Preguntas Alejandro

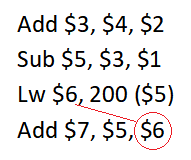
1-Identificar en el código ¿Cuál dependencia causará un stall? Justifique porqué y realice los diagramas correspondientes.

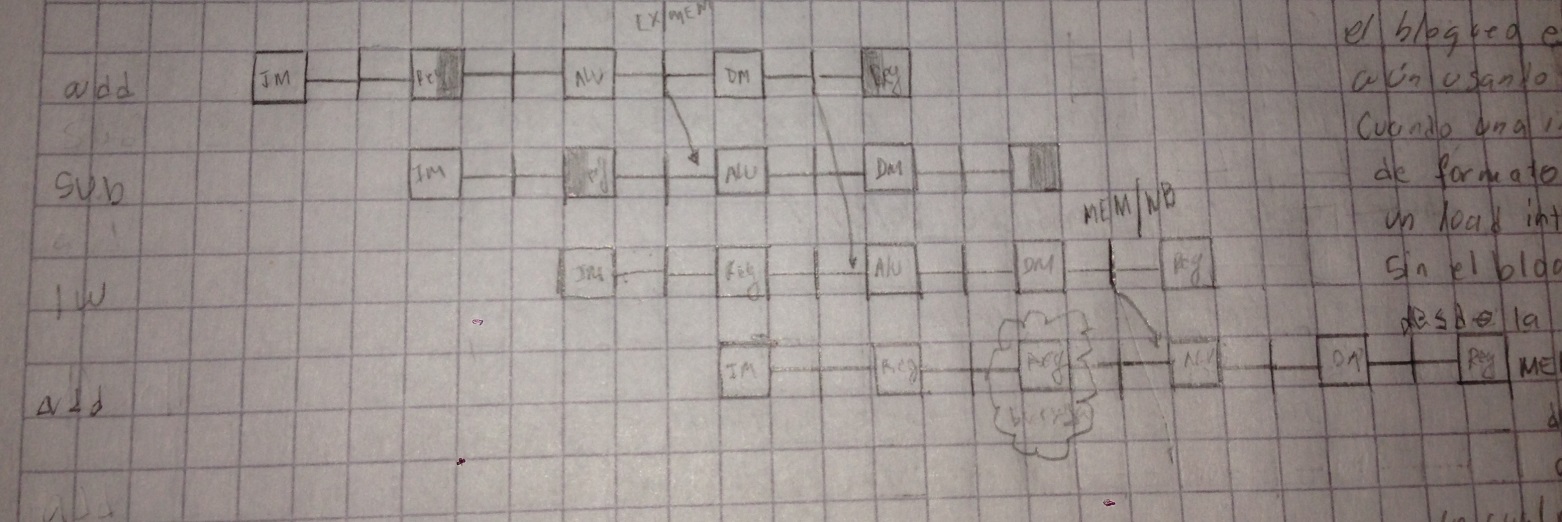
Add $3, $4, $2

Sub $5, $3, $1

Lw $6, 200 ($5)

Add $7, $5, $6

el bloqueo es necesario aun usando anticipación cuando una instrucción de formato R que sigue a un load intenta usar el dato. Sin el bloqueo, el camino desde la salida de la etapa MEM hasta la entrada de la etapa EX iría atrás en el tiempo, lo cual es imposible



2-¿Qué función cumplen los registros de inter-segmentación?

Los registros de inter-segmentación sirven para almacenar los datos de modo que se puedan compartir partes del camino de datos durante la ejecución de instrucciones.

Para conservar el valor de una instrucción individual durante sus otras 4 etapas, se debe guardar el valor leído de memoria en un registro. El mismo razonamiento se aplica a cada una de las etapas de la segmentación, por lo que se deben colocar registros en todas las líneas entre etapas

3-En el código identifique los riesgos existentes, clasificándolos y señalando las dependencias con un círculo sobre los registros involucrados y su flecha correspondiente.

Loop:

lw $t3, $0($t0)

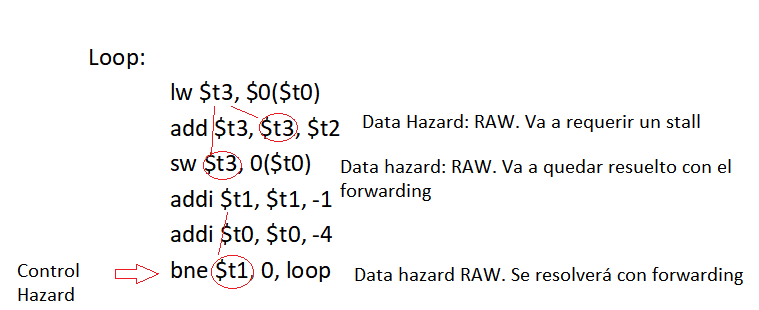
add $t3, $t3, $t2

sw $t3, 0($t0)

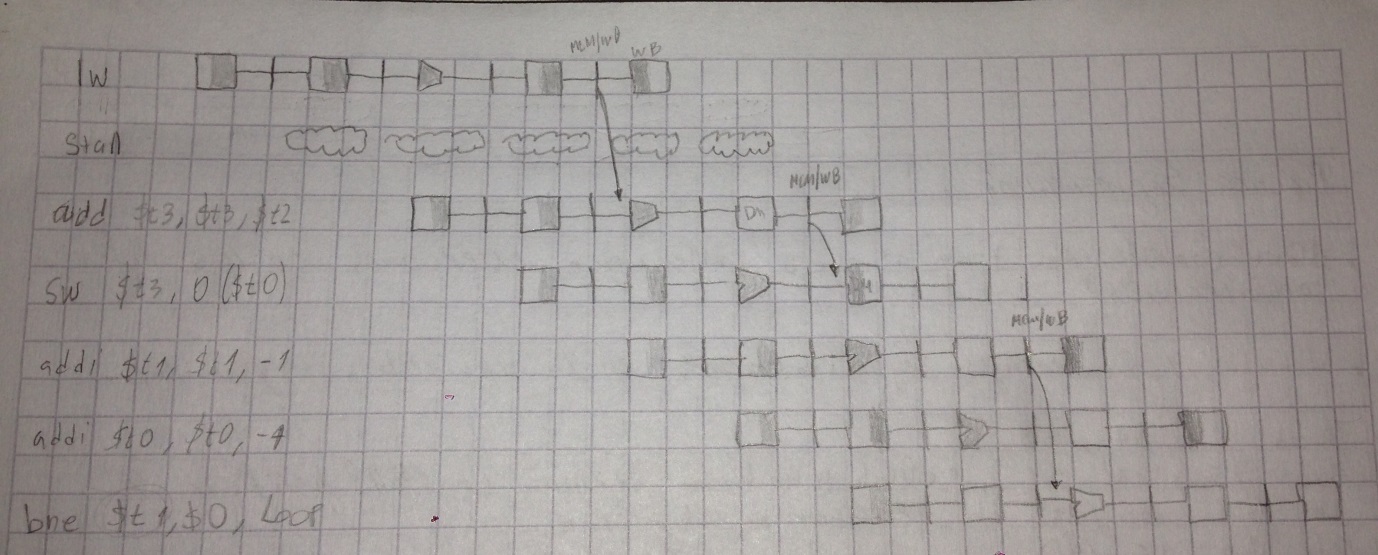
addi $t1, $t1, -1

addi $t0, $t0, -4

bne $t1, 0, loop



4-Utilizando el ejemplo anterior realice los diagramas (según template provisto) correspondientes indicando los forwardings para minimizar las burbujas que se necesitan sin implementar saltos en 2 estados (2-stage branch comletion)



5-Conteste brevemente las siguientes cuestiones, explicándose en el caso que la respuesta sea S/N:

1. ¿Cuál es la ganancia esperada ideal en un procesador segmentado?

En condiciones ideales, el incremento de la velocidad de la segmentación iguala al número de etapas: un pipeline de cinco etapas será cinco veces más rápido comparado con uno no segmentado (por ejemplo: la realización uniciclo en donde el tiempo entre instrucciones es de 8 ns, mientras que en el segmentado es 2 ns).

1. ¿Por qué normalmente no se puede llegar a dicho valor?

Si las etapas están perfectamente equilibradas:

Tiempo entre instrucciones segmentadas = tiempo entre instrucciones no segmentadas/n° de etapas de segmentación.

Esta fórmula sugeriría que una segmentación de 5 etapas debería ofrecer una mejora multiplicada por 5 sobre los 8ns que tiene la realización no segmentada o lo que es igual a un ciclo de reloj de 1.6ns. Sin embargo, el ejemplo muestra que las etapas pueden estar mal equilibradas, por lo que el tiempo por instrucción en una maquina segmentada excederá el mínimo posible y el incremento en velocidad será menor al número de etapas de la segmentación. Además, la segmentación conlleva cierta carga adicional.

1. ¿depende de la distribución de instrucciones?

No precisamente, pero puede verse afectada ya que cuando no encontramos que las instrucciones están distribuidas de manera que pueden presentarse riesgos de datos los cuales no puedan resolverse solo con la anticipación tendremos que recurrir a algún bloqueo lo cual llevará una carga adicional que provocara un retardo y por ende un decremento en el rendimiento. Un ejemplo sería un lw seguido de una instrucción que intenta usar el dato que load va a escribir.

1. ¿Qué ventaja tiene utilizar read after write en el banco de registros?

Permite que una instrucción lea un registro y otra instrucción escriba en un registro en mismo ciclo de reloj.

1. ¿Qué unidades funcionales permites hacer los forwards?

Los multiplexores que se encuentran delante de la ALU en la etapa EX se usan para añadir los caminos de anticipación. Se añade la unidad de anticipación también en esta etapa que va a permitir controlar esos mux para poder elegir entre el valor del registro de propósito general o bien el registro de segmentación correspondiente.

Para el caso que se requiera anticipación hacia la etapa MEM se añade circuitería justamente también en esta etapa.

También se ve involucrada la unidad de detección de riesgos ubicada en la etapa ID utilizada para cuando una instrucción situada después de un load intenta leer el registro que el load escribe. Esta unidad es la encargada de detectar este riesgo y de controlar el multiplexor que elige entre los valores originales del control o 0 en caso que la condición de riesgo de load sea cierta.

1. ¿Es ventajoso utilizar saltos de dos estados para el código dado?

Verdaderamente los más ventajoso para este caso sería utilizar “branch prediction” ya que es mucho más eficiente para tratar bucles.

Pero en el caso que esto no fuese una opción y seria ventajoso en el sentido que podríamos determinar con anterioridad si el salto será tomado o no tomado.

Si usáramos una combinación de 2-stage branch completion + asumir el salto como no tomado, en el caso que el salto sea tomado se tendrá que eliminar solamente una instrucción, en cambio si solamente asumimos el salto como no tomado y recién en la etapa MEM de la instrucción de salto determinamos si se debe tomar o no el salto en ese caso tendríamos que eliminar 3 instrucción.

Entonces vemos como en ese sentido es ventajoso, pero nuevamente no es la solución más adecuada para este tipo de ejemplo ya que asumir el salto como no tomado por más que adelantemos la etapa de cálculo de salto fallará la mayoría de las veces para un bucle.

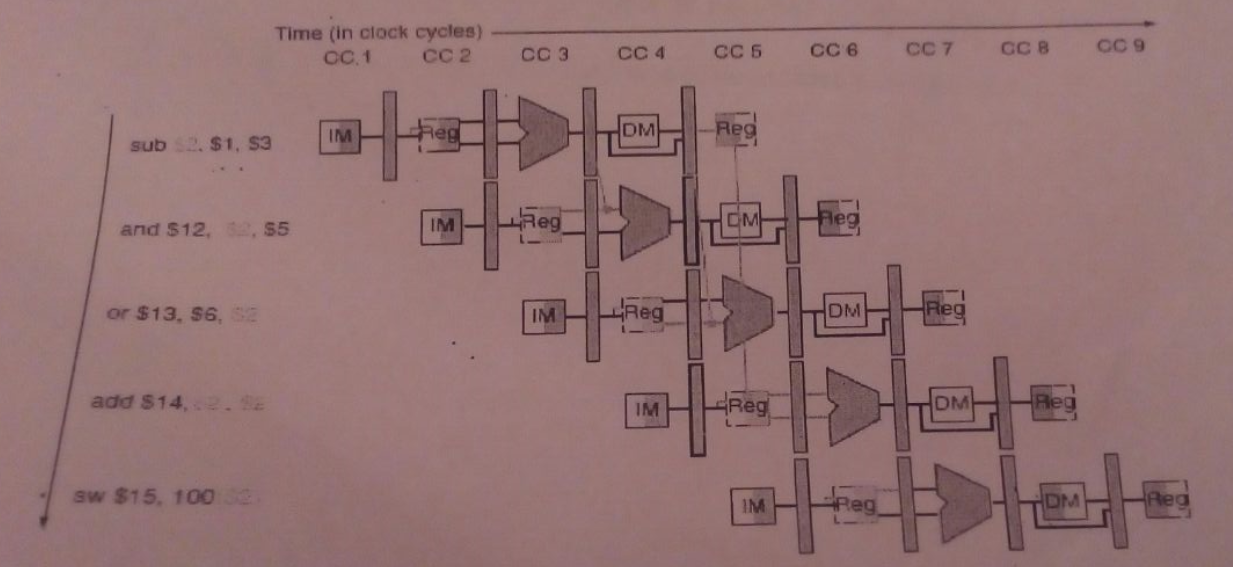
1. ¿Que podría hacer un compilador para reducir los *stalls* en el código dado?

Reordenar el código. Por ejemplo, colocar la instrucción $t1, $t1, -1 seguida del lw

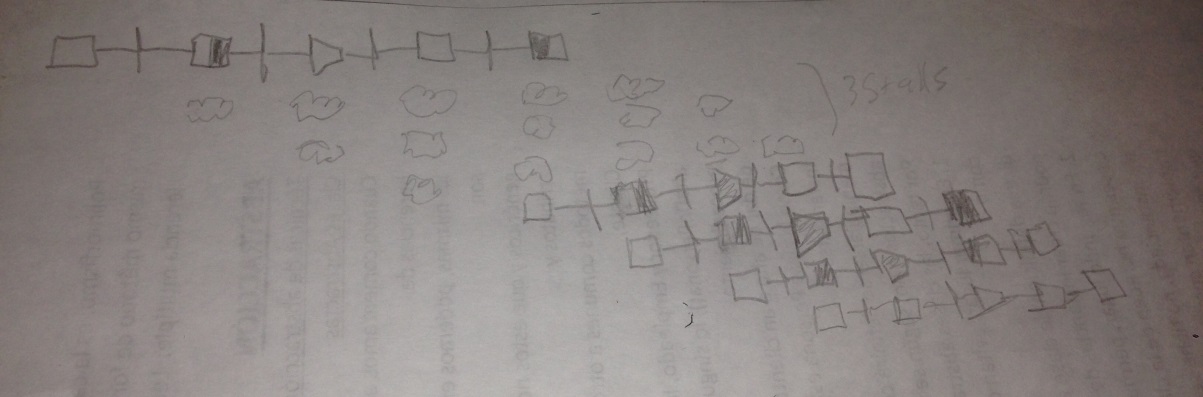
1. ¿Quién es responsable de los cambios de secuencialidad?

El compilador

6-En el diagrama se observan 3 dependencias de datos. Identificar el número de stalls totales bajo las siguientes condiciones: a) no hay forwarding, b) no RAW. Justificar la respuesta



En caso de que si no hubiera forwarding pero si RAW podría resolverse con 2 stalls ya que en el ciclo 5 la lectura se sucedería luego de la escritura, pero como esto no es factible necesitaríamos 3 stalls y al realizar el grafico veríamos que con esos 3 stalls el resto de las dependencias ya estarían cubiertas, ya que la escritura del registro $2 bastante atrás en el tiempo como para que el dato ya esté disponible para el resto de las instrucciones que hacen lectura de ese registro.



7-Indique V o F justificado en el último caso:

1. Los riesgos estructurales solo se pueden resolver con hardware adicional.

V

1. El Forward de datos puede eliminar los riesgos de datos

F. El forward de datos ayuda a reducir los riesgos de datos, pero por ejemplo en el caso de una instrucción lw seguida de una instrucción tipo-R que intenta usar el dato, aun con anticipación deberíamos recurrir a 1 stall, ya que sin el bloqueo la salida de la etapa MEM hasta la entrada de la etapa EX iría atrás en el tiempo, lo cual es imposible.

1. La solución de delayed branch es implementada por el compilador

V

1. Los nop pueden utilizarse para evitar stalls

F. Los nop se utilizan para implementar los stalls

8- Explique bajo qué condiciones los riesgos de control producen stalls.

Esto va a depender de que técnica utilicemos para resolver los riesgos de control.

Asumir el salto como no tomado: en el caso que el salto sea tomado deberá bloquearse el pipeline para poder eliminar las 3 instrucciones que se encuentras en la etapa EX,ID y IF cuando la instrucción de salto está en la etapa MEM.

2- stage branch completion: si asumimos el salto como no tomado y a su vez adelantamos el cálculo de la dirección y la comparación de los registros a la etapa ID entonces el deberá ocurrir un bloqueo para poder eliminar la instrucción que se encuentra en la etapa IF mientras la instrucción de salto está en la etapa ID.

Branch prediction: si la predicción del salto resulta incorrecta se deberá eliminar la instrucción buscada.

Delay branch: en el caso que el compilador no pueda encontrar una instrucción válida para llenar el hueco de salto retardado se llenará ese hueco con una instrucción NOP.

9- ¿Qué complejidades trae apareado implementar excepciones en las diferentes arquitecturas?

10-Ventajas y desventajas según rendimiento en la arquitectura segmentada

Ventaja:

* la segmentación es totalmente invisible al programador.
* Incrementa el número de instrucciones que se están ejecutado a la vez y la rapidez con que las instrucciones empiezan y acaban.
* Mejora la productividad de las instrucciones en vez de del tiempo de ejecución de cada instrucción
* La predicción de saltos, la anticipación y los bloqueos ayudan a hacer un computador más rápido que siga dando las respuestas correctas

Desventaja:

* La segmentación no reduce el tiempo que se tarda en completar una instrucción individual: las 5 etapas hacen que la instrucción tarde 5 ciclos.
* Conseguir una ejecución segmentada no es sencillo. Por ejemplo, vimos que para poder tratar los riesgos de datos implementando forwarding y stall el control se volvía más complejo y debíamos agregar hardware adicional.

Desventaja: Muchas de las dificultades de la segmentación son debidas a las complicaciones del repertorio de instrucciones. Ejemplos:

* Longitudes de las instrucciones y tiempos de ejecución variables pueden llevar a un desequilibrio entre las etapas de la segmentación provocando que algunas se bloqueen más. También puede complicarse severamente la detección de riesgos, así como el mantenimiento de excepciones precisas.
* Modos de direccionamiento muy sofisticados pueden conducir a diferentes clases de problemas. Modos de direccionamiento que actualizan registros complican la detección de riesgos. Otros modos de direccionamiento que requieren accesos múltiples a memoria complican el control de la segmentación y hacen difícil el mantener que las instrucciones avancen por e pipeline uniformemente.

Desventaja: El incrementar la longitud de la segmentación no incrementa siempre el rendimiento.

La combinación de 3 factores, limitan la mejora del rendimiento conseguida por la segmentación:

1. Los riesgos de datos en la segmentación hacen que incrementando la longitud del pipeline se incremente el tiempo por instrucción ya que un porcentaje alto de ciclos se convierten en bloqueos.
2. Los riesgos de control hacen que incrementar la longitud provoque saltos más lentos, incrementando por lo tanto los ciclos de reloj del programa.
3. La sobrecarga en los registros de segmentación puede limitar el decremento en el periodo de reloj obtenido por una segmentación más avanzada.

11- ¿Qué modificaciones hay que realizar en el datapath segmentado para soportar stalls?

12- ¿Qué técnicas conoce para minimizar los riesgos de control?

* 2-stage branch complation
* Delayed branch
* Asumir el salto como no tomado
* Branch prediction

13- En el siguiente código se observa un riesgo tipo RAW en el pipeline que necesita al menos un stall aun usando todas las técnicas por hardware conocidas, sin escalar. Detecte el mismo y explique que ayuda asistida por software podría minimizar dicho riesgo si están soportado en hardware las condiciones adicionales que usted disponga. Tenga en cuenta que lamentablemente dicho procesador no implementa la instrucción nop.

Sub $4, $1, $5

Sub $3, $3, $1

Add $2, $2, $3

Beq $5, $2, L1

Preguntas Walter E/S

1-¿En cuáles condiciones es más conveniente usar E/S programada?

Es conveniente su utilización para dispositivos de E/S de bajo ancho de banda y frecuencia regular y baja ó cuando el procesador no tiene tareas adicionales.

Por ejemplo, un ratón es un dispositivo sólo de entrada, que inicia E/S de forma independiente, cuando el usuario mueve el ratón o pulsa un botón. Para realizar la interficie con el ratón se usa frecuentemente la encuesta, ya que el ratón tiene una velocidad de E/S baja.

Otro ejemplo podría ser el del lavarropas, en donde hay un único proceso corriendo, por lo que no habrá otra cosa por hacer, ahí se le podrá pedir al CPU que controle, entonces se podrá usar un esquema de E/S programada.

2- ¿En qué consiste raid 5? ¿Cuáles son sus ventajas?

Raid 5 es un esquema en el que se intentará reducir el tamaño de la redundancia, esto viene a costa de que en un esquema como el raid 1 si teníamos, por ejemplo, 2 discos de 1 TB se nos presentaba como un único disco de 1TB perdiendo así la mitad de la capacidad de almacenamiento. Entonces justamente un esquema como este aparece para mejorar este punto.

La idea en este esquema es tener varios discos trabajando en paralelo y usar uno de ellos para almacenar información de paridad. Es decir, se escribe una parte en cada uno de los N-1 discos y en el disco N se almacena la información de paridad. Esta información de paridad nos va a permitir que si alguno de los discos falla, a partir de la información de los otros discos y la información de paridad vamos a poder reconstruir la pieza faltante.

El disco N, donde se almacena la paridad, no es siempre el mismo, de esta forma se distribuye la paridad, y esto es un punto importante ya que si siempre almacenáramos es paridad en el mismo disco se volvería un cuello de botella, si se pierde ese disco hay que recalcular la paridad de todo.

Obviamente un esquema como este es mucho más costoso ya que necesitaremos una controladora más inteligente para poder hacer eso, ya que esto se deberá realizar por hardware, ya que si lo hiciéramos por software estaríamos penalizando mucho al procesador para calcular esa paridad.

3-¿Cuáles son las ventajas del uso de raid en general? ¿Qué mejoras incorpora RAID 5?

Cuando hablamos de discos rígidos hablamos de parámetros que no están relacionados solamente con la performance, sino que debemos tener en cuenta cuestiones como la CONFIABILIDAD.

No es lo mismo que, por ejemplo, que deje de funcionar un procesador que un disco, si bien ambos son reemplazables, en un disco yo tengo información, y si esa información no está backaupeada o no tengo algún sistema de redundancia habré perdido esa información.

Lo mismo sucede, por ejemplo con los bancos de memoria en donde deberemos tener mecanismos para sostener el sistema a pesar de alguna falla de sus componentes, y eso es una de las cuesitones claves cuando se ataca la problemática de la DISPONIBILIDAD

**Lo que nos brinda una configuración de raid es, a partir de discos que tienen un posibilidad de falla x, obtener un sistema que tiene una probabilidad de fallas mucho menor, porque ahora para que falle todo el sistema se tiene que dar la falla de más de un disco. Sumado al hecho que me da la posibilidad de mantener la disponibilidad, yo pierdo un disco, pero el sistema sigue funcionando**

Entonces, la idea básica es que un tenemos un disco rígido que es susceptible a fallos y la idea de un esquema de raid es no usar un único disco rígido sino usar un arreglo de discos rígidos lo cual me va a permitir en algunos esquemas incorporar redundancia en caso que algún disco fallé y poder recuperar esa información y/o que mi sistema siga funcionando en caso que algún disco falle. De esta manera garantizamos tanto la confiabilidad como la disponibilidad.

4-¿Cómo mejora Raid 0 la disponibilidad del sistema? ¿Previene los errores lógicos? ¿Previene los errores fiscos?

El esquema de raid 0 es un esquema que no me ofrece redundancia. Pero si me garantiza la disponiblidad del sistema en caso que un disco falle y aparte tiene la ventaja que al distribuir las escrituras se mejora la performance de acceso tanto para la lectura como para las escrituras.

Dicho esto podemos deducir que no previene errores lógicos en el sentido que si algún disco falla no hay posibilidad de recuperar esa información ya que no hay ningún tipo de redundancia , no así previene los errores físicos desde el punto de vista que me garantiza que mi sistema siga funcionando?

5- ¿En cuáles condiciones es más conveniente usar Interrupciones? ¿En cuales DMA?

Las interrupciones son útiles para dispositivos de E/S de bajo ancho de banda y frecuencia irregular, como por ejemplo, un mouse. Mientras que utilizar DMA es más útil para dispositivos de E/S de gran ancho de banda, ej. Disco rígido

6- Conteste brevemente las siguientes cuestiones.

1. ¿Dónde se almacena la dirección de retorno en el llamado a una interrupción?

En el EPC

1. ¿En qué caso se debe inhibir la atención de interrupciones?

Cuando se salta a una rutina de interrupción se carga el registro EPC con la dirección de retorno. En ese momento se deben deshabilitar las interrupciones para evitar que ese registro se pise con otra dirección en el caso que se produzca otra interrupción. Luego si se vuelve a preparar el sistema para permitir que haya interrupciones nuevamente.

1. ¿Qué problemática hay entre DMA y memoria caché?

El DMA genera problemas en sistemas con caches porque puede haber dos copias de un dato: una en cache y otra en memoria. Dado que el procesador de DMA envía peticiones directamente a la memoria, sin pasar por la cache, el valor de la posición de memoria visto por el controlador de DMA puede ser diferente del valor visto por el procesador.

Por ejemplo, consideremos una lectura del disco, que el controlador de DMA pondrá directamente en memoria. Si alguna de las posiciones en las que escribe el DMA está en cache, el procesador recibirá un valor obsoleto cuando realice la lectura. De forma similar, si la cache es de escritura diferida, el DMA puede leer un valor directamente de memoria cuando el valor más actual está en la cache, porque todavía no ha sido escrito en memoria. Esto se denomina problema de los *datos obsoletos* o *problemas de coherencia*

1. Defina latencia y ancho de banda

Latencia: cuando hablamos de latencia hacemos referencia a un tiempo, hacemos referencia a una demora. Por ejemplo: puede ser el tiempo entre que se envía una orden y se ejecuta, o también podría ser el tiempo o demora entre que se envía una señal y ésta llega a destino.

Ancho de banda: está asociado a una cantidad de información por unidad de tiempo. Es la cantidad de información o datos que se pueden transmitir en un periodo de tiempo dado.

* Latencia (latency): Es el tiempo o lapso necesario para que una señal se transfiera de un punto a otro.
* Ancho de banda (bandwidth): Es la cantidad de datos que pueden ser transportados por unidad de tiempo.
* Tiempo de respuesta (elapsed time): Es el tiempo o lapso necesario desde que se envía una orden hasta que se recibe su respuesta.
* Productividad (throughput): Cantidad de operaciones que se completan por unidad de tiempo.

1. ¿En qué casos se requiere arbitraje de bus?

Se requiere arbitraje de bus en el caso que haya 2 o más maestros del bus

1. ¿Cuáles son los parámetros que intervienen en la performance de un disco rígido?

Los parámetros que intervienen son: tiempo de posicionamiento, latencia de rotación, tiempo de transferencia, tiempo del controlador, y tiempo de espera.

7-Analizar el material sobre interrupciones y responder las siguientes consignas

1. ¿Qué cambios de hardware se necesitan para incorporar interrupciones?

Un circuito que me permita realizar el salto en la llegada de una interrupción. Agregar varios registros, como ser el de causa y estado que me permitirán obtener información acerca de las interrupciones

1. ¿Puede presentarse una interrupción en el medio de la ejecución de una instrucción? ¿Qué sucede?

Si, puede presentarse una interrupción en el medio de la ejecución de una instrucción. En ese caso la instrucción termina de ejecutarse para luego ser atendida dicha interrupción.

1. ¿De dónde obtiene el MIPS la información sobre la interrupción?

Del registro de causa

1. ¿Cómo determina el MIPS si deben o no atender interrupciones?

A través del bit 0 del registro de estado el cual indica si las interrupciones están o no habilitadas

Los bits 15 al 8 del registro de estado nos van a permitir manejar una máscara de bits, lo que significa que vamos a poder habilitar o deshabilitar las interrupciones de forma selectiva (son 8 niveles). Por otro lado el bit 0 de ese mismo registro indica si las interrupciones están o no activadas, es decir que si el bit 0 del registro de estado está 0 significa que no va a haber comprobación de interrupciones.

Luego en el registro causa los bits del 15 al 8 muestran banderas sobre interrupciones pendientes. Como mencionamos anteriormente las interrupciones las asociamos a niveles, entonces cuando se genera una interrupción habrá que ubicarla en un algún nivel, a ese nivel se le asocia un bit, por dar un ejemplo, el bit 15, y cuando llega una interrupción para ese nivel el hardware tendrá que poner a 1 ese bit para señalizar que hay una interrupción pendiente.

Luego se comprobará si las interrupciones están habilitadas y en caso de estar habilitadas corroborar si ese nivel en particular también está habilitado. Si se dan todas esas condiciones entonces se realizará el salto.

1. ¿Cómo determina el MIPS si debe o no atender una interrupción determinada?

Además de la comprobación de l bit 0 del registro de estado, se debe comprobar si esa interrupción determinada la cual estará asociada a un nivel está habilitada para ese nivel determinado.

1. ¿Que implica ejecutar una rutina de atención de interrupción? Mencione los pasos y cuales se realizan por hardware y por software.

1: Determinar cuál de todos los dispositivos fue el que causó la interrupción.

2: Determinar el porqué, es decir, la causa de la interrupción.

3: Ejecutar la rutina de interrupción.

Se ejecutará un manejador de interrupciones que será algo genérico que luego se irá derivando a rutinas más específicas ya que cuando se quiera establecer una comunicación con ese hardware específico vamos a necesitar rutinas específicas para ese hardware, estas serán rutinas que agrega el propio driver.

Entonces el SO va a tener un manejador genérico que nos va a permitir determinar quién fue el causante de la interrupción (determinar el dispositivo que generó la interrupción es un mecanismo que se realiza por una combinación de hardware y software), y luego por Software, propio del controlador, se verificará cuál fue la causa de la interrupción y en consecuencia a esto se obrará.

Supongamos que se está ejecutando un proceso P0 y en algún momento llega una interrupción, obviamente primero deberá terminar de ejecutarse la instrucción que está en ejecución antes de poder atender la interrupción. Llegado ese momento se deberá ejecutar una rutina de interrupción que es propia del SO, es decir, está dentro del contexto del SO, mientras que lo otro estaba dentro del contexto de usuario. Entonces vemos como esto conlleva a un cambio de contexto por lo cual deberemos salvar ese contexto (esto será guardar algunas cuestiones mínimas como el PC) para luego una vez que se haya terminado de atender la interrupción poder volver al contexto anterior. Esto de saltar a ejecutar una rutina de interrupción y guardar la dirección de retorno es algo que se hace por hardware.

Luego una vez dentro la rutina de interrupción, en la fase inicial tendremos que salvar los registros necesarios para ejecutar esa rutina y eso ya es algo que se realiza por software.

1. ¿Dónde guarda el MIPS la dirección de retorno de la subrutina? ¿Por qué?

Se guarda en el EPC. Se utiliza este registro especial y no, por ejemplo, el $ra ya que no hay forma de garantizar que ese registro $ra no esté siendo usado. Nosotros no sabemos cuándo puede llegar una interrupciones ya que estas son asincrónicas y puede suceder que cuando llega la interrupción tengamos en el $ra un dato que todavía no fue salvado, entonces para garantizar que no se pise la información se usa este registro especial llamado EPC.

1. ¿Cuál es el papel del SO?

Cuando se desea hacer una operación de e/s se generará una llamada al SO, el SO iniciará la operación, y en algún momento el controlador de e/s generará una señal de interrupción para comunicar que ha terminado o bien porque requiera atención y debe ejecutarse alguna rutina de atención de interrupción.

Dado que las interrupciones provocan el paso a modo supervisor éstas deben ser atendidas por el SO. Estas rutinas se encuentran dentro de contexto del SO.

10- Analizar el material sobre DMA y responder las siguientes consignas

1. ¿Qué cambios de hardware se necesitan para incorporar DMA?

El DMA se realiza con un controlador especializado que transfiere los datos entre el dispositivo y la memoria.

Además muchos controladores de DMA tienen algo de memoria que les permite adaptarse a los retardos producidos en la transferencia o al esperar la concesión del bus.

Se necesitará incorporar interrupciones y todo lo que ello conlleve para que cuando el DMA complete una operación de e/s pueda señalizar si esa operación terminó y si terminpo correctamente

1. ¿Qué problema se presenta con el bus y el procesador al incorporar DMA?

Al incorporar DMA ahora tendremos 2 o más entidades que querrán hacer uso del bus. Si el DMA quiere acceder a la memoria va a usar el bus y por ende el procesador no puede.

1. ¿Cuáles son las alternativas para solucionarlo?

Una alternativa es usar algún esquema de arbitraje de bus que me permita quien puede usar el bus en cada momento.

Otra alternativa es tener una caché, la cual nos reducirá el impacto del DMA ya que, debido a los principios de localidad temporal y espacial, el procesador muy probablemente se las pueda arreglar usando solamente la cache sin tener que estar accediendo constantemente a memoria principal. Sin embargo el incorporar una cache trae apareado complejidades

1. ¿Qué problema se presenta con la memoria cache al incorporar DMA?

El DMA genera ciertos problemas en sistema con caches porque puede haber dos copias de un dato: una en cache y otra en memoria. Dado que el procesador de DMA envía peticiones directamente a la memoria, sin pasar por la cache, el valor de la posición de memoria visto por el controlador de DMA puede ser diferente al valor visto por el procesador.

Por ejemplo: consideremos una lectura del disco, que el controlador de DMA pondrá directamente en memoria. Si alguna de las posiciones en las que escribe el DMA está en cache, el procesador recibirá un valor obsoleto cuando realice la lectura. De forma similar, el DMA puede leer un valor directamente de memoria cuando el valor más actual está en la cache, porque todavía no ha sido escrito en memoria. Esto se denomina problema de los datos obsoletos o problema de coherencia.

1. ¿Cuáles son las alternativas para solucionarlo?

Existen 3 técnicas:

* Encaminar las operaciones de E/S a través de la cache. Esto asegura que las lecturas obtienen el valor más reciente, mientras que las escrituras modificas los datos en la cache.
* El SO invalida de forma selectiva el contenido de la cache antes de una operación de E/S de lectura, o que fuerce la actualización de la memoria principal con el contenido de la cache antes de una operación de de E/S de escritura.
* Incorporar un mecanismo de circuitería para invalidar de forma selectiva entradas de la cache.

La primer técnica es costosa y tiene un fuerte impacto negativo en el rendimiento del procesador, ya que los datos de E/S no suelen usarse inmediatamente y pueden desplazar fuera de la cache datos útiles que necesita el programa que se esté ejecutando.

La segunda técnica requiere una pequeña cantidad de soporte de circuitería y es más eficiente si el software puede realizar la operación fácil y eficientemente. Debido a que el vaciado de partes grandes de la cache sólo ocurre en caso de transferencias de DMA, será infrecuente.

La tercera técnica que es ese mecanismo de invalidación para asegurar la coherencia es típico de sistemas multiprocesadores y la misma técnica puede usarse para E/S.

Invalidar la cache se hace por hardware o por software, por software me demanda más tiempo y por hardware más circuitería

1. ¿Qué problema se presenta con la memoria virtual al incorporar DMA?

La dificultad con las direcciones virtuales es que el controlador de DMA debe realizar la traducción de la dirección virtual a física. El problema más importante cuando se usan direcciones físicas en una transferencia por DMA es que la transferencia no puede traspasar los límites de la página, entonces las posiciones de memoria involucradas en la transferencia no estarán contiguas en memoria física: las posiciones de memoria corresponderán a múltiples páginas virtuales, cada una de las cuales puede estar asignada a cualquier página física. Por tanto, si se usan direcciones físicas, hay que restringir las transferencias de DMA de forma que en cada una de ellas sólo se vea involucrada una página.

1. ¿Cuáles son las alternativas para solucionarlo?

* Un método que permite iniciar transferencias de DMA que sobrepasen los límites de las páginas es hacer que el DMA trabaje con direcciones virtuales. En este tipo de sistemas, la unidad de DMA tiene un pequeño número de registros que proporcionan la correspondencia entre páginas virtuales y físicas para la transferencia. El SO proporciona la información cuando se inicia la operación de E/S. Utilizando esta información, el DMA no necesita preocuparse de la posición ocupada por las páginas virtuales involucradas en la transferencia.
* Otra técnica consiste en hacer que el SO descomponga la transferencia de DMA en varias transferencias más pequeñas, cada una de ellas confinada dentro de los límites de una página. Las transferencias se encadenan y son gestionadas por un procesador de E/S o por un controlador de DMA inteligente que realiza la secuencia completa de transferencia; alternativamente, el propio SO puede iniciar individualmente cada una de las transferencias.

Sea cual sea el método el SO debe asegurarse de que no se cambie la ubicación de una página en memoria mientras se está realizando una transferencia por DMA qe involucre dicha página.

1. ¿Cuál es el papel del SO? Se responde con la pregunta anterior

11- Realizar un cuadro comparativo entre E/S programada, Interrupciones y DMA, donde se detalle: ventajas, desventajas, complejidades y contexto en el que es recomendable su aplicación

E/S Programada

* Ventajas: forma más sencilla de comunicación
* Desventajas: procesador mal gastado en esperas
* Complejidades: dificulta el manejo de prioridades
* Contexto en el que es recomendable su aplicación: útil para dispositivos de bajo ancho de banda y frecuencia regular y baja o cuando el procesador no tiene tareas adicionales

Interrupciones

* Ventajas: libera al procesador de las esperas
* Desventajas: mayor costo, mayor complejidad.
* Complejidades: se deben implementar rutinas de atención de interrupciones las cuales van a implicar un cambio de contexto.

Deberemos incluir información adicional como ser registros que nos sirvan para obtener información sobre las interrupciones

Deberemos establecer una manera de poder determinar las prioridades de estas interrupciones.

* Contexto en el que es recomendable su aplicación: útil para dispositivos de e/s de bajo ancho de banda y frecuencia irregular

DMA

* Ventajas: el controlador de DMA realiza la transferencia entre el dispositivo de E/S y memoria sin intervención del procesador
* Desventajas: mayor complejidad, mayor coste, se necesitará algún esquema de arbitraje de bus ya que ahora habrá al menos 2 maestros del bus.
* Complejidades: se requiere interrupciones, problema de cache coherencia, problemas con el uso de memoria virtual
* Contexto en el que es recomendable su aplicación: útil para dispositivos de E/S de gran ancho de banda. Ejemplo: disco rígido.

12-Describa los pasos a seguir para realizar una operación de lectura de E/S desde espacio de usuario utilizando interrupciones.

Primero se envía la orden de lectura y el procesador puede realizar otra cosa.

Cuando una interrupción llega se lee el estado del dispositivo y así se corrobora si hay una condición de error o no. Si hay una condición de error, se tendrá que ejecutar la rutina correspondiente; si no hay condición de error y el dispositivo está listo, se lee la palabra del dispositivo y luego se escribe la palabra en memoria.

13- Diferencia entre un host convencional y un servidor

-Un servidor tendrá un esquema de raid más complejo, como raid 5 o 6, un servidor tendrá muchos más discos rígidos que una PC de escritorio.

-El requerimiento de RAM en un servidor será mayor ya que se podría pensar que el hardware de un servidor va a servir de base para correr n máquinas virtuales, entonces el uso de RAM ahí va a ser importante.

-Seguramente un servidor no tenga mucha performance para lo que es placa de video. Un servidor probablemente no va a tener placa de sonido.

-Cambiarán los buses. La conexión de un disco rígido en el caso de una pc de escritorio será un conector SATA y en un servidor probablemente sea un conector SAS.

-Las controladoras de disco serán mucho más complejas en un servidor, serán controladoras con mayor inteligencia.

- Un servidor debe poder informar fallas. Las memorias van a tener control de paridad para poder reconocer si sucede un fallo en la memoria RAM, detectarlo, informarlo y corregirlo.

-La calidad de los componentes en un servidor deberá ser superior.

- Un servidor tendrá más de un procesador.

-En un servidor habrá varias fuentes alimentación, para que si alguna deja de funcionar el sistema pueda seguir funcionando

14- Comparación con una PC de escritorio y un sistema embebido

-La PC de escritorio es algo mucho más genérico mientras que en el caso de un sistema embebido va a ser algo muy específico, creado para un fin que ya esté auto contenido. Por ejemplo, un televisor es un sistema embebido, tiene hardware específico para decodificar audio y video.

- Los tamaños de RAM en un sistema embebido serán mucho más acotados.

- Las interfaces de usuario serán mucho más acotadas.

-Los buses internos no serán los mismos para un sistema embebido que para una PC de escritorio. Una PC de escritorio usará típicamente un PCI y un sistema embebido puede usar un I2C

15- Raid 0

No hay redundancia. Si tenemos un disco A y un disco B y configuramos un Raid 0, lo que tendremos es un disco lógico donde el tamaño del disco es igual al tamaño de A + el tamaño de B.

Aquí la funcionalidad del raid me permite que mi SO vea un disco que en realidad es la suma de ambos discos. La ventaja de esto es distribuir las escrituras y así poder disminuir el tiempo empleado en escritura y obviamente también para la lectura. Entonces el RAID 0 mejora la performance de acceso tanto para lecturas como escrituras.

16. DMA

se ha inventado un mecanismo para descargar de trabajo al procesador y permitir al controlador del dispositivo pueda transferir los datos directamente de la memoria o hacia ella sin la intervención del procesador. Este mecanismo se llama *acceso directo a memoria (DMA)*. El dispositivo continúa usando el mecanismo de interrupciones para comunicarse con el procesador, pero sólo al finalizar la transferencia o en caso de error.

El controlador de DMA puede actuar como maestro del bus y dirigir las lecturas y escrituras entre él mismo y a memoria. Hay tres pasos en una transferencia de DMA:

1. El procesador inicia el DMA proporcionando la identidad del dispositivo, la operación a realizar, la dirección de memoria que es la fuente o el destino de los datos a transferir y el número de bytes a transferir.
2. El DMA inicia la operación en el dispositivo y se encarga de obtener el acceso al bus. Cuando los datos están disponibles (procedentes del dispositivo o de memoria), los transfiere. El controlador de DMA proporciona la dirección de memoria para las lecturas o escrituras. Si la petición requiere más de una transferencia por el bus, el controlador de DMA genera la siguiente dirección de memoria e inicia la siguiente transferencia. Usando este mecanismo, el controlador de DMA completa la transferencia, que puede tener un tamaño de miles de bytes sin molestar al procesador. Muchos controladores de DMA tienen algo de memoria que les permite adaptarse a los retardos producidos en la transferencia o al esperar la concesión del bus.
3. Una vez finalizada la trasferencia por DMA, el controlador interrumpe el procesador, que puede determinar si la operación ha terminado con éxito, interrogando al controlador de DMA o examinando la memoria.

17. Raid

RAID (redundant array of independent disks)

La idea básica es: si yo tengo un disco rígido, como el descripto anteriormente, el cual es susceptible a fallas, y si falla puedo perder mucha información. ¿Como podemos evitar, o al menos, mitigar este problema? Hay varias alternativas. Muchas de estas alternativas son complementarias.

La idea de raid es justamente no usar 1 disco rígido sino usar un arreglo de discos rígidos, un conjunto de discos que se vean hacia el exterior como una única unidad. Entonces iremos de un sistema más sencillo a un sistema más complejo.

RAID 0

Es un caso de RAID donde en realidad no hay redundancia, yo tengo el disco A y el disco B y si configuro un raid 0, lo que tengo es un disco lógico (DL) donde este DL es igual al tamaño de A + el tamaño de B. No hay redundancia por eso DL = A+B.

Entonces aquí la funcionalidad del raid me permite que mi SO vea un disco que en realidad es la suma de ambos discos. La ventaja de esto es DISTRIBUIR LAS ESCRITURAS, entonces cuando yo escribo, en lugar de estar escribiendo solamente en A, escribo en A y en paralelo escribo en B. Ejemplo: si yo tengo que escribir “cosa” escribo “co” en A y “sa” en B. Entonces, lo que se hace es distribuir las escrituras, lo cual tiene una ventaja que radica en que al hacer escribir en paralelo lo que se hace es disminuir tiempo empleado en escritura, y obviamente también se va a ver en la lectura. Entonces, MEJORA LA PERFORMANCE DE ACCESO TANTO PARA LECTURA COMO PARA LE ESCRITURA. Pero si por alguna razón alguno de los 2 discos falla, pierdo todo, ya que no hay redundancia de información.

RAID 1

Si tengo 2 disco A y B, el DL = menor (A, B). Ejemplo: si yo tengo 2 discos de 1 TB (total=2TB), lo que yo voy a ver como disco va a ser 1 TB porque ahora cuando queramos escribir escribiremos en unos y usaremos el otro como un espejo. Ejemplo: si quiero escribir “cosa” escribo “cosa” en A y escribo “cosa” en B. Con lo cual lo que ganamos es incluir el concepto de redundancia. Esto es como decíamos anteriormente ESTOAMOS DULPICANDO LA INFROMACION POR LO QUE ESTAMOS YENDO AL EXTREMO DE LA REDUNDANCIA. Pero obviamente si algún disco el equipo puede seguir funcionando sin ningún problema.

PUNTO IMPORTENTE: para diferenciar algunas cuestiones, vemos que de esta forma con raid 1 estamos soportando una falla sin perder información. Ahora si yo tuviese un back up tampoco perdía información, suponiendo que el back up estuviese actualizado, sin embargo, con este mecanismo de raid 1 el disco rígido se rompió, pero el SISTEMA SIGUE FUNCIONANDO, el servidor sigue sirviendo, en cambio si yo tengo un único disco y se me rompe por más que no pierda información porque tenía un back up, estuve perdiendo tiempo teniendo que cambiar el disco, y es tiempo que ESTUVE SIN SERVICIO. Entonces vemos que con raid 1 HAY REDUNDANCIA QUE ME GARANTIZA LA DISPONIBILIDAD DEL SISTEMA, es redundancia que está metida dentro del sistema.

Entonces en raid 1 la escritura es como si fuera de un único disco, porque todo lo que yo escriba en un disco se escribirá también en el otro.

ESTE CONCEPTO DE RAID SE PUEDE APLICAR O POR SOFTWARE O POR HARDWARE. Si yo hago un raid por software ese trabajo lo va hacer el PROCESADOR, entonces estaré usando tiempo del procesador para gestionar el tema del array. Si lo hago por hardware ese trabajo lo hará la CONTROLADORA y es trabajo que el procesador le delega al hardware, que para el procesador es transparente porque típicamente la controladora es la que es configurada en algún tipo de RAID, y es la que se encarga.

También hay versiones tipo mix en donde la controladora hace algunas cositas y el procesador hace 2 o 3 cositas; todo DEPENDE DE CUANTO ESTE YO INTERESADO EN PAGAR POR HARDWARE, porque obviamente una controladora que haga uso de un RAID realmente eficiente será más cara que una controladora que haga alguna cosita por hardware y delegue todo el resto al procesador, es siempre una solución de compromiso.

LA CONTROLADORA ES LA INTERFAZ ENTRE EL PROCESADOR Y EL DISPOSITIVO EXTERNO. EL PROCESADOR HABLA CON LA CONTROLADORA A TRAVES DE ALGUNOS REGISTROS Y ES LA QUE LUEGO HABLA CON EL DISPODITIVO EXTERNO.

RAID 3 y RAID 4: no son realmente usados.

RAID 5

Es un esquema en el que intentaremos reducir el tamaño de la redundancia, porque como vimos en RAID 1, si yo tenía 2 discos de 1 TB, se me presentaba como 1 disco de 1TB perdiendo así la mitad de la capacidad de almacenamiento. Entonces este es un punto a mejorar, para eso aparecieron los esquemas de RAID 2, 3 ,4 y 5, en donde la idea es tener varios discos trabajando en paralelo y usar uno de esos discos para almacenar información de PARIDAD, lo que significa esto es que, por ejemplo, si yo quiero almacenar la palabra “cosa” escribo “c” en el disco A, “o” en el disco B, “s” en el disco C, “a” en el disco D y en el disco E agrego INFORMACION DE PARIDAD. Esta información de paridad me va a permitir que cualquiera de esos discos que se pierda, a partir de la información de los otros que no se perdieron y a partir de esa información de paridad reconstruir la pieza faltante. Entonces si se me daña el disco B, a partir de la información del disco A, C, D y la información de paridad que está en el disco E, yo puedo deducir que en el disco B había una “o”. Es decir que en este esquema la redundancia va a estar dada por esa información de paridad. En este este esquema estamos usando 1 disco de 5, en raid 1 usábamos un disco de 2 que teníamos, entonces vemos que este mecanismo es mucho más eficiente para almacenar esa redundancia.

Obviamente este esquema es más caro ya que NECESITO UNA CONTROLADORA MUCHO MÁS INTELIGENTE para poder hacer eso, ya que tenemos que hacerlo por hardware (se encarga la controladora) porque si lo hiciéramos por software estaríamos penalizando mucho al procesador para calcular esa paridad, y reconstruir el arreglo si se cae.

Si bien no vamos a hablar de RAID 2, 3 y 4 vamos a ver un punto que es la diferencia entre raid 2, 3, 4 y el 5.

El 5 lo que tiene es justamente una mejora de los otros raid, es como distribuye la información de paridad. Por ejemplo: antes almacenamos “cosa” de la siguiente manera: C O S A \* (el ultimo es la información de paridad). Ahora si yo quiero almacenar “pepe” podría hacer lo siguiente: P E \* P E, y vemos como la información de paridad ya no está en el último disco, sino que está en el tercero.

El tema es que, si yo USARA UN UNICO DISCO PARA MANEJAR TODO LO QUE INFORMACION DE PARIDAD ESE DISCO SE VUELVE UNA ESPECIE DE CUELLO DE BOTELLA, donde si falla ese disco yo tengo que recalcular la paridad de todo, y eso en la práctica genera una sobrecarga del sistema, entonces lo que PROPONE EL RAID 5 ES DISTRIBUIR ESA PARIDAD. Entonces al almacenar P E \* P E, la redundancia no va a estar siempre en el mismo disco, sino que se va a ir distribuyendo en todos los discos, de forma de distribuir la carga.

Entonces observemos nuevamente cuestiones de performance, en RAID 5 de nuevo puedo escribir en paralelo, tengo 4 discos para escribir en paralelo y aparte de eso estará generando la paridad para llenar el quinto disco. Como dijimos aquí en RAID 5 tendremos SI O SI NECESITAR UNA CONTROLADORA POR HARDWARE, porque por software penalizaremos de gran manera al procesador.

Obviamente esto del RAID 5 es para algún servidor, la PC de escritorio tienen una controladora que puede usarse para generar un raid 0 o raid 1, no creo que raid 5.

Observar una cuestión IMPORTANTE: cuando hablamos de raid 0 dijimos que fallaba un disco se caía el sistema, en raid 1 si se cae un disco el sistema siguen funcionando, en raid 5 lo que sucede cuando se pierde un disco rígido, por ejemplo si se pierde el disco que tenía la “a” de la palabra C O S A \*, lo que voy a tener que hacer es leer todos los discos sanos + el disco que tiene la información de paridad y calcular lo que había en el disco que falló. Entonces por eso decimos que la controladora tiene que ser mucho más inteligente para hacer eso de forma eficiente porque si no el procesador para leer cada pedacito tiene que hacer un montón de cálculos, es por eso que es un sistema mucho más complejo; yo ahorro en discos, en discos en el sentido de espacio de almacenamiento, porque antes desperdiciaba la mitad y ahora desperdicio solo 1/5, pero ahora también necesito más complejidad.

Observemos que es importante distribuir la paridad, porque si yo tuviese toda la paridad en único disco y se pierde no pasa nada porque yo accedo a la información normalmente; ahora si en lugar de perderse ese disco antes mencionado se pierde otro yo para ACCEDER A CADA BLOQUE TENGO QUE SIEMPRE CALCULAR, en cambio si la paridad está repartida, se pierde tal disco allá lo calculo porque tengo que reconstruir uno y el otro no????????????????.

Ahora, ¿Qué pasa si en este esquema de RAID 5 perdemos dos discos en lugar de solo uno? Lo que sucede es que este esquema de RAID 5 SOPORTA SOLO LA FALLA DE UN DISCO, cualquier disco, pero solo uno, si fallan dos discos al mismo tiempo SE CAE EL SISTEMA.

El RAID 5 se puede aplicar con un mínimo de 3 discos, de ahí en más no hay limitaciones.

EL RAID VA A APARECER CUANDO YO QUIERA GARANTIZAR LA DISPONIBILIDAD DEL SISTEMA, Y DE QUE SI SE ME CAE UN DISCO EL SISTEMA PUEDA SEGUIR FUNCIONANDO (por ejemplo: el banco Santander Rio no va a tener todos los datos de las cuentas en un solo disco rígido, seguramente tendrá algún tipo de RAID)

RAID 6

En este esquema se puede soportar la pérdida de dos discos de forma simultánea.

RAID 5 + 1 DISCO DE SPARE

Si se cae un disco en un RAID 5 tengo que cambiarlo, y obviamente mientras reemplazo ese disco que falló, lo conecto y se va recuperando automáticamente la información, esto se conoce como reconstruir el RAID, y esto puede demorar varias horas. Pero ahora podemos ser más inteligentes:

Tendremos todo ese RAID 5, como vimos antes (5 discos), y aparte tengo 1 o más discos spares, que son discos que están ahí vacíos de repuestos, y cuando alguno de mis 5 discos falle, la controladora se dará cuenta, lo marcará como fallado y empieza a usar uno de esos discos spares para reconstruir el raid. Ahí obviamente nos ahorramos un montón de demoras. Entonces este tipo de soluciones nos permiten dejar el sistema al 100%, es decir que soporte nuevamente la pérdida de un disco. Obviamente en este esquema si se caen dos discos al mismo tiempo sigue fallando el sistema; acá lo que nos ahorramos es esa operatoria manual que implica darme cuenta de que falla un disco, ir y cambiarlo y hacer click para que se reconstruya el raid.

Obviamente que son importante en un lugar donde la información tiene valor. Obviamente quizás si me pasa a mí en PC no pasa nada, pero en un banco no es lo mismo, entonces hay que entender el contexto en que estamos hablando.

RAID DE 2 NIVELES -> R1+0

Usaremos un conjunto de 4 discos para hacer un esquema de un raid en 2 niveles. En el Primer nivel usamos el disco A y el disco B para hacer un R1 (es decir que lo que está en el disco A es lo mismo que está en el disco B, es decir que se espeja), y lo mismo hacemos con el disco C y el disco D, hacemos otro R1. Entonces tendremos varios raid en espejo, pero la idea es verlo como una única unidad, entonces con todo eso (los 4 discos) genero un raid 0. Entonces, desde la perspectiva del sistema operativo yo tengo un gran disco rígido que es confiable, porque en realidad si se me cae un disco tengo el espejo y que me sirve también para describir bien las escrituras, yo escribo en un disco y en paralelo escribo en el otro.

Ahora en este esquema, si se cae A o B o C o D todo ok; se cae B y D o B y C o A y C o A y D o C y B todo ok, se cae A y B o C y D chau.

18. Memorias Flash

Ahora hablaremos generalidades de una Memoria Flash, que es otro dispositivo de almacenamiento muy distinto. No posee partes mecánicas, lo cual nos trae ventajas. Primero: los tiempos antes mencionados desaparecen, entonces veremos que una memoria flash es mucho más rápida que un disco regido, también son más resistentes por no tener partes mecánicas. Y otra cuestión es el costo por unidad de byte, un disco rígido es mucho más económico cuando hablamos de precio por unidad de almacenamiento.

Típicamente hay 2 tipos de memoria flash: NAND o NOR

. La cuestión importante, ya que veremos memorias en un pen drive, en una SD, en los celulares, en los discos de estados sólidos, en todos estos ejemplos hay memorias de este tipo, y en ese tipo de memorias una característica fundamental es el desgaste de la misma, típicamente las NAND, que son las más económicas, tienen una cuestión que es que a medida que yo las escribo estas memorias van desgastándose, por lo tanto se dice que cada sector de una memoria NAND tiene una cantidad máxima de escrituras. Cuando se llega a ese desgaste ya la información que está ahí no es confiable. Para atacar esta problemática es este tipo de memorias hay una parte adicional que sirve para tener información de la información (metadatos que dicen por ejemplo la cantidad de escrituras que hubo en x lugar, me da información de validar para tratar de corregir errores o detectarlos al menos), y además todo eso está metido dentro de un chip que posee una controladora, esa controladora lo que va a intentar es distribuir las escrituras en todos los sectores de la NAND para que el desgaste sea parejo. Entonces muchas de las diferencias que hay en los distintos tipos de memorias flash está dada por la tecnología que usan y el tipo de control que tienen para manejar y hacerla más robusta.

La NOR es similar, pero tecnológicamente distinto lo que la hace mucho más fiables, pero mucho más cara. Otra característica de las memorias flash es que se escriben por sectores; yo cuando quiera escribir no puedo cambiar solamente un bit, tengo que cambiar todo el bloque para actualizar ese sólo bit, entonces es por eso que parte de esa controladora también se encarga de hacer esa escritura de forma eficiente y no desgastar inútilmente la memoria**. Lo importante de esta parte es que nosotros vimos la problemática de e/s como la idea de conectar la computadora, el corazón de esta, que es justamente el procesador y la memoria con el resto del mundo, y este es un ejemplo de dispositivo de e/s y nos sirve justamente para entender que hay problemáticas que van más allá de lo que es la e/s que vimos hasta ahora, que hay problemáticas específicas de cada dispositivo. Así como ahora estamos mencionando problemáticas sobre unidades de almacenamiento (mencionando 2: memorias flash y discos rígidos) para hacer una comparativa, pero cuando hablamos por ejemplo de conexión a redes también nos vamos a encontrar con otras problemáticas. CADA DISPOSITIVO DE E/S YA QUE CUMPLE UNA FUNCION ESPECIFICA Y ESTA ATADO A UN MONTON DE TECNOLOGIAS TIENE SU PROPIA PROBLEMÁTICA, por eso decimos que la problemática de e/s es muy amplia. Lo importante a saber para Lozano es saber cómo funciona el disco rígido, entender los conceptos que hay detrás de una memoria flash y entender en que se basan esas diferencias que hablamos. Muchas de las características del disco rígido están justamente atadas a su constitución, a que es un elemento netamente mecánico y eso le da ciertas limitaciones; la memoria flash también tendrá sus limitaciones.**

El disco rígido se va desgastando porque va perdiendo sus propiedades magnéticas a causa del envejecimiento, los golpes hacen que ese mecanismo no sea confiable.