este registro permite enmascarar selectivamente las interrupciones que va a recibir el PIC. Si el bit N es puesto en 1, la interrupción N estará deshabilitada.

El EOI se encarga de avisar al PIC que la interrupción ya fue atendida.

El IRR es el registro de peticiones de interrupciones.

El ISR es el registro de interrupción de servicios. (El bit que este en 1 será la interrupción que se esté ejecutando actualmente).

En los INT 0..7 se almacena la dirección de la subrutina a ejecutar en caso de una interrupción... El valor que se carga se multiplica por 4 y el resultado debería ser la dirección a la subrutina.

B)

En este caso la CPU mantiene el control total de la operación de E/S que hay que realizar. Ordena al módulo y dispositivo que debe ejecutarse y que hacer a través de una orden. Periódicamente, la CPU pregunta por los bits de estado del módulo que indican si la E/S finalizó, ya que este no tiene la capacidad de notificar cuando terminó. Mientras tanto la CPU se mantiene ociosa a la espera, lo cual resulta ineficiente.

3)

A)

Aspectos que se tienen en cuenta:

- La organización (tamaño, costos y niveles de caché)
- Políticas de ubicación
- Políticas de reemplazo
- Políticas de escritura

B)

A pesar de que el DMA trabaja independientemente del CPU, esto podría generar inconsistencias en cuanto a los datos almacenados en los niveles de jerarquía de memoria cuando múltiples dispositivos acceden a modificar esos datos, y esto es una posibilidad, ya que el DMA accede directamente a la memoria sin consultar al CPU.

4)

A)

Depende de si las instrucciones son independientes entre sí, o sea, que no tengan dependencia de datos o control entre sí.

Depende de la cantidad de unidades de ejecución, ALU, unidades de salto, etc que tenga el hardware.

Depende del ancho del cauce, cuanto más grande sea, más instrucciones pueden ejecutarse simultáneamente.

Depende de las técnicas de predicción y especulación que tenga para evitar un “cuello de botella” y mantener un flujo constante.

Puede ayudar a la reorganización de instrucciones para evitar dependencia de datos.

B)

El objetivo de la técnica de renombrado de registros en una máquina superescalar es evitar las dependencias de datos, asignando un registro físico a cada operando en una instrucción en el momento en que se decodifica. De esta manera, cada operando en una instrucción puede ser renombrado con un registro físico diferente, lo que permite que múltiples instrucciones que tienen los mismos registros de destino (y que, por lo tanto, podrían causar dependencias de datos) se ejecuten simultáneamente y de forma especulativa.

5)

A)

Elementos que caracterizan un bus son:

Ancho del bus.

Velocidad del bus.

Tipo de bus (interno o externo).

El protocolo de comunicación.

La función del bus (transferencia de datos/transferencia de direcciones, transferencia de control).

B)

Conjunto de procesadores que ejecutan conjuntos de instrucciones diferentes y utilizan conjuntos de datos diferentes por procesador, con acceso o a una memoria compartida o distribuida.

Se pueden sub-clasificar por las formas de comunicación y gestión de memoria:

Memoria Compartida: Multiprocesadores Simétricos SMP y sistemas NUMA

- Multiprocesadores Simétricos (SMP):

Conjunto de procesadores físicos que **comparten una sola memoria y tienen conexión a través de un mismo bus de sistema.** Además, manejan un tiempo de acceso a memoria similar para todos los procesadores, a lo que se define como una arquitectura de memoria UMA.

Sus ventajas son: Mayores prestaciones en lo que es la organización para poder realizar tareas de forma paralela. **Una falla en alguno de los procesadores no detendrá la ejecución.** Escalable en la integración de nuevos procesadores con las mismas propiedades para poder contar con más unidades.

Sus desventajas se pueden ver del lado del bus de uso compartido. Si bien todos los procesadores tienen acceso a una misma memoria física, **al momento que se requiera un acceso múltiple entre varios procesadores, se puede ver afectado el rendimiento por el acceso al bus que comparten y esto se puede seguir prolongando si la cantidad de unidades aumenta.** Como posible solución ya se implementan métodos para que cada procesador pueda estar equipado con un caché jerárquico, para que así se reduzcan la cantidad de acceso y reduzca así también la latencia, mejorando así el rendimiento.

Si bien el uso compartido de recursos resulta ineficiente para SMP, existen métodos para poder optimizar y dar solución a estos casos.

- Sistemas UMA (UMA, NUMA, CC-NUMA):

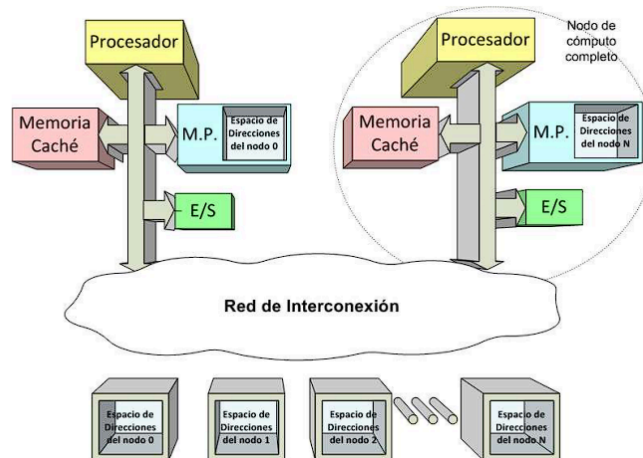
UMA: Define una arquitectura donde todos los procesadores tienen acceso a una misma memoria física mediante un mismo bus de sistema **con un mismo tiempo de acceso a memoria hacia todas las regiones de memoria.**

NUMA: Mismo concepto de UMA, pero con la diferencia de que **el tiempo de acceso a memoria deja de ser uniforme para todos los procesadores**, ya que se hace uso de red para poder acceder a una memoria compartida.

CC-NUMA: Mismo concepto de NUMA, pero con la diferencia que **se hace uso de un controlador (SMC) de memoria**, que se encarga de las comunicaciones entre los procesadores para poder mantener la coherencia, o sea, que pasa a ser todo a nivel de hardware.

Memoria Distribuida: CLÚSTER

- Clúster: Tipo de sistema de procesamiento distribuido conformado por múltiples nodos interconectados mediante redes de alta velocidad donde cada uno hace uso y gestión de su propia memoria, es decir, cada uno gestiona su propio espacio de direcciones. Gran capacidad de prestaciones y escalabilidad, evolución de sistemas SMP.



SEGUNDO FINAL

1)

A)

	Memoria E/S
20h	EOI
21h	IMR
22h	IRR
23h	ISR
24h	INT0
	INT1
	INT2
	INT3
	INT4
	INT5
	INT6
	INT7

Todos los registros están compuestos por 8 bits.

En el IMR cada bit representa una interrupción (del 0 al 7), este registro permite enmascarar selectivamente las interrupciones que va a recibir el PIC. Si el bit N es puesto en 1, la interrupción N estará deshabilitada.

El EOI se encarga de avisar al PIC que la interrupción ya fue atendida.

El IRR es el registro de peticiones de interrupciones.

El ISR es el registro de interrupción de servicios. (El bit que este en 1 será la interrupción que se esté ejecutando actualmente).

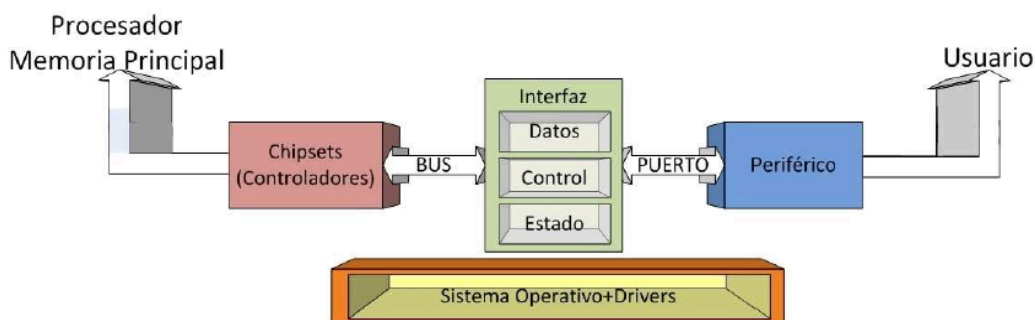
En los INT 0..7 se almacena la dirección de la subrutina a ejecutar en caso de una interrupción... El valor que se carga se multiplica por 4 y el resultado debería ser la dirección a la subrutina.

B)

Las interrupciones por software surgen por dependencia de datos cuando dos instrucciones imponen una comunicación entre ellas mediante datos y estos no llegan a estar disponibles en el momento en el que los requieren. Este tipo de atasco se divide en 3 tipos, los RAW, WAW y WAR. Estas interrupciones se utilizan para implementar subrutinas que solucionan posibles errores y mantienen la coordinación de las instrucciones.

2)

A)



PUERTO: A través de este se realizan envíos de 3 señales.

Señales de control: donde se deja en claro la función u operación que se va a llevar a cabo.

Señales de estado: mediante el cual se notifica el estado de cada uno de los autores.

Señales de datos: Como lo dice su nombre, a través de estas señales se transportarán los datos que sean necesarios enviar como un conjunto de bits los cuales podrán ser enviados en ambas direcciones según sea el caso de la operación a realizar.

B)

- E/S mapeada a memoria: En este modo, los registros de control y estado del dispositivo de E/S se asignan a direcciones de memoria específicas.
- E/S aislada: los dispositivos de E/S están completamente separados del espacio de direcciones de memoria del procesador principal. En este modo, los dispositivos de E/S tienen su propio espacio de direcciones dedicado y son accesibles a través de instrucciones especiales de entrada/salida

3)

A)

Correspondencia Directa:

Un bloque de memoria se asigna/mapea a una sola línea de caché, por lo tanto, cuando este bloque se actualice en memoria, la misma línea de caché asignada también será actualizada

Correspondencia Asociativa:

En este caso un bloque de memoria puede ser asignado o mapeado en cualquier línea de caché, por lo que para este caso se trata de manera distinta la metodología de localización

del bloque almacenado, los bloques ahora cuentan con una etiqueta que sirve para poder ser ubicados en caché.

Correspondencia Asociativa en Conjuntos:

En este tipo de correspondencia la caché se divide en bloques, por lo que el mapeo con la memoria pasa a ser de a conjuntos de líneas (bloques) donde se le da el mismo tratamiento de etiquetado para identificar el bloque en caché.

B)

Políticas de escritura:

Lo que buscan las políticas de escritura es ante diferentes situaciones donde se accede a memoria principal, tratar en caché que los bloques relacionados mantengan la misma consistencia de información, o sea, los datos se mantengan iguales en caso de modificaciones.

- **Write-Through (escritura directa):** Esta política pone como prioridad la integridad de los datos, por lo que si se almacena en caché, también se hará en memoria principal.
- **Write-back (escritura-diferida):** Esta política pone como prioridad el menor acceso a memoria posible, mejorando así la performance y reduciendo la cantidad de accesos, almacenando los datos en caché y solo actualizando en memoria principal cuando se necesite hacer un reemplazo. Esto puede hacer que en algún momento los datos en memoria principal sean erróneos o incorrectos.
-