lo cual se deduce que duplicar el tamaño de línea tiene un impacto negativo sobre el tiempo de acceso a memoria.

5.11. Problema 11

Una caché tiene líneas de 4 palabras, y el siguiente nivel inferior en la jerarquía de memoria se implementa mediante DRAM en modo página con una latencia para la primera palabra de 100 ns, y una latencia de 10 ns para las siguientes palabras accedidas. Suponga que el 25 % de todas las líneas de caché que tengan que ser eliminadas están modificadas, y que en media en una línea modificada se escribe 5 veces antes de que sea eliminada.

1. ¿Cuál es el tiempo medio de acceso de una línea en la caché si se utiliza una política de escritura inmediata? ¿Y si es de post-escritura?
2. ¿Qué tipo de caché, escritura inmediata o post-escritura, tardará más tiempo en escribir datos al siguiente nivel de la memoria?
3. ¿Cuántas veces, en media, se tiene que escribir en una línea de caché para que la respuesta al apartado b) se modifique?

Solución:

1. Una escritura inmediata implica una penalización de $100 + 3 * 10 = 130$ns y la de post-escritura $0100 + 3 * 10 + 0,25 * 130 = 162,5$ns.
2. Ambas hacen el mismo número de escrituras, la de escritura inmediata toma 100 ns por cada escritura y como realiza 5 escrituras/línea en total tenemos 500 ns por línea, mientras que la de post-escritura tendrá 130 ns por línea independientemente del numero de escrituras.
3. La post-escritura necesita 130 ns/línea con lo cual estadísticamente son 1,3 accesos a memoria principal por tanto el número de escrituras medio debe ser menor de 1,3

5.12. Problema 12

Un computador tiene una memoria principal de 32Kpalabras de 16 bits, y una memoria caché de 4Kpalabras, dividida en 4 particiones/conjunto, con 64 palabras/partición. Se supone que inicialmente la memoria caché está vacía. La CPU lee de forma consecutiva el contenido de las posiciones de memoria 0,1,2,...,4351. A continuación, repite 9 veces más esta secuencia de lectura. La memoria caché es 10 veces más rápida que la memoria principal. Estimar la mejora que se obtiene por la utilización de la memoria caché. Para el reemplazamiento de bloques en la memoria caché se emplea una estrategia LRU. Suponer que cada vez que se transfiere un bloque desde memoria principal, la palabra que origina el fallo se envía directamente a la CPU, sin necesidad de ser leída desde la memoria caché.

Solución:

Como la memoria tiene 32K, las direcciones tienen un tamaño total de $log_2 32K = 15$bits, además las palabras se agrupan en bloques de 64 resultando $\frac{32K}{64} = 512$Bloques, y en la memoria caché tenemos $\frac{4K}{4*64} = 16$ conjuntos que hacen falta 4 bits para direccionarlos y para direccionar la palabra dentro, se necesitan 6 bits, puesto que las particiones son de 64 palabras. con lo cual por eliminación, para la etiqueta necesitamos $15-6-4 = 5$bits.

Definimos:

- t_c- tiempo de acceso por palabra a caché=T
- t_m- tiempo de acceso por palabra a memoria principal=10T

1. Sin memoria caché tendríamos lo siguiente:
   $t_m = 4352 palabras * 10 * 10T = 435,200T$
2. Usando caché:

   Las direcciones corresponden a los bloques 0..67, solo hay 64, luego habrá que hacer reemplazos.

   El primer acceso al bloque 0 produce un fallo. Se trae el bloque completo a la caché, y por tanto los 63 accesos restantes no producen fallo. Cuando se accede a la dirección 64 hay un nuevo fallo y se trae otro bloque a la caché.

   Al llegar el bloque 64, se corresponde con el bloque 0 de la caché que está ocupado por los bloques 0,16,32 y 48. Hay que reemplazar un bloque, que por LRU será el bloque 0 quedando ahora los bloques 64,16,32,48. Lo mismo sucede con los bloques 65,66 y 67.

   En el segundo ciclo de lecturas, al volver a referenciar a la dirección 0 se produce un nuevo fallo y el bloque 0, sustituirá al 16.

   a) Primer ciclo de lectura:
   $T_{fallo} 4352 palabras * 10T palabras = 43520T$
   Hay que descontar 68 accesos, pues la primera palabra de cada bloque va directamente a MP.
   $T_{cache} = (4352 - 68) * T = 4284T$
   $T_1 = T_{fallos} + T_{cache} = 43520T + 4248T = 47804T$

   b) Segundo ciclo de lectura:
   En este caso sólo se producen 5 fallos en 4 conjuntos, cada uno de los cuales debe leer 64 palabras
   $T_{fallo} = 5 * 4 * 64 * 10T = 12800T$
   $T_{cache} = 4352 * T - 5 * 4 * T = 4332T$
   $T_2 = T_{fallos} + T_{cache} = 17132T$

   c) Tercer al décimo ciclos:
   Son idénticos al segundo ciclo.
   $T_{cache} = 47804T + 9 * 17132T = 201992T$
   $Mejora = \frac{T_{mp} - T_{cache}}{T_{mp}} * 100 = 53,6\,\%$

5.13. Problema 13

Considere el siguiente programa:

```
main ( ) {
int a [128], b[128], c[128], i;
  for (i=0; i<128; i++) {
    a [i]=b [i]+c [i];
  }
}
```

Suponga que el programa se ejecuta en un sistema con 32 KB de caché de datos y 16 KB de caché de instrucciones. Cada caché es de asignación directa y tiene líneas de 128 bytes y los números enteros se representan mediante 32 bits (4 bytes) en el computador utilizado.

1. ¿Puede que la selección de donde ubicar las instrucciones del programa en memoria afecte a la tasa de aciertos de ambas cachés? (Suponga que el programa es el único que se ejecuta en el computador, despreciando las operaciones del sistema.)
2. Suponga que la asociatividad de la caché se incrementa. ¿Cuál es la mayor asociatividad que puede tener la caché en la que aún existan fallos debido a conflicto?
3. Suponga que las cachés separadas de datos e instrucciones se reemplazan por una caché unificada de 48 KB. ¿Cómo se modifican las respuestas a los apartados a) y b)?

Solución:

1. Gracias a la existencia de cachés separadas, la ubicación del programa no afectará, en general, a la tasa de aciertos, ya que la asignación directa asegura que cada instrucción tiene una ubicación libre en la caché (sin contar el S.O.).

   Cada array necesita $128 * 4 = 512 Bytes$ de memoria, lo que implica un tamaño total de 1,5Kb. Esto es menor que el tamaño de la memoria, por lo tanto se puede ubicar sin generar conflictos.

   Una caché de 32 KB con líneas de 128 bytes tiene :

   N^o líneas $= \frac{32KB}{128} = \frac{2^{15}}{2^7} = 2^8 = 256$ líneas

   Por tanto los bits 0-6 de la dirección se usarán para acceder a una palabra dentro de una línea y los bits 7-14 para determinar la línea asignada. Por lo tanto para eliminar conflictos, se necesita asegurar que las direcciones de los arrays tiene los bits del 7 al 14 distintos

2. Nos ponemos en el caso peor que sería que los 3 arrays se asignen a la misma dirección. Con asociatividad 2 sería el peor caso, ya que con asociatividad 4 cada array puede ir a un conjunto o vía del mismo conjunto.

3. En este caso el código y los datos pueden entrar en conflicto.

   a) Sí puede haber conflictos

   b) Asociativa por conjuntos de 4 vías.

5.14. Problema 14

Suponga que se diseña un sistema de memoria con dispositivos que tienen una latencia de 10 ns y sin retardo de precarga. ¿Cuántos bancos necesitará el sistema para conseguir un valor pico del ancho de banda de al menos 1,5 · 1010 bytes/s, si cada banco transfiere 4 bytes por acceso?

Solución:

Calculamos el ancho de banda de cada banco y dividimos el ancho total por ese valor.
$B_{banco} = \frac{1}{10*10^{-9}} * 4 = 4 * 10^8 Bytes/seg$
$NBancos = \frac{1,25*10^{10}}{4*10^8} = 38$Bancos

5.15. Problema 15

Para este problema suponga un sistema de memoria con dos bancos, uno que se encarga de las palabras con dirección par y el otro de las que tienen dirección impar. Suponga que ambos bancos tienen conexiones independientes con el procesador, por lo que no habrá conflictos con el bus de memoria, y que el procesador puede ejecutar hasta dos operaciones de memoria en cada ciclo, pero que las operaciones de memoria deben ser ejecutadas en orden. También, suponga para simplificar que la latencia de cada banco de memoria es de un ciclo del procesador, por lo que los bancos nunca estarán ocupados gestionando peticiones de ciclos anteriores. Finalmente, suponga que el procesador dispone siempre de dos operaciones de memoria que deben ser ejecutadas en cada ciclo.

1. ¿Cuál es el valor pico del rendimiento del sistema de memoria (en operaciones por ciclo)?
2. Si la dirección de cada petición de memoria es aleatoria (una hipótesis bastante poco realista), ¿cuántas operaciones de memoria, en media, será capaz de ejecutar el procesador en cada ciclo?
3. Si cada banco de memoria devuelve 8 bytes de datos por petición y los ciclos del procesador son de 10 ns, ¿cuál será el valor pico del ancho de banda (en bytes/s) de este sistema de memoria y cuál será el ancho de banda que conseguirá en media?

Solución:

1. Se pueden realizar 2 operaciones por ciclo, principalmente una por cada banco dada la independencia entre ellos que presentan.
2. En media tenemos:

   a) 1 operación siempre se puede resolver.

   b) el 50 % de las veces, además se podrá realizar una segunda operación.

   Lo cual resulta en 1,5 operaciones/ciclo

3. $B = \frac{1}{10*10^{-8}} = 8 * 10^8 Bytes/seg * 2 = 1{,}600{,}000{,}000 bytes/seg$

5.16. Problema 16

Un procesador trabaja con direcciones virtuales de 32 bits, direcciones físicas de 28 bits y 2 KB de tamaño de página. ¿Cuántos bits se necesitan para determinar el número de página virtual y el número de página física?

Solución:

Como el tamaño de página es de 2 KB para especificar el desplazamiento dentro de ésta necesitamos
$log_2 2048 = 11$ bits.

Entonces, si la dirección es de 32 bits tendremos que el número de página virtual viene dado por el resto de bits, $32 - 11 = 21$bits, y la página física es $28 - 11 = 17$ bits

5.17. Problema 17

Un sistema es capaz de gestionar direcciones virtuales de 48 bits, direcciones físicas de 36 bits y tiene 128 MB de memoria principal. Si el sistema utiliza páginas de 4096 bytes ¿Cuántas páginas virtuales y físicas puede gestionas el espacio de direcciones? ¿Cuántos marcos de página hay en la memoria principal?

Solución:

Al tener cada página 4096 bytes, necesitaremos para el desplazamiento dentro de esta $log_2 4096 = 12$bits
Una vez conocido el desplazamiento, si la dirección virtual es de 48 bits las páginas serán direccionadas con $48 - 12 = 36$bits lo cual da lugar a 2^{36}páginas virtuales.
Para marcos, al tener 36 bits sus direcciones. tendremos en total $36 - 12 = 24$bits
El número de marcos de página viene vado por $\frac{128MB}{4KB} = 32{,}768 = 2^{15}$Marcos.

5.18. Problema 18

Una arquitectura cuyas direcciones físicas y virtuales son de 32 bits y las páginas tienen un tamaño de 4 KB ¿Cuál es la dirección física que corresponde a cada una de las siguientes direcciones virtuales?

1. 0x22433007
2. 0x13385ABC
3. 0xABC89001

Número de Página Virtual	Número de Página Física
0xABC89	0x97887
0x13385	0x99910
0x22433	0x00001
0x54483	0x1A8C2

Solución:

1. 0x22433007

   Los 12 bits menos significativos corresponden al desplazamiento 007 y los 20 superiores 0x22433 a la página virtual, observando la TP a esta página virtual le corresponde, la dirección física 00001, entonces, concatenando dirección física y desplazamiento se obtiene 0x00001007

2. 0x13385ABC

   Si tomamos los 12 bits inferiores, tenemos un desplazamiento ABC, y una página virtual 0x13385 que en la TP es la página física 0x99910, con lo cual la dirección resultante es 0x99910ABC

3. 0xABC89001

   Siguiendo el mismo método, el desplazamiento en esta dirección es 001 y la página 0xABC89, que traducida mediante la TP nos da de dirección física 0x97887001

5.19. Problema 19

Sea un sistema con direcciones virtuales de 32 bits, direcciones físicas de 24 bits y páginas de 2 KB.

1. ¿Cuál es el tamaño de cada elemento de la página si se utiliza una tabla de páginas de un nivel?
2. ¿Cuántos elementos de tabla de páginas se necesitan?
3. ¿Cuánta memoria se necesita para la tabla de páginas?

Solución:

1. Si la página tiene un tamaño de 2 KB, tenemos $log_2 2048 = 11$bits de desplazamiento, y $24 - 11 = 13$ bits para el número de página física por tanto el total es de:

   13+bit de validez +bit modificación=15 bits

2. Como las páginas tienen un offset de 11 bits, para saber cuantos elementos son necesarios simplemente basta con el calculo $2^{32-11=21}$páginas.

3. Con los datos anteriores, se puede deducir que la tabla de páginas ocupará aproximadamente $2bytes * 2^{21} = 2^{22}$ que son 4MB de tamaño en memoria.

5.20. Problema 20

El TLB de un procesador requiere 2,2 ns para traducir una dirección en caso de acierto en la caché. El tiempo de acceso a las etiquetas de la caché es de 2,5 ns, 1,0 requiere la lógica de acierto/fallo, y el acceso a los datos necesita 3,4 ns. Finalmente se necesitan 0,5 ns para devolver el dato al procesador en caso de acierto. ¿Cuál es el tiempo de acierto en la caché cuando existe acierto de TLB si la caché tiene direccionamiento y etiquetado virtual? ¿Y si tiene direccionamiento virtual y etiquetado físico? ¿y direccionamiento físico y etiquetado físico?.

Solución:

1. Al producirse acierto en caché no es necesaria la traducción, por tanto, el camino crítico lo marca la caché siendo de:

   $3,5 + 0,5 = 4ns$

2. En este caso vendría dado por, Tiempo Acierto/Fallo + max(búsqueda de etiqueta, tiempo de traducción en TLB) + devolver los datos. que es:

   $2,5 + 1 + 0,5 = 4ns$

3. Con esta situación habría que completar la traducción antes de acceder a las etiquetas teniendo: 2,2 (TLB) +2'5(etiqueta)+1(A/F)+0,5=6,2 ns.

5.21. Problema 21

Un computador tiene una memoria principal de 1 MB y una memoria caché con 16 marcos de 64 bytes cada uno.

1. Determinar el formato de la dirección física desde el punto de vista del emplazamiento en memoria caché, especificando el número de bits que ocupa cada uno de los campos, para cada uno de estos casos:

   a) Caché directa.

   b) Caché asociativa.

   c) Caché asociativa por conjunto con 4 marcos por conjunto.

2. Calcular el número de bits necesarios para implementar la memoria caché en cada uno de estos casos (incluyendo datos y etiquetas).

Solución:

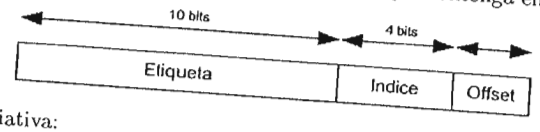
Como la memoria caché tiene $16 = 2^4$ marcos, hace falta 4 bits para indicar el marco. Además sabemos que en la memoria principal se puede dividir en:

$$\frac{1MB}{64B} \quad \frac{2^{20}}{2^6} = 2^{14} bloques$$

Por lo tanto con esta información, podemos deducir el formato de la dirección física en cada uno de los casos.

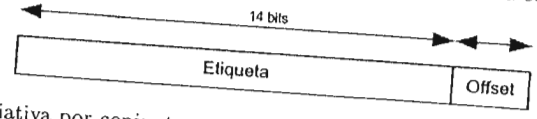
- Caché directa:

  Para la caché de mapeo directo utilizamos 4 bits para seleccionar el marco donde se va a colocar el bloque y 10 bit de etiqueta. Los bits menos significativos de la dirección se utilizarán para seleccionar la palabra dentro del bloque y el número de bits necesario dependerá del número de palabras que contenga en bloque:



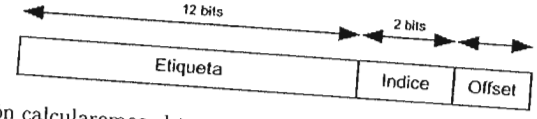
- Caché asociativa:

  En la caché completamente asociativa el bloque puede ir en cualquier marco, por lo tanto no se necesita índice y los 14 bits se corresponden con la etiqueta.



- Caché asociativa por conjuntos:

  En una caché de este tipo el bloque puede colocarse en cualquier marco dentro del conjunto que le corresponde. Como tenemos una caché de 4 marcos por conjunto, necesitamos 2 bits para seleccionar el conjunto en el que vamos a colocar el bloque y los restantes 12 bits de etiqueta.



A continuación calcularemos el tamaño total necesario para implementar cada una de estas cachés, incluyendo memoria de datos y etiquetas:

	Datos	Etiquetas	Total
Mapeo Directo	16*64B=1024B	16*10=160bits	1044B
Asociativa	1024B	16*14=224 bits	1052B
Asociativa por conjuntos	1024B	16*12=192 bits	1048B

5.22. Problema 22

Un computador tiene una memoria principal de 32Kpalabras de 16 bits y una memoria caché de 4Kpalabras dividida en conjuntos de 4 bloques por conjunto y 64 palabras por bloque. Se asume que inicialmente la memoria está vacía y el algoritmo de reemplazamiento es LRU. La CPU lee de forma consecutiva el contenido de las direcciones 0-4351, repitiendo este patrón de acceso 10 veces. La memoria caché es 10 veces más rápida que la memoria principal. Estimar el factor de mejora obtenido al utilizar la memoria caché. Suponer que cada vez que se transfiere un bloque desde MP a MC la palabra que origina el fallo se envía directamente a la CPU, sin necesidad de leerla de la MC. Y que el tiempo que se tarda en enviar un bloque de MP a MC es inferior que la latencia de MP.

Solución:

Para calcular el factor de mejora tenemos que calcular cuanto tarda el programa en ejecutarse sin memoria caché y con memoria caché.

Sin memoria caché el programa realiza 10 veces un bucle que accede a las posiciones 0-4351, y el tiempo que se tarda en realizar cada uno de esos accesos es 10 veces el tiempo de acceder a la caché. Por lo tanto si llamamos t al tiempo de acceso de la caché, el tiempo que se tarda en ejecutar el programa sin caché es:

$$t_{sincache} = 10 * (4352) * 10t = 435200t$$

Si utilizamos caché debemos calcular el número de fallos que se produce al ejecutar el programa. Y con esa información:

$$t_{concache} = naciertos * t_{acccesoMC} + nfallos * t_{accesoMP}$$

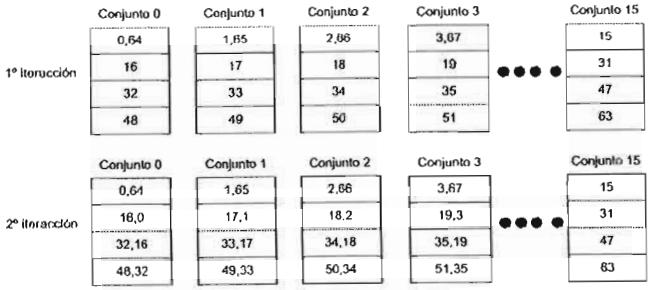
De estos datos podemos extraer la organización de la caché.

Si la memoria principal es de $32Kpalabras = 2^{15}palabras$ que dividido por el número de palabras por bloque $64palabras/bloque = 2^6$ nos da una memoria principal dividida en 2^9 bloques.

La caché tiene 4 bloques por conjunto, 64 palabras por bloque y 4Kpalabras de tamaño, por lo tanto, $\frac{2^{12}palabras}{2^6palabras/bloque} = 2^6bloques$ y junto con el número de bloques por conjunto obtenemos que $\frac{2^6bloques}{2^2bloques/conjunto} = 2^4conjuntos$.

Por lo tanto la caché está dividida en 16 conjuntos de 4 bloques de tamaño.

Los accesos a memoria al ser secuenciales originarán la transferencia de un bloque por cada 64 accesos , existiendo 4351/64=67 bloques que deberán ser mapeados en los conjuntos de la memoria caché.



El bloque 1 se mapeará en el conjunto 0, el bloque 2 en el conjunto 1, el bloque 3 en el conjunto 2 y asi sucesivamente hasta el bloque 63, que al ser cargado en el conjunto 15 llena completamente la memoria caché, por lo tanto los bloques 64-67 sustituirán a los bloques 0-3 y en primera iteración del bucle se originará un fallo por cada bloque.

En la segunda iteración se volverá a cargar los bloques 0-3, ya que han sido reemplazados y estos sustituirán a los bloques 16-19 los cuales originarán nuevos fallos en esta iteración. Siguiendo esta pauta los bloques 16-19 al haber sido reemplazados deberán traerse de nuevo y reemplazarán a los bloques 32-35. Lo mismo ocurre con los bloques 48-51 y 65-68.

De esta forma para cada una de las 10 iteraciones podemos calcular el número de fallos como, los fallos de la primera iteración, que al estar vacía la caché serán 68 más los fallos en cada una de las iteraciones, que serán los originados al acceder a los bloques 0-3, 16-19, 32-35 y 48-51.

$$nfallos = 68 + 15fallos * 9iteraciones = 203$$

El tiempo total será por tanto, el número de aciertos que es el número de accesos menos el número de fallos más

$$t_{concache} = (43520 - 203) * t + 203 * 10t = 45347t$$

Por lo tanto el factor de mejora será de:

$$S = \frac{435200t}{45347t} = 9{,}59$$

5.23. Problema 23

Un sistema tiene una memoria principal de 16Kpalabras y una memoria caché de 4Kpalabras dividida en conjuntos de 4 bloques y 64 palabras por bloque. Se asume que la caché está inicialmente vacía y la política de reemplazamiento es LRU. El procesador accede secuencialmente a las direcciones desde la 0 a la 2400, repitiendo este patrón de acceso N veces. La memoria caché es M veces más rápida que la principal. Estimar el factor de mejora en función de N y M al usar la memoria caché realizando las mismas suposiciones que en el ejercicio 2.

Solución:

Este ejercicio es muy similar al anterior y para resolverlo procederemos de igual forma. Primeramente calcularemos el tiempo de acceso en el caso de que el sistema no disponga de memoria caché. El procesador accede secuencialmente a las posiciones 0-2400 N veces, lo que significa que realiza 2401*N accesos. Si llamamos t al tiempo de acceso de la memoria caché, como la memoria principal es M veces más lenta, su tiempo de acceso será Mt. Con estos datos:

$$t_{sincache} = N * 2401 * M * t$$

En el caso de utilizar memoria caché debemos calcular cual es su estructura al igual que en el ejercicio anterior. La memoria es de 4Kpalabras que dividido por el número de palabras por bloque, que es de 64, nos da $\frac{2^{12}palabras}{2^6palabras/bloque} = 2^6bloques$. Si cada conjunto tiene 4 bloques el número de conjuntos será de $\frac{2^6bloques}{2^2bloques/conjunto} = 2^4conjuntos$. Por lo tanto la caché es de 16 conjuntos de 4 bloques cada uno.

Calculamos ahora en que bloque se encuentra la última referencia del programa para ver su secuencia de accesos, teniendo en cuenta que cada bloque tiene 64 palabras, la palabra 2400 se encuentra en el bloque 2400/64=37. Si el programa ocupa 38 bloques y en la caché caben 64, el programa cabe entero en la caché y los únicos fallos que tenemos son los iniciales, y las siguientes N-1 iteraciones acertarán en su acceso a la caché.

$$t_{concache} = 38 * Mt + 2401 * (N-1) * t$$

Por lo tanto la mejora será de:

$$S = \frac{t_{sincache}}{t_{concache}} = \frac{2401NM}{2401(N-1) + 38M}$$

Si M< <N entonces la mejora $S \simeq M$.

5.24. Problema 24

Se tienen dos computadores con el mismo procesador y la misma jerarquía de memoria pero con diferentes organizaciones de la memoria caché.

- Computador A: Asociativa por conjuntos, 128 conjuntos, 2 bloques por conjunto, bloque de 32 bytes, escritura directa sin asignación en escritura.
- Computador B: Asignación directa, 256 bloques, de 32 bytes. Post-escritura con asignación en escritura.

En los dos casos la penalización por fallo es 10 veces el tiempo de acierto y escribir una palabra de 32 bit en MP es 5 veces el tiempo de acierto (escritura directa). Además la caché es unificada para instrucciones y datos.

1. Describir un programa que hace al computador A mucho más rápido que el B.
2. Describir un programa que hace al computador B mucho más rápido que al A.
3. ¿Cuánto más rápido es el programa de a) en el computador A que en B?
4. ¿Cuánto más rápido es el programa de b) en el computador B que en A?

Solución:

En este ejercicio se nos pide una reflexión sobre las dos políticas de escritura en memorias caché, escritura directa y post-escritura. Cada uno de los dos computadores posee uno de los esquemas y se nos pide describir un programa que en cada uno de los casos ofrezca un mejor rendimiento con ese esquema de escritura.

En el caso del computador A que posee una caché con escritura directa, sus cachés ofrecerán mejor rendimiento cuando el programa realice muchas escrituras en posiciones de memoria dispersas, que no pertenezcan al mismo bloque de caché. En este caso la escritura directa no originará reemplazos ya que no tiene asignación en escritura.

En el caso 2, un computador que realice múltiples escrituras sobre las mismas posiciones o posiciones dentro de los mismos bloques de caché tendrá un rendimiento elevado ya que estas escrituras serán tratadas como aciertos en caché y no serán reenviadas a memoria principal hasta que ocurra un reemplazo.

En el caso 1 el computador A tendrá para cada acceso que escribir la palabra en memoria principal, lo cual se nos dice que tarda 5t, mientras que el computador B originará un fallo de caché para cada una de estas escrituras, tardando en resolverlo el tiempo de acierto más la penalización que se nos dice que es 10 veces el tiempo de acierto, por lo tanto t+10t=11t. Por lo tanto la diferencia entre los dos computadores en este caso será de 11t/5t=2.2t

En el segundo caso, para el programa que realiza accesos consecutivos el computador A seguirá tardando 5t en cada acceso ya que tiene que hacer la escritura en memoria principal, mientras que el computador B al tener los bloques en memoria caché tardará únicamente t en realizar el acceso sin ninguna penalización. La deferencia en este caso es de 5t/t=5 veces más rápido el computador B en ejecutar este programa.

5.25. Problema 25

Se dispone de un computador con las siguientes características:

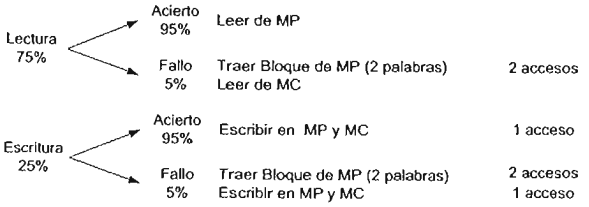
- El 95 % de los accesos se encuentran en la caché.
- Cada bloque de caché es de dos palabras y se lee completo en caso de fallo.
- El procesador envía peticiones a la caché a razón de 10^9 por segundo.
- El 25 % de los accesos son escrituras.
- El sistema de memoria admite 10^9 accesos por segundo en lectura o escritura.
- El bus permite leer o escribir una palabra por ciclo.
- El 30 % de los bloques de caché son modificados en el tiempo que permanecen en la caché. - Se utiliza escritura asignada.

Se desea añadir un periférico al sistema y queremos conocer el porcentaje de ancho de banda de memoria que ya está utilizado en cada uno de los siguientes casos:

1. Caché de escritura directa.
2. Caché de post-escritura.

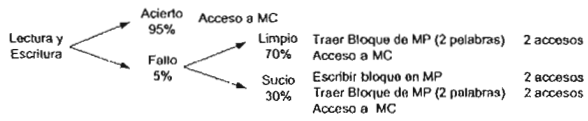
Solución:

Primeramente debemos calcular cuantos accesos se hacen a memoria principal en cada uno de los casos, para seguidamente poder calcular el ancho de banda. En el primer caso, con una caché de escritura directa:



$accesos/s = peticiones/s * (2 * \%FallosLectura + \%AciertosEscritura + +3 * \%FallosEscritura) = 10^9 * (2 * 0{,}75 * 0{,}05 + 0{,}25 * 0{,}95 + 3 * 0{,}25 * 0{,}05) = = 0{,}35 * 10^9$

En el caso de una caché de post-escritura, podemos resumir el comportamiento de la caché de datos con este otro diagrama:



$accesos/s = peticiones/s * (2 * \%FallosLimpios + 4 * \%FallosSucios) = = 10^9 * (2 * 0{,}05 * 0{,}7 + 4 * 0{,}05 * 0{,}3) = 0{,}13 * 10^9$

5.26. Problema 26

Un computador con memoria caché dividida tiene las siguientes características:

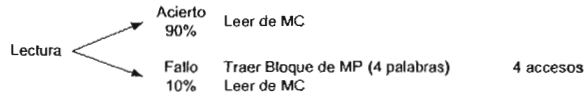
- El 90 % de todas las referencias a instrucciones generadas por la CPU se encuentran en la caché.
- El 80 % de todas las referencias a datos generadas por la CPU están en la caché.
- El bloque de la caché de instrucciones es de 4 palabras y el de datos es de 2 palabras.
- La CPU genera 10^7 referencias por segundo a instrucciones y $2 * 10^6$ referencias por segundo a datos. El 20 % de estas referencias son escrituras y el 25 % de los bloques en la memoria caché de datos son modificados durante su permanencia en ella.

Calcular el ancho de banda mínimo necesario entre memoria principal y memoria caché en los siguientes casos:

1. Con política de escritura directa.
2. Con política de post-escritura

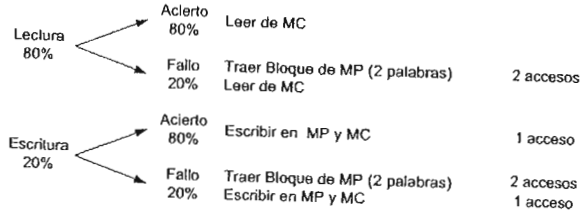
Solución:

Para resolver este ejercicio primero calculamos el ancho de banda que consume la caché de instrucciones que es independiente de la política de escritura que se utilice:



$accesos/s = instr/s * 4 * \%FallosLectura = 10^7 * 4 * 0{,}1 = 0{,}4 * 10^7 palabras/s$

En el caso de que la caché sea de escritura directa, debemos distinguir entre el comportamiento de lecturas y escrituras.

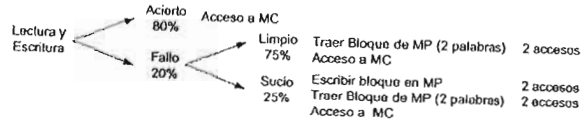


$accesos/s = peticiones/s * (2 * \%FallosLectura + \%AciertosEscritura + +3 * \%FallosEscritura) = 2 * 10^6 * (2 * 0{,}8 * 0{,}2 + 0{,}2 * 0{,}8 + 3 * 0{,}2 * 0{,}2) = = 1{,}2 * 10^6 palabras/s$

Sumando los dos resultados obtenemos el ancho de banda consumido en el primer caso:

$$AnchoBanda = instrucciones + datos = 0{,}4 * 10^7 + 1{,}2 * 10^6 = 5{,}2 * 10^6 palabras/s$$

Cuando utilizamos una caché de post-escritura no debemos distinguir entre accesos de lectura y escritura. En ese caso el ancho de banda necesario será de:



$accesos/s = peticiones/s * (2 * \%FallosLimpio + 4 * \%FallosSucios) = = 2 * 10^6 * (2 * 0{,}2 * 0{,}75 + 4 * 0{,}2 * 0{,}25) = 10^6 palabras/s$

En este caso el ancho de banda consumido es:

$$AnchoBanda = instrucciones + datos = 0{,}4 * 10^7 + 1 * 10^6 = 5 * 10^6 palabras/s$$

Con lo que se ve que se consume menos ancho de banda utilizando una caché de datos con post-escritura.

5.27. Problema 27

Un computador tiene un sistema de memoria compuesto por una caché con un tamaño de bloque de 1 palabra y una tasa de fallos del 3 %. Se realizan 1.2 accesos a memoria por instrucción y la memoria principal tiene un ancho de bus de 1 palabra. Para llevar una palabra de la MC a MP se necesitan 4 ciclos para enviar la dirección, 56 ciclos de tiempo de acceso y 4 ciclos para enviar una palabra de datos. Se estudia modificar el tamaño del bloque a 2 palabras, con una tasa de fallo de 2 %, ó a 4 palabras con tasa de fallo de 1.2 %. ¿Cuál es la mejora en cada caso usando o no entrelazado?

Solución:

En el caso original en el que la caché tiene un tamaño de bloque de 1 palabra podemos calcular el CPI del sistema como el CPI original (suponemos que es 1), más la penalización media que introduce el subsistema de memoria. En nuestro caso será de:
$CPI = CPI_{original} + AccesosaMem/instruccin * \%Fallos * PenalizacionFallo =$
$= 1 + 1{,}2 * 0{,}03 * (4 + 56 + 4) = 3{,}3$
Si aumentamos el tamaño del bloque a 2 palabras distinguimos los siguientes casos:

- Misma jerarquía de memoria, es igual salvo que la penalización es el doble por transferir dos palabras en lugar de una y la tasa de fallos disminuye a 0.02 como se nos dice en el enunciado.

$$CPI = 1 + 1{,}2 * 0{,}02 * 2 * 64 = 4{,}07$$

- Ensanchando la memoria a dos palabras. El acceso a memoria nos entrega las dos palabras de golpe, con lo cual sólo tenemos que hacer un acceso.

$$CPI = 1 + 1{,}2 * 0{,}02 * 64 = 2{,}54$$

- Usando dos bancos de memoria entrelazada, lanzamos la dirección a la vez y el tiempo de acceso en los dos banco se solapa, incrementándose el tiempo total sólo en transferir las dos palabras secuencialmente.

$$CPI = 1 + 1{,}2 * 0{,}02 * (4 + 56 + 8) = 2{,}63$$

Si aumentamos el tamaño del bloque a 4 palabras distinguimos los mismos casos:

- Misma jerarquía de memoria.

$$CPI = 1 + 1{,}2 * 0{,}012 * 4 * 64 = 4{,}69$$

- Ensanchando la memoria a cuatro palabras.

$$CPI = 1 + 1{,}2 * 0{,}012 * 64 = 1{,}92$$

- Usando cuatro bancos de memoria entrelazada.

$$CPI = 1 + 1{,}2 * 0{,}012 * (4 + 56 + 16) = 2{,}09$$

Capítulo 6

Sistema de E/S

6.1. Problema 1

Calcular el ancho de banda de un bus de 32 bits con una frecuencia de trabajo de 200 MHz y una latencia inicial entre operaciones de 5 ns.

Solución:

Atendiendo a la definición de ancho de banda como el num. de bytes transferidos por unidad de tiempo, podemos calcular B como sigue: $B = \frac{Nbytes}{T.transferencia}$. En este caso, como el bus es de 32 bits a 200MHz, se deduce que transferimos 4 bytes en un tiempo de , más la latencia entre operaciones de 5 ns. Resultando en un total de $B = \frac{4}{10ns} = 4 * 10^8 B/s$.

6.2. Problema 2

Se desea comparar los anchos de banda máximos de un bus síncrono y otro asíncrono. El bus síncrono tiene un tiempo de ciclo de reloj de 50 ns, y cada transacción del bus requiere un ciclo de reloj. El bus asíncrono requiere 40 ns para el protocolo de transferencia. Ambos buses tienen una anchura de 32 bits. Determinar el ancho de banda de ambos buses cuando realizan lecturas de una memoria de 200 ns. Suponer que las lecturas son siempre de una palabra.

Solución:

Bus Síncrono: Un acceso a memoria se divide en los siguientes tiempos:

- Enviar dirección: 50 ns.
- Leer Memoria: 200 ns.
- Recibir Datos : 50 ns.

Por lo tanto para una operación completa de lectura, son necesarios 300 ns, y el ancho de banda total viene dado por :
$B = \frac{4Bytes}{300ns} = 13{,}3MB/s.$

Bus Asíncrono: Para un acceso a memoria con un protocolo asíncrono hacen falta señales de control como MSyn y SSyn.

- Enviar dirección y solicitar lectura (Read + MSyn): 40 ns.
- Operación de lectura: 200 ns.
- Leer Datos y finalizar el protocolo
  - Desactivar MSyn y data lines: 40 ns.
  - Desactivar SSyn: 40 ns.
  - Desactivar Read y lineas de dirección: 40 ns.

$40 * 3 = 120$ns.

En total el protocolo tarda 360 ns, lo que implica $B = \frac{4Bytes}{360ns} = 11,1MB/s$.

6.3. Problema 3

Se dispone de un sistema de memoria y bus que soporta acceso a bloques de 4 y 16 palabras de 32 bits, un bus síncrono de 64 bits a 200Mhz, en el que una transacción de datos o direcciones requiere un ciclo de reloj. Se necesitan dos ciclos de reloj entre dos operaciones de bus. El tiempo de acceso a memoria para las cuatro primeras palabras es de 200 ns y cada grupo adicional de 4 palabras se lee en 20 ns y la transferencia de los datos puede solaparse con la lectura de las cuatro palabras siguientes.

1. Determinar el ancho de banda mantenido y la latencia para la lectura de 256 palabras con transferencias de bloque de 4 y 16 palabras.
2. Calcular el número de transacciones de bus por segundo en cada uno de los casos.

Solución:

1. Memoria/Bus: Bloques de 4 y 16 palabras de 32 bits. Bus de 64 bits a 200 MHz, 1 ciclo de transferencia y 2 entre operaciones. Acceso a memoria de 200 ns para el primer bloque de 4 palabras y 20 ns para el resto.

   a) Caso de 4 palabras:
   - Enviar dirección : 5 ns.
   - Lectura de datos: 200 ns.
   - Enviar datos desde memoria: 2 ciclos por ser 4 palabras de 32 bits y un bus de 64 bits : 10 ns.
   - Tiempo de espera: 2 ciclos = 10 ns.

   El bloque es de 256 palabras, luego 256/4 = 64Transacciones. Por tanto la latencia es $64*225ns = 14{,}400ns$. y el ancho de banda $B = \frac{256*4}{14{,}400} = 71,11MB/s$.

   b) Caso de 16 palabras:
   Ahora todo será igual en el acceso a memoria salvo que la Lectura de datos son $T_{bloque} = 5+200+max(20,10+10)*4 = 285$. el máximo se utiliza puesto que el envio y la lectura de bloques se realizan simultáneamente, son actividades que se solapan en el tiempo. La latencia resultará en $L = 285*16bloques = 4{,}560$ns. y como $Nbloques = \frac{256palabras}{16pal./bloque} = 16$implica $B = \frac{256*4}{4560*10^{-9}} = 224,56MB/s$.

2. Definimos Transacción como la operación completa de Bus que comprende el envío de una dirección, más un conjunto de datos (el conjunto de datos puede requerir varios accesos al bus). Por tanto, las transacciones por segundo que suceden son:

   a) 4 palabras: $NTrans = \frac{64Trans}{14{,}400} = 4,44 * 10^6 Trans/s$.

   b) 16 Palabras: $NTrans = \frac{16Trans}{4{,}560} = 3,51 * 10^6 Trans/s$.

6.4. Problema 4

El sistema descrito en el ejercicio anterior se aplica ahora para tratar los accesos a un disco con una velocidad de transferencia de 5MB/s. Si se permite que la E/S consuma el 100 % del ancho de banda del bus y del sistema de memoria, ¿Cuál es el máximo número de transferencias de disco simultáneas que pueden mantenerse para cada uno de los dos tamaños?

Solución:

- 4 palabras:

  B=71,11 MB/s, Disco=5 MB/s.

  Con estos datos podemos deducir el número de discos que soportará el sistema, asignando una parte del ancho de banda del bus a cada uno de ellos siendo $\frac{71,11}{5} = 14,22$ con lo cual garantizamos el funcionamiento de 14 discos.

- 16 palabras:

  B=224,51 MB/s.

  Aplicando el mismo razonamiento de antes, ahora el número de discos soportados es de $\frac{224,51}{5} = 44$ discos.

6.5. Problema 5

Para el mismo sistema de memoria descrito en el ejercicio 3 se considera ahora utilizar un bus asíncrono. Cada petición de E/S solicita 16 palabras de datos de memoria y el ancho del bus es de 4 palabras. El bus sigue el protocolo descrito con 40 ns para cada paso. Pero en este caso la memoria puede continuar la transacción enviando bloques de datos adicionales hasta que se complete la transacción. Indicar cómo podría llevarse a cabo esta transferencia. Suponiendo que cada paso del protocolo requiere ahora 20 ns y cada acceso a memoria requiere 60 ns ¿cuánto tiempo tardará en completarse la transferencia?¿Cuál es el máximo ancho de banda que puede sostenerse en este bus asíncrono y cómo resulta comparado con el bus síncrono del ejercicio 3?

Solución:

El protocolo a seguir es el siguiente:

1. Periférico activa una solicitud de ráfaga activando Read Req y Bus Req.
2. Cuando la memoria ve BR y RR activas lee la dirección de comienzo del bloque de 16 palabras y activa una señal de reconocimiento.
3. I/O Dispositivo desactiva ReadReq, pero mantiene BR activa.
4. La memoria envía un Ack para reconocer la operación de lectura.
5. Cuando la memoria tiene un bloque de 4 palabras listo las coloca en el bus y activa Data Rdy.
6. El periférico lo lee y activa Ack.
7. Cuando la memoria ve Ack desactiva Data Rdy y las líneas de datos.
8. Cuando el periférico ve que DataRdy baja, quita el Ack pero mantiene BReq, no quedan datos pendientes. Entonces se repite el protocolo desde el paso 4.
9. Cuando se envían las últimas 4 palabras, el periférico desactiva BReq, que indica que la transacción de la ráfaga se ha completado.

Estas transacciones se denominan en modo ráfaga. Para leer 16 bloques en un bus de 4 bytes de ancho necesitamos 4 accesos o envíos por el Bus.

Necesitamos una forma de indicar al periférico y memoria el tamaño de la ráfaga (el número de bytes o envíos consecutivos que hay que hacer). En lugar de introducir un nuevo dispositivo, esto se hace mediante la señal "BusReq" que permanecerá activa todo el tiempo necesario para transferir una ráfaga (que puede variar en tamaño 4,8,16 palabras).

Si al iniciar una lectura BusReq. está a 0 es que se va a hacer una lectura sencilla de una sola palabras.

- Cada paso dura 20 ns y el acceso a memoria 60 ns la transferencia de la ráfaga dura:
  - Paso 2: 20 ns (Memoria recibe dirección).
  - Pasos 3,4,5: max((3*20),60) = 60 ns.
  - Pasos 6,7,8,9: max((4*20),80)=80 ns.
  - Se repiten 6,7,8,9: 80 ns.
  - Se repiten 6,7,8,9: 80 ns.

  (la repetición es debida al número de accesos para transferir 16 bloques). Tenemos un total de 320 ns. de latencia, con lo cual $B = \frac{16*4}{320*10^{-9}} = 200MB/s$.
- Para comparar con el ejercicio 3 necesitamos que la memoria tarde 200 ns. En ese caso la latencia sería: $L = 20+200+200+200+200 = 820$ns. y $B = \frac{16*4}{820} = 78MB/s$. que es mucho menor.

6.6. Problema 6

Suponga un protocolo de bus que requiere de 10 ns para que los dispositivos realicen sus peticiones 15 ns para el arbitraje y 25 ns para completar cada operación. ¿Cuántas operaciones por segundo se pueden realizar?

Solución:

Consideramos que una operación tarda un tiempo de $10 + 15 + 25 = 50$ns. por tanto $\frac{1}{50ns} = 20 * 10^6$operaciones por segundo.

6.7. Problema 7

Un bus SCSI dispone de cuatro dispositivos conectados, con los identificadores 1,2,3,4. Si cada dispositivo desea hacer uso de 30MB/s del ancho de banda del bus y el ancho de banda total de un dispositivo SCSI es de 80MB/s, ¿cuánto ancho de banda será capaz cada dispositivo de utilizar? Asuma que cada vez que a un dispositivo se le deniega el acceso al bus, lo vuelve a intentar tan pronto como el bus vuelve a estar libre y lo sigue intentando hasta que consigue el acceso.

Solución:

Si suponemos un acceso del dispositivo 1 con la mayor prioridad, este toma 30 MB/s, seguidamente el dispositivo 2 tomará otros 30 MB/s, en el siguiente acceso 30MB/s más serán consumidos por el dispositivo 3, dejando 20MB/s al 4 cuando este lo vaya a usar.

6.8. Problema 8

Suponga que el bus del problema anterior empleará una política de arbitraje del bus equitativa, en la que el dispositivo que estuviera esperando el bus más tiempo tuviera la prioridad más alta. ¿De cuánto ancho de banda dispondría cada dispositivo en tal caso?

Solución:

Ahora mismo todos tendrán las mismas oportunidades de acceso al bus, con lo cual este se dividirá entre el número de dispositivos resultando a 20MB/s para cada uno.

6.9. Problema 9

Un procesador determinado dispone de ocho líneas de interrupción (numeradas del 0 al 7) y una política en la que las interrupciones con un número bajo tienen mayor prioridad sobre aquellas de número más alto. El procesador comienza sin interrupciones pendientes y se produce la siguiente secuencia de interrupciones : 4,7,1,3,0,5,6,4,2,1. Asuma que la gestión de una interrupción tarda el tiempo suficiente para que dos nuevas interrupciones se produzcan y que las interrupciones no se pueden interrumpir entre sí. ¿En qué orden se gestionan las interrupciones?

Solución:

Suponemos una cola ordenada de interrupciones pendientes donde se van encolando a medida que se procesan las de un nivel superior, por tanto la 4 según llega se atiende, cuando termina 7 y 1 están disponibles y tiene prioridad 1 por lo que se gestiona esta, teniendo ahora 0,3,7 en la cola, iterando sobre el proceso llegamos a un orden final de 4,1,0,3,2,1,4,5,6,7.

6.10. Problema 10

Un procesador determinado precisa de 1000 ciclos para llevar a cabo un cambio de contexto y comenzar un gestor de interrupción (y el mismo número de ciclos para realizar el cambio de contexto en sentido inverso para el programa que se estaba ejecutando cuando ocurrió la interrupción) o 500 ciclos para sondear un dispositivo de E/S. Un dispositivo de E/S conectado al procesador realiza 150 peticiones por segundo, cada una de las cuales tarda 10.000 ciclos en finalizar una vez que el gestor ha comenzado. Por defecto, el procesador sondea cada 0,5 ms si no está usando interrupciones.

1. ¿Cuántos ciclos por segundo emplea el procesador en gestionar la E/S del dispositivo si se emplean interrupciones?
2. ¿Cuántos ciclos por segundo se emplean E/S si se usa sondeo (incluyendo todos los intentos de sondeo)? Asuma que el procesador tan sólo sondea durante periodos de tiempo en que no se están ejecutando programas de usuario, por lo que no se debe incluir ningún tiempo de cambio de contexto en el cálculo de los costes de sondeo.
3. ¿Con qué frecuencia se debe sondear el procesador para que esta técnica dedique tantos ciclos por segundo como las interrupciones?

Solución:

1. Si son necesarios 1000+10.000+1000 ciclos por interrupción, en total por segundo si llegan 150 peticiones, harán un total de $150 * 12{,}000 = 1{,}800{,}000$ciclos.
2. El sondeo se produce cada 0,5 ms. donde en cada sondeo consumimos 500 ciclos, empleando un total de 2000*500=1.000.000 ciclos por segundo, si hay 150 peticiones y cada una tarda 10.000 ciclos en terminar, resulta en un total de $1{,}000{,}000 + (150 * 10{,}000) = 2{,}500{,}000$.
3. Si sondeamos 2000 veces por segundo consumiendo 500 ciclos y el procesado son 1.500.000 ciclos, para alcanzar a las interrupciones como mucho sólo se pueden perder 300.000 ciclos por segundo en sondeos, por lo tanto utilizando 500 ciclos en sondeo sólo podremos sondear 600 veces por segundo.

6.11. Problema 11

Un dispositivo de E/S transfiere 10MB/s de datos a la memoria de un procesador a través del bus de E/S, el cual tiene un ancho de banda total de 100MB/s. Los 10MB/s de datos se transfieren en 2500 páginas independientes, cada una de ellas ocupa 4 KB.

Si el procesador opera a 200MHz, requiere 1000 ciclos para iniciar una transferencia de DMA y 1500 ciclos para responder a la interrupción del dispositivo cuando finaliza la transferencia, ¿Qué fracción de tiempo de CPU se emplea en gestionar la transferencia con y sin DMA?

Solución:

1. SIN DMA:

   Sabemos que el bus tiene B=100MB/s y el dispositivo en total usa 10MB/s con lo cual se deduce que es el 10 %, el cual además, sin DMA representaría el tiempo de computo de CPU para E/S puesto que es ella la encargada de hacer la transferencia.

2. CON DMA:

   El total de computo de la CPU con DMA son los 1000 ciclos de inicio y 1500 ciclos de final, con lo cual representan 2500 ciclos de 200MHz , además el total de la transferencia son 2500 paginas por lo cual el consumo final es de 6250000 que es aproximadamente $\% = \frac{6{,}250{,}000 * 100 ciclos}{200 * 10^6 Hz} = 3{,}125\,\%$ de un segundo.

6.12. Problema 12

Un controlador de DMA está transmitiendo, mediante robo de ciclos, caracteres a memoria desde un periférico a una velocidad de 19200 bps (bits/seg). Por su parte la CPU realiza la búsqueda de las instrucciones con una velocidad de 2 millones de instrucciones por segundo. ¿En qué porcentaje se reduce la velocidad del procesador debido a la DMA?
Supones palabra de M bits y la CPU emplea N ciclos/instrucción.

Solución:

Con los datos deducibles es fácil deducir que el C.DMA transfiere $\frac{19200}{M} palabras/seg.$ que además es el número de robo de ciclos por la razón de que sólo se roba un ciclo para transferir una palabra. Esto se traduce en dejar de ejecutar instrucciones durante esos ciclos perdidos, con lo que perdemos aproximadamente $\frac{19200}{M*N}$ Instrucciones, entonces como el procesador tiene un MIP de 2 $2 * 10^6 ins./seg.$ con la formula de $\% = \frac{Ins.Perdidas}{Ins.SinPerdida} * 100$ obtenemos un resultado de $\frac{0{,}96}{M*N}$ %de cota máxima puesto que nunca superara eso.

6.13. Problema 13

Se dispone de un sistema formado por un disco duro que transfiere datos en bloques de 4 palabras a una velocidad de 4MB/s y un procesador que funciona a 500MHz. Supones que el disco duro está siempre ocupado. Determinar la proporción de tiempo de CPU que se consume en Entrada/Salida en cada uno de los siguientes casos:

1. Usando E/S programada en la que cada comprobación del dispositivo consume 400 ciclos del reloj.
2. Usando E/S por interrupciones en la que la sobrecarga de cada transferencia incluida la interrupción es de 500 ciclos de reloj.

3. DMA con un tiempo de iniciación de 1000 ciclos, un tiempo de tratamiento de la interrupción de 500 ciclos y un tamaño medio de transferencias de disco de 8KB.

Solución:

1. Hay que determinar el número de veces que se debe sondear al disco. Para eso determinamos el número de bloques que es capaz de transferir por segundo. $\frac{4MB/s}{16B} = 250 * 10^3 Bloques/seg.$ que es además el número de sondeo necesario, sabemos que cada sondeo consume 400 ciclos, luego perderemos $250 * 10^3 * 400 = 100 * 10^6 ciclos$ que con una CPU a 500MHz representa un total de $\frac{100*10^6}{500*10^6} = 20\,\%$ de tiempo de CPU se pierde.

2. La frecuencia de la interrupción es la misma ya que estamos suponiendo un disco siempre ocupado: $CiclosInt = 250 * 10^3 * 500 = 125 * 10^6 ciclos.$ lo cual repercute en tiempo de CPU de la siguiente forma $\frac{125*10^6}{500*10^6} = 25\,\%$ de tiempo perdido.

3. Cada transferencia de DMA requiere: $\frac{8KB}{4MB/s} = 2ms$de transferencia resultando en $\frac{1}{2*10^{-3}} = 500$transferencias/seg suponiendo un flujo de datos desde el disco constante. Cada transferencia. consume 1000 ciclos Inicio y 500 ciclos de interrupción, lo que hace un total de $1500 ciclos * 500 Transferencias/seg. = 750 * 10^3 ciclos/seg$ y el % perdido es $\frac{750*10^3}{500*10^6} = 0,15\,\%$.

6.14. Problema 14

Consideramos un computador formado por los siguientes elementos:

1. CPU que ejecuta 300 millones de instrucciones por segundo y que emplea una media de 500.000 instrucciones de sistema operativo en cada operación de E/S.
2. Un bus de memoria con una velocidad de transferencia de 100 MB/s.
3. Controladores SCSI-2 con velocidad de transferencia de 20MB/s que permiten la conexión de hasta 7 discos duros.
4. Unidades de disco con ancho de banda de lectura/escritura de 5MB/s y un tiempo medio de búsqueda más latencia rotacional de 10 ms.

Suponiendo que la carga de trabajo consiste en lecturas de bloques de 64KB (cada bloque almacenado secuencialmente en una pista) y que un programa de usuario necesita 100.000 instrucciones por cada operación de E/S, calcular la máxima velocidad de E/S que puede mantenerse y el número de discos y controladores SCSI necesarios. Ignorar los conflictos de disco.

Solución:

Dos elementos fijos del sistema son la CPU y el bus de memoria, hay que ver cual es la velocidad de E/S de cada uno de estos elementos para saber cual es el cuello de botella:

- CPU:

  $MaxVelEsCPU = \frac{VelocidadEjecucion}{NInst.} = \frac{300*10^6}{(50+100)*10^3} = 2000$ Instrucciones de E/S por segundo.

- BUS:

  $MaxVelEsBUS = \frac{anchoBandaBus}{BytesE/s} = \frac{100*10^6}{64*10^3} = 1562$ Operaciones de E/S por segundo.

Luego el bus es el cuello de botella. Configuraremos el resto del sistema para que funcione al ritmo que dicta el bus.

$T_{operacionESdisco} = T_{bus} + T_r + T_{acceso} = 10ms + \frac{64KB}{5MB/s} = 10 + 12,8 = 22,8ms.$

Sí cada operación necesita 22,8 ms tenemos $\frac{1}{22,8*10^{-3}} = 43,9$operaciones de E/S por segundo en el disco, entonces, para saturar el bus, son necesarias 1562 con lo que necesitamos $\frac{1562}{43,9} \sim 36 Discos$

Para saber el número de controladores SCSI hace falta conocer la velocidad media de transferencia por disco, siendo esta: $\frac{64KB}{22,8ms} = 2,74MB/s$, en cada controlador SCSI se pueden conectar 7 discos, por lo tanto no saturan el ancho de banda del controlados: 20MB/s, y el número de controladores es $\frac{36}{7} = 6$Controladores.

6.15. Problema 15

Un disco duro de 5 platos contiene 2048 pistas por plato, 1024 sectores por pista y 512 bytes por sector. ¿Cuál es su capacidad total de almacenamiento?

Solución:

La capacidad es el total de bits que puede almacenar, según la topología de un disco duro, sabemos que cada plato contiene pistas, y estas a su vez sectores con lo cual $C = 5 * 2048 * 1024 * 512 = 5{,}368{,}709{,}120 = 5GB$

6.16. Problema 16

Se desea fabricar un disco duro con una capacidad de, al menos 30 GB. Si la tecnología empleada para fabricar los discos permite sectores de 1024 bytes, 2048 sectores por pista y 4096 pistas por plato, ¿cuántos platos se necesitan?

Solución:

Siguiendo el principio de antes ahora buscamos el número de platos que se puede deducir de $30 = NPlatos * 4096 * 2048 * 1024 = 3,75 \sim 4$ Platos son necesarios al menos.

6.17. Problema 17

La pista más interna del plato de un disco duro tiene un radio de 0,25 pulgadas. La pista más externa uno de 1,75 pulgadas. ¿Cuál es la relación entre la capacidad del disco

si se emplea un número variable de pistas por sector con respecto a la capacidad si se emplea un número fijo de pistas por sector?

Solución:

Ambos esquemas tienen el mismo número de pistas por plato, por lo tanto la relación entre las capacidades será la misma que entre el número de sectores medio por pista.

En el esquema de número de sector fijo, este viene determinado por el número de sectores de la pista más interna, y todos tienen la misma cantidad, sin embargo, con sectores variables, todos los sectores ocupan lo mismo y lo que varía es el número de sectores por pista, Esto viene marcado por la longitud de la pista.

Así pues, basta calcular la relación de la longitud de la pista media con la de la pista más interna, la pista media tiene una longitud de : $1,75 - 0,25 = 1,5/2 = 1,0$ por tanto la relación es $1,0/0,25 = 4$, de lo que se deduce que el disco con sectores variables tiene 4 veces más capacidad

6.18. Problema 18

Un disco duro de un plato gira a 15.000 RPM y dispone de 1024 pistas, cada una con 2048 sectores. La cabeza del disco comienza en la pista 0 (las pistas se numeran de la 0 a la 1023). En ese momento el disco recibe una petición para acceder a un sector aleatorio en una pista aleatoria. Si el tiempo de búsqueda de la cabeza del disco es de 1 ms por cada 100 pistas que se recorren:

1. ¿Cuál es el tiempo medio de búsqueda?
2. ¿Cuál es la latencia de rotación media?
3. ¿Cuál es el tiempo de transferencia de un sector?
4. ¿Cuál es el tiempo medio total para resolver la petición?

Solución:

1. Como mínimo la cabeza debe recorrer 1023 pistas para atender una petición y como mínimo 0, con lo cual en promedio recorre 511,5 pistas, entonces como en recorrer 100 pistas tarda 1 ms, en recorrer 511,5 tarda 5,115 ms.
2. 15000 RPM equivalen a $\frac{15000}{60} = 250$ revoluciones por segundo, tardando $\frac{1}{250} = 4ms$ por vuelta y como el retardo rotacional es la mitad, la latencia de rotación son 2 ms.
3. Cada giro implica 4 ms y en cada pista hay 2048 sectores, en leer un sector tardamos $\frac{4}{2048} = 1,95\mu s$.
4. El tiempo medio para resolver la petición es la suma de los 3 anteriores $5,115ms + 2ms + 0,002ms = 7,117ms$.

6.19. Problema 19

El tiempo medio de búsqueda de un disco que gira a 5400 RPM es de 3 ms, la velocidad de transferencia de 5 MB/s, la sobrecarga media del controlador de 2 ms. Supones que el disco está desocupado. Calcular el tiempo medio de acceso a un sector de 512 bytes.

Solución:

De la misma forma que en el ejercicio anterior podemos saber que el disco gira a $\frac{5400}{60} = 90revoluciones/seg.$ y su latencia rotacional es de $\frac{1}{90} = 11,1ms$ $\frac{11,1}{2} = 5,6ms$.con lo cual el tiempo total es $3ms + 5,6ms + \frac{512B}{5MB/s} + 2ms$=10,7ms

6.20. Problema 20

1.Un programa realiza de forma repetitiva un proceso con tres pasos: lee del disco un bloque de 4 KB, realiza un cierto proceso con los datos y escribe el resultado en otro bloque de 4 KB. Cada bloque está formado por sectores consecutivos en la pista y los bloques se encuentran ubicados de forma aleatoria en una única pista del disco. El disco rota a 7200 RPM, tiene un tiempo medio de búsqueda de 8 ms y una velocidad de transferencia de 20 MB/s. La sobrecarga del controlador es de 2 ms. Ningún otro programa usa el disco ni el procesador y no hay solapamiento de operaciones en disco con el proceso de los datos. La etapa de proceso de datos requiere 20 millones de ciclos de reloj y la frecuencia de reloj es de 400 MHz. ¿Cuál es la velocidad del sistema de E/S en bloques procesados por segundo?

Solución:

El tiempo de acceso al disco es de $8ms + 4,2ms + \frac{4KB}{20,000KB/s} + 2ms = 14,4ms$ estos datos surgen del calculo del retardo rotacional $\frac{7200}{60} = 60rps.$ $\frac{1}{2*60} = 4,2ms$ y la aplicación de los dados en el enunciado.

Entonces, cada proceso del bloque necesita 2 accesos, lectura y escritura, por tanto 28,8ms de disco son necesarios, y el computo que no se puede solapar necesita $\frac{200*10^6 ciclos}{400*10^6 Hz} = 50ms.$ como un bloque se tarda en procesar 50+28,8=78,8ms, el número de bloques por segundo es: $\frac{1}{78,8*10^{-3}} = 12,7Bloques/s.$

6.21. Problema 21

1.Se desea leer un archivo de 256 KB de un disco que tiene un tiempo medio de búsqueda de 25 ms, velocidad de rotación de 3600 RPM, una velocidad de transferencia de 800 MB/s, el tamaño del sector es de 256 Bytes y el número de sectores por pista es de 64.

1. Calcular el tiempo de transferencia si el fichero está almacenado de la forma más compacta posible.
2. Calcular el tiempo de transferencia si el fichero está almacenado de forma que los sectores están distribuidos de forma aleatoria por su superficie.

Solución:

1. En este caso se realiza un PRIMER acceso secuencial con un tiempo de búsqueda de 25ms y un retardo rotacional de $\frac{1}{2*3600/60} = 0,0083s. = 8,3ms$ y un tiempo de lectura de 64 sectores de $\frac{256*64}{800} = 20,5ms$, sumado todo son 53,8ms a los que hay que añadir el tiempo que tardaríamos en leer las siguientes pistas, que son $\frac{256KB}{64*256B} = 16pistas$ las cuales se leen de forma continua y sin retardos rotacionales o de búsqueda, por tanto $T_{lectura} = 53,8ms + 28,8 * 15 = 485,8ms$

2. Con acceso aleatorio tendríamos que leer sector a sector, el tiempo de búsqueda y el retardo rotacional son los mismos pero el tiempo de transferencia es sólo de un sector siendo $\frac{256}{800*10^3} = 0,32ms$, por tanto cada bloque tarda en leerse 33,62 ms y son un total de 1024 bloques, así que el tiempo de leer todas es $1024 * 33,62 = 34427ms$, comparándolo con el tiempo de antes $\frac{34427}{485,8} = 70,86$veces es más rápido.