

Trabajo Práctico 1: Memoria Cache

Carrera: INGENIERIA EN INFORMATICA

Asignatura: 3638-Arquitectura de computadoras

Tema: Memoria Caché

Unidad: 1-Memoria

Objetivo: Comprender el funcionamiento y organización de la memoria caché. Comprender el impacto de utilizar los distintos modos de organizar la memoria caché. Comprender cómo impacta la forma de programar en la performance de la memoria caché.

Competencia/s a desarrollar:

Genéricas:

- Desempeño en equipos de trabajo.
- Comunicación efectiva.
- Actuación profesional ética y responsable.
- Aprendizaje continuo.
- Desarrollo de una actitud profesional emprendedora.

Específicas:

- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.
- Gestión, planificación, ejecución y control de proyectos de ingeniería en sistemas de información / informática.
- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.

Descripción de la actividad:

1-Tiempo estimado de resolución: 2 semanas

2-Metodología: Escrito y mediante la herramienta de simulación Ripes.

3-Forma de entrega: La entrega y la resolución no es obligatoria.

4-Metodología de corrección y feedback al alumno: Presencial y por Miel.

A- Memoria Cache

A.01 Una memoria caché posee líneas de 128 bits y palabras de 2 bytes. Indique cuántas palabras por línea posee. Si se direcciona al byte, indique cuántos bits se utilizan de offset.

A.02 Una memoria caché posee líneas de 128 bits y almacena 32 palabras. Indique cuál es el tamaño de la palabra. Si se direcciona al byte, indique cuántos bits se utilizan de offset.

A.03 Una memoria caché posee líneas de 128 palabras de 4 bytes. Indique cuál es el tamaño de la línea. Si se direcciona al byte, indique cuántos bits se utilizan de offset.

A.04 Dada una computadora que direcciona 4G Palabras de un byte, se decide implementar una memoria caché configurada con mapeo directo con el fin de hacer la misma más eficiente.

Para identificar al bloque en una dirección (etiqueta) se utilizan tantos bits como los primeros dos dígitos de su DNI sobre dos (redondeado hacia arriba en el caso impar).

Ejemplo, DNI=41.456.789 entonces se utilizan $41/2 = 20,5$ entonces 21.

La cantidad de palabras por línea se definen según el último dígito de su DNI de forma tal que:

DNI's terminados en 0,1,2,3,4 = 4 palabras por línea.

DNI's terminados en 5,6,7,8,9 = 8 palabras por línea.

Indique (para su DNI y recomendamos practicar con otros DNI's):

- Cuántas líneas tiene la memoria caché.
- El tamaño de la memoria caché (en múltiplo de bytes).
- El tamaño de cada bloque (en múltiplo de bytes).
- La cantidad de bloques en que queda dividida la MP.
- Convierta su DNI a binario asumiendo que cada dígito es BCD:

Ej DNI 12.345.678 = 0001 0010 0011 0100 0101 0110 0111 1000 .

Indique en qué línea y en qué offset debería encontrarse esa dirección en MC.

- Si un acceso a MC tarda 1ns, un acceso a MP tarda 10ns, y la tasa de aciertos es del 99%, indique si es conveniente utilizar esta memoria caché. Indique qué ocurre si la tasa de aciertos es del 99,9%

A.05 A un sistema de computación se decide incorporar una memoria caché de 1024 Kilo Palabras con líneas de 1024 bits. La computadora trabaja en palabras de 16 bits y puede direccionar 8192 Mega Palabras. La computadora direcciona al byte.

El sistema es utilizado para almacenar secuencialmente video por ende se realizan permanentemente escrituras en direcciones consecutivas (de manera secuencial) pero rara vez se accede aleatoriamente a las direcciones.

Según su DNI, se decide organizar la memoria en:

DNI's terminados en 0,4,5,9 ---> Mapeo Directo

DNI's terminados en 1,3,6,8 ---> Mapeo Asociativo

DNI's terminados en 2,7 --> Mapeo asociativo por conjunto (2 vías)

Indique según el esquema que le haya tocado:

- El tamaño de la memoria caché
- El tamaño máximo de la memoria principal
- Indique cuántas líneas forman la memoria caché
- Indique cómo queda dividida una dirección de memoria (cuántos bits para offset, etiqueta y línea según su caso)
- Indique cuántas comparaciones se realizan en paralelo para saber si un dato se encuentra en memoria caché.
- Para una línea, cuántos bloques de memoria pueden asignarse a ella.
- Elija 34 bits al azar. Indique en qué offset debería encontrarse en una línea. Luego indique (según el caso) en que línea/conjunto debería buscarse, y cual es el valor de la etiqueta a comparar.
- Si el tiempo de acceso a memoria caché es 1ns, y el tiempo de acceso a memoria principal es 10ns, y se produce un fallo cada 100 accesos, indique si es conveniente utilizar esta memoria caché justificando.
- Teniendo en cuenta el uso dado al sistema (grabación de vídeo) , indique qué método de reemplazo es el más apropiado justificando brevemente.

B - Memoria Caché con Ripes

B.01 Teniendo en cuenta el siguiente ejemplo de programa escrito en Ripes:

```
.data
cadena: .string "Trabajo practico de memoria caché en Ripes"
.text
    la x5,cadena

salto:
    lb x6,0(x5)
    add a0,x0,x6
    addi a7,x0,11
    ecall
    addi x5,x5,1
    bne x6,x0,salto
fin:
    beq x0,x0,fin
```

Indique, utilizando las configuraciones de la memoria cache en Ripes, cuál sería la configuración que utilice la **caché de datos** de menor tamaño en:

- Asignación asociativa
- Asignación Directa
- Asignación asociativa por conjunto

Indique cómo quedarían configurados los diferentes campos en cada caso

B.02 Teniendo en cuenta direcciones de 32 bits, una asignación asociativa de 4 líneas y dos palabras por línea, en que línea y con cual offset se ubicaría el inicio de la palabra “memoria” del ejercicio anterior.

B.03 Teniendo en cuenta direcciones de 32 bits, una asignación directa de 4 líneas y cuatro palabras por línea, en que línea y con cual offset se ubicaría el inicio de la palabra “cache” del ejercicio anterior.

B.04 Teniendo en cuenta direcciones de 32 bits, una memoria caché de 512 bits y una asignación asociativa por conjunto de 2 vías y dos palabras por línea, en que línea, cuál conjunto y con cual offset se ubicaría el inicio de la palabra “Ripes” del ejercicio anterior.

B.05 El siguiente ejercicio escrito para Ripes invierte el vector al finalizar su ejecución.

```
1 .data
2 vector: .word 1,2,3,4,5,6,7,8,9,10,20,30,40,50,60,70,80,90
3 final:
4 .text
5
6 la x5,vector
7 la x6,final
8 addi x6,x6,-4
9 invertir:
10 lw x10,0(x5)
11 lw x11,0(x6)
12 sw x11,0(x5)
13 sw x10,0(x6)
14 addi x5,x5,4
15 addi x6,x6,-4
16 bge x6,x5,invertir
17 fin:
18 beq x0,x0,fin
```

Indique el formato y contenido de los campos de la caché de instrucciones, para cada método de asignación pre-seteados en Ripes para la instrucción `sw x11,0(x5)`

B.06 Al finalizar la ejecución del ejercicio anterior que instrucciones queda cargadas en la línea 2 de la memoria caché de instrucciones con asignación asociativa 32 líneas 4 palabra por línea.

B.07 Al finalizar la ejecución del ejercicio 5 en qué línea de la memoria caché de datos queda cargado el valor 20 teniendo en cuenta asignación asociativa 32 líneas 4 palabra por línea, y en qué línea si fueran 2 palabras por línea.

B.08 Indicar en que se modifica la configuración de la memoria caché en Ripes dependiendo si la escritura es inmediata o diferida (Write through vs write back), indique que sucede y justifique.

B.09 Teniendo en cuenta el siguiente programa escrito en Ripes

```

1  .data
2  vector: .word 1234,5678,3756,2569
3  final:
4  .text
5  la x12,vector
6  la x13,final
7
8  siguiente:
9      lw x5,0(x12)
10     li x6,10
11     li x7,0
12 loop:
13     sub x5,x5,x6
14     addi x7,x7,1
15     bge x5,x6,loop
16     addi x5,x5,0x30
17     add a0,x0,x5
18     addi a7,x0,11
19     ecall
20     add x5,x0,x7
21     addi x7,x0,0
22     bge x5,x6,loop
23
24     addi x5,x5,0x30
25     add a0,x0,x5
26     addi a7,x0,11
27     ecall
28     addi x12,x12,4
29     bne x12,x13,siguiente
30
31 fin:
32     beq x0,x0,fin

```

Indique cuál sería la caché de instrucciones de menor tamaño, en cada una de los métodos de asignación, en el cual el programa anterior produce la menor cantidad de fallos posibles.

B.10 Utilizando el programa del ejercicio B.09, si la caché de instrucciones está configurada con asignación directa 4 líneas y 2 palabra por línea, indique lo siguiente

- En qué línea y con qué offset se carga la instrucción sub x5,x5,x6 del programa.
- Qué instrucciones se cargan en la línea 3 en el transcurso del programa

B.11 Utilizando el programa del ejercicio B.09, si la caché de instrucciones está configurada con asignación asociativa por conjunto de 8 líneas y 2 vías de 2 palabra por línea, indique lo siguiente

- Al momento de completarse por primera vez el conjunto 0 qué instrucciones se encuentran en dicho conjunto.
- Indique cuántos fallos se producen al finalizar la ejecución del programa
- Cómo podría mejorarlo, justificando su decisión

B.12 Escriba un programa en C que implemente varias funciones que reciban distintos argumentos (valores enteros signados). Ej: una función que reciba 4 elementos y retorne el promedio, otra que reciba 3 elementos y retorne la suma, etc. Asegúrese de declarar variables locales dentro de cada función. Analice el impacto que tiene la utilización del stack pointer y el frame pointer para almacenar las variables locales en la eficiencia de la memoria caché. Realice este análisis para configuraciones en mapeo directo y mapeo asociativo.