Práctico 2 - Jerarquía de Memoria

Ejercicio 1

Considere una CPU de 32 bits operando con una memoria de 64MB. Adicionalmente, entre la CPU y la memoria se coloca una memoria caché de 32KB. El caché tiene 512 líneas y la memoria es direccionable por bytes. Se utiliza una función de correspondencia asociativa por conjuntos de 4 vías y una política de reemplazo FIFO

Se pide:

1. Indique cómo se interpreta una dirección de memoria para indexar en el caché. Calcule el valor de todos los parámetros relevantes.
2. Considere que se accede a la dirección de memoria 0x1234567 y se guarda el bloque correspondiente en caché. Indique un conjunto ordenado de accesos a memoria de forma tal que en el último se reemplace el bloque que contiene a la dirección de memoria 0x1234567.
3. Sea a un arreglo de enteros de 2048 posiciones ubicado a partir de la posición 0 de memoria. Considere la siguiente porción de código:

```
for (int i = 0; i < 2048; i++){
      a[i] = i;
}
```

Sabiendo que el tipo de datos 'int' ocupa 32 bits de memoria.

   a. Calcule el hit rate en la ejecución de dicho código.
   b. Calcule la cantidad de reemplazos ocurridos durante la ejecución del código.

Ejercicio 2

Sea un computador de 32 bits y memoria direccionable por bytes que dispone de una memoria caché de 128 KB y líneas de 32 bytes, con una organización de correspondencia directa. Considere el siguiente fragmento de código:

```
for (i = 0; i< 512; i++)
        for ( j = 0; j < 512; j++)
                C[i][j] = a[i][j] + b[i][j]
```

Donde a, b y c representan matrices cuadradas de 512 x 512 números enteros, que se almacenan en memoria de forma consecutiva. Cada matriz, a su vez, se almacena por filas. La matriz a comienza en la dirección hexadecimal 0x00100000.

Se pide

a) Especifique cómo se divide la dirección de memoria entre etiqueta, línea, desplazamiento. Justifique.

b) Calcule la tasa de aciertos que se produce en la memoria caché para el fragmento de código anterior, suponiendo que la caché está inicialmente vacía.

c) Se propone un cambio en el algoritmo que recorra primero las columnas y luego las filas de las matrices ¿esto mejoraría la tasa de aciertos? ¿y almacenando las matrices por columnas? Justifique.

d) Proponga un diseño de memoria caché que mejore la tasa de aciertos para el algoritmo sin que ello suponga un incremento excesivo en el tiempo de búsqueda en la memoria caché. Calcule la tasa de aciertos para este diseño.

Ejercicio 3

Considere un CPU de 16 bits, que emite direcciones de 24 bits, operando con una memoria RAM y un Cache L1 de 1 KB, con 16 líneas y una función de correspondencia totalmente asociativa.

a) Indique cómo se interpreta la dirección de memoria para el cache indicado. Halle el valor de todos los parámetros relevantes
b) En caso de producirse un miss en el Cache L1, ¿Cuántos bytes se transfieren entre memoria principal y el cache?

1

Suponga que entre el Cache L1 y la memoria principal se agrega un Cache L2 de 32 KB, con 128 líneas y una función de correspondencia asociativa en conjuntos de 4 vías. En este tipo de jerarquías, en los caches de niveles inferiores (L2, L3), típicamente la línea no se subdivide en bytes o palabras de memoria, sino en palabras de tamaño igual al ancho de la línea del cache de nivel anterior.

Suponiendo buses de datos de 16 bits entre el CPU y el cache L1, entre el cache L1 y el cache L2, y entre el cache L2 y la memoria principal, todos operando a la misma frecuencia:

c) ¿Cómo se interpreta la dirección de memoria para el caché L2? Halle el valor de todos los parámetros relevantes.
d) ¿Cuántos bytes se transfieren entre la CPU y el cache L1 en caso de ocurrir hit en la caché L1? ¿Cuántos ciclos de bus se demora en hacer la transferencia?
e) ¿Cuántos bytes se transfieren entre el caché L1 y el caché L2 en caso de ocurrir miss en el caché L1 pero hit en el caché L2? ¿Cuántos ciclos de bus se demora en hacer la transferencia?
f) ¿Cuántos bytes se transfieren entre el caché L2 y la memoria principal en caso de ocurrir miss en el caché L1 y también en el L2? ¿Cuántos ciclos de bus se demora en hacer la transferencia?

Ejercicio 4

Considere un caché L1 con líneas de 32 bytes y un caché L2 con líneas de 256 bytes. El caché L1 y L2 están unidos por un bus de 64 bits y una ciclo de bus con frecuencia = ¼ ciclo de reloj del CPU. El caché L2 y la memoria principal se conectan con un bus de 64 bits y frecuencia = ⅛ del ciclo de reloj de CPU. El hit time de los cachés L1 y L2 son menores a un ciclo de reloj, mientras que el acceso a la memoria principal es en ráfaga, con un tiempo de acceso de 32 ciclos de reloj para la primer palabra (para el resto de las palabras, el tiempo limitante es el ciclo de bus)

Suponga que se realiza una lectura a la dirección 0xFF0000A4 que resulta miss en el caché L1 y hit en el caché L2.

a. Indique la cantidad de ciclos de reloj que pasan desde que se inicia el acceso hasta que se recibe el byte requerido en el CPU
b. Repita el cálculo anterior si se implementa *early restart* en el caché L1
c. Repita el cálculo anterior si se implementa *critical word first* en el caché L1.

Luego se realiza un acceso a la dirección 0xFE000040 que resulta miss en el caché L1 y miss en el caché L2

d. Indique la cantidad de ciclos de reloj que pasan desde que se inicia el pedido de bloque del caché L1 hasta que este se recibe completamente.
e. Explique de qué modo se podría adoptar la técnica *critical word first* y *early restart* en el caché L2 para acelerar el acceso del bloque de la parte anterior.

Ejercicio 5

Explique por qué una caché con correspondencia totalmente asociativa no es apta para usar la técnica de multibanking

Ejercicio 6

Diseñe el circuito de una caché multibanco de 8 conjuntos y 2 bancos de 8 conjuntos cada uno. Recordar que esta caché puede obtener dos accesos simultáneos.

a. Implementar el circuito lógico que indica si hay hit o miss
b. Implementar la conexión de salida de la caché

Ejercicio 7

Diseñar el circuito que detecta secuencias de la forma b, b + N, b + 2N, etc. Utilizarlo para implementar un circuito que realice el strided prefetch en un caché. **Sugerencia:** implementar la función lógica que detecta las secuencias pedidas.