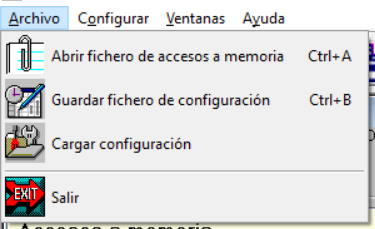
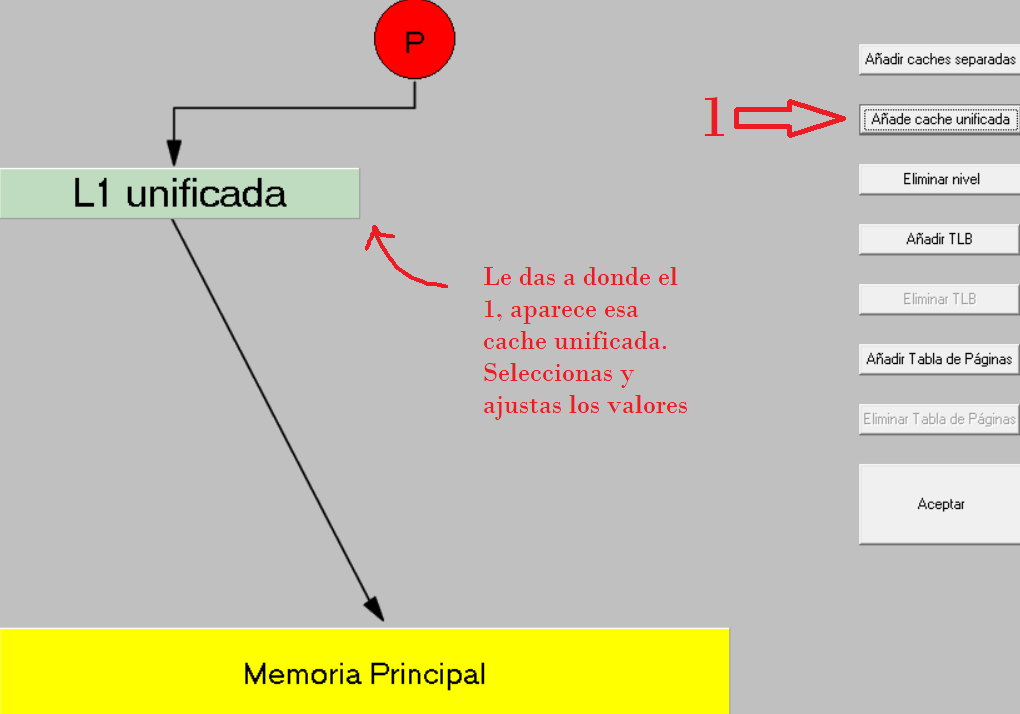
**Práctica 4 OC 2024**

Para montar la práctica descargar la trazaF

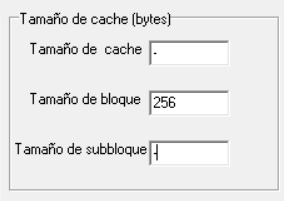


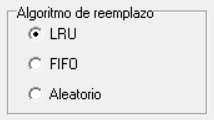
Clicar en abrir fichero de accesos a memoria y eliges la traza que necesitas (trazaF en tu caso)

Para poder elegir el tipo de memoria y todo eso selecciona el martillo



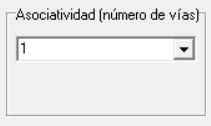
Estudio 1   
1. Tamaño de bloque (fijo): 256B. EL TAMAÑO DEL SUB-BLOQUE DEBE DE SER EL MISMO

  
2. Algoritmo de reemplazo (fijo): LRU.

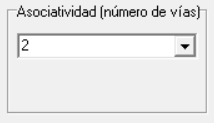


3. Política de correspondencia (variable):

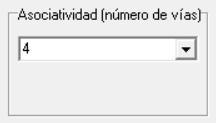
3.1 Directa



3.2 Asociativa por conjuntos de 2 vías



3.3 Asociativa por conjuntos de 4 vías,.



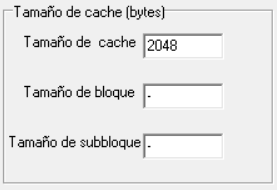
4. Tamaño de caché (variable): 2KB, 4KB, 8KB, 16KB.

2KB = 2048 bytes

4KB = 4096 bytes

8KB = 8192 bytes

16KB = 16384 bytes



Sustituye el 2048 por 4096, 8192… cada vez que haces un proceso

En el primer caso deberá de ser

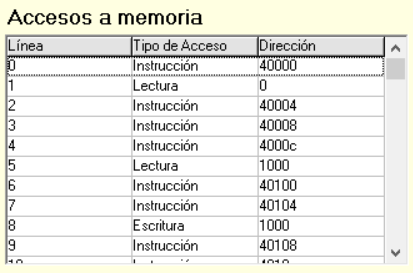
Tamaño de cache: 2000

Tamaño de bloque: 256

Tamaño de subbloque: 16

Asociatividad: 1

LRU



Primero debemos analizar los accesos a memoria (40 primeros), tienes sobre todo que ver las direcciones, que se van repitiendo, por lo que podemos suponer que habrá algún data hazard debido al acceso de dos instrucciones.

Si nos paramos a ver la instrucción 5 y 8 vemos que se produce una lectura de la dirección 1000 y una escritura en 1000. Esto puede ser un **RAW** **(Read After Write)**, si el dato leído en la línea 5 es usado por una instrucción posterior y luego sobrescrito en la línea 8 antes de que la instrucción que usa ese dato termine de procesarse.

Entre la 21 y la 24 lo mismo de antes

Entre la 29 y la 32 lo mismo

**IMPORTANTE**

### **1. Localidad Temporal (Temporal Locality):**

La localidad temporal ocurre cuando una dirección de memoria es accedida repetidamente en un intervalo corto de tiempo.

**Ejemplo en tus datos:**

* Dirección 4010c:
  + Aparece en las líneas **10**, **18**, y **26** en un intervalo corto de líneas.
* Dirección 40200:
  + Aparece en las líneas **11**, **20**, y **27**.

Es común en instrucciones de bucles, ya que las mismas direcciones se acceden varias veces durante la iteración.

### **2. Localidad Espacial (Spatial Locality):**

La localidad espacial ocurre cuando las direcciones de memoria accedidas están físicamente cercanas unas a otras. Esto suele suceder cuando se accede a arreglos, estructuras o bloques consecutivos de memoria.

**Ejemplo en tus datos:**

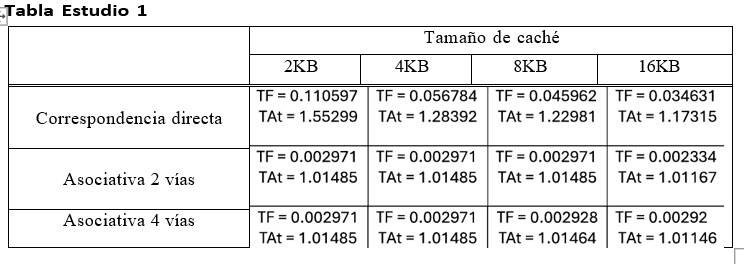
* Direcciones consecutivas de instrucciones:
  + 40000, 40004, 40008 y 40008 aparecen en las líneas **0**, **2**, **3** y **4** respectivamente. Esto sugiere que se están ejecutando instrucciones consecutivas.
* Direcciones en el rango 40100-40108:
  + Accedidas en las líneas **14**, **15**, y **17**. Las direcciones cercanas sugieren que podrían pertenecer a un bloque de código o datos contiguos.

Por lo que en esta traza tenemos tanto localidades espaciales como temporales

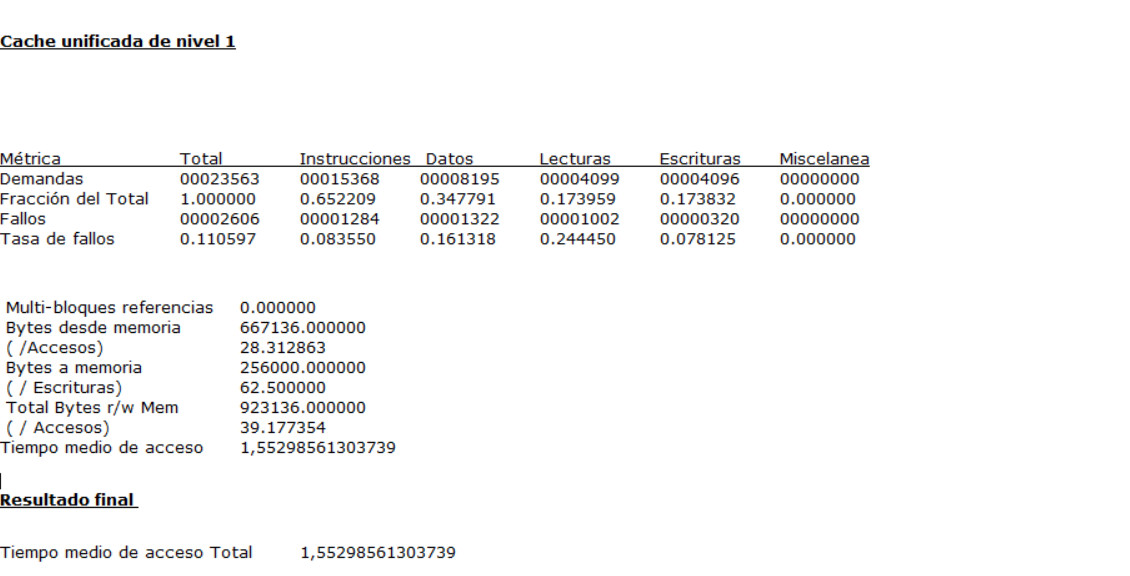
**Estudio 1. Tasa de fallos medio. Política de correspondencia-Tamaño caché**

Cada vez que cambies la configuración, reinicia y dale a completo

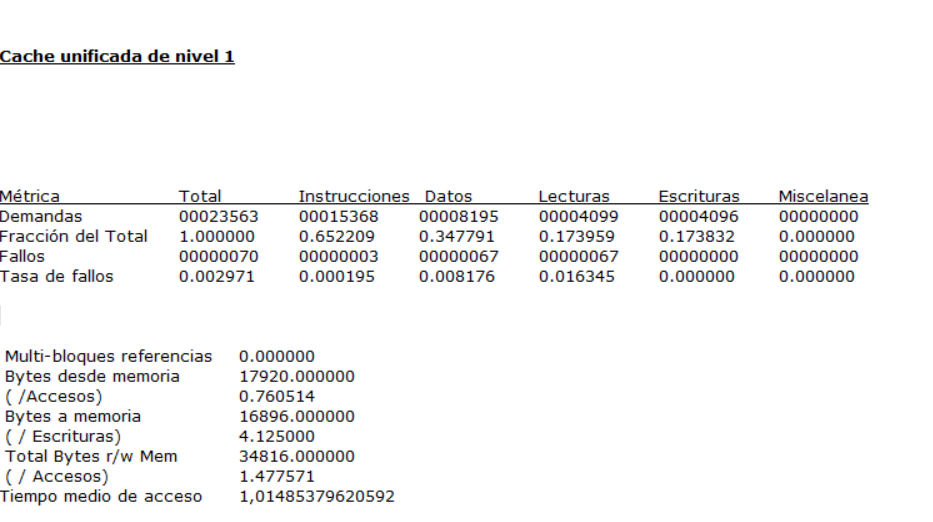
A la figura esta para ver los datos después



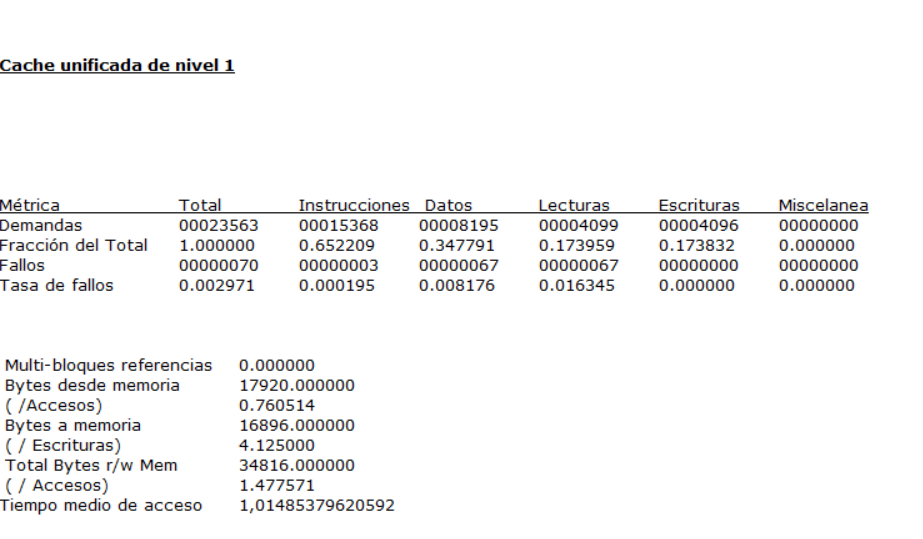
2K Direct mapping



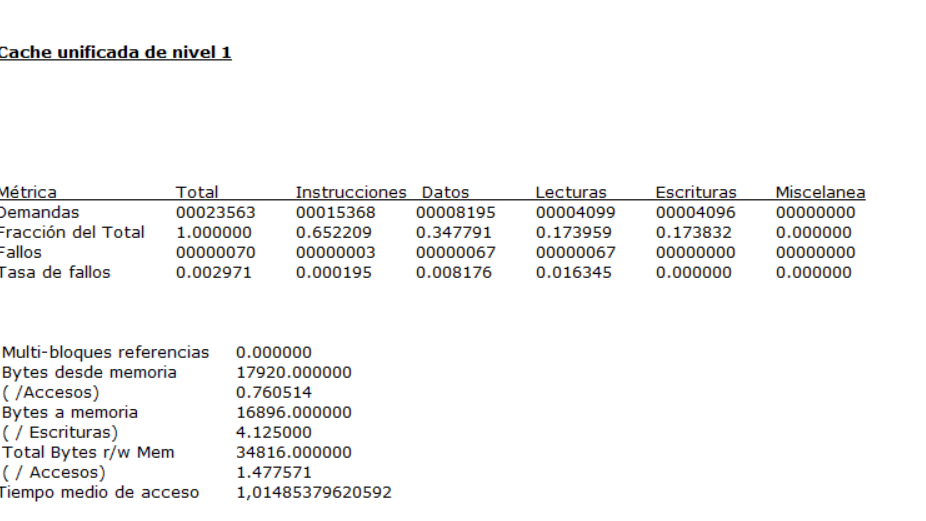
2K 2-way



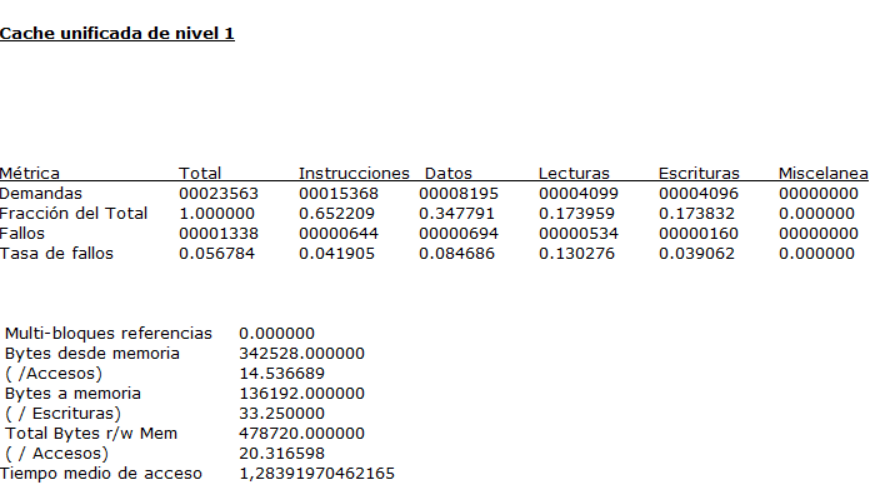
2K 4-way



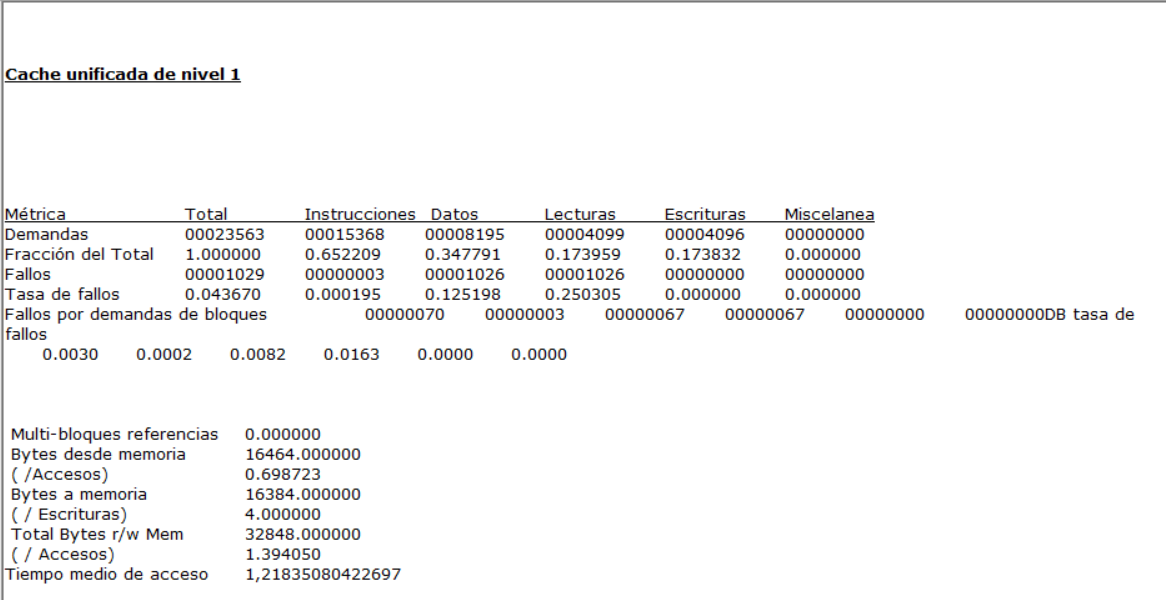
2K fully aso



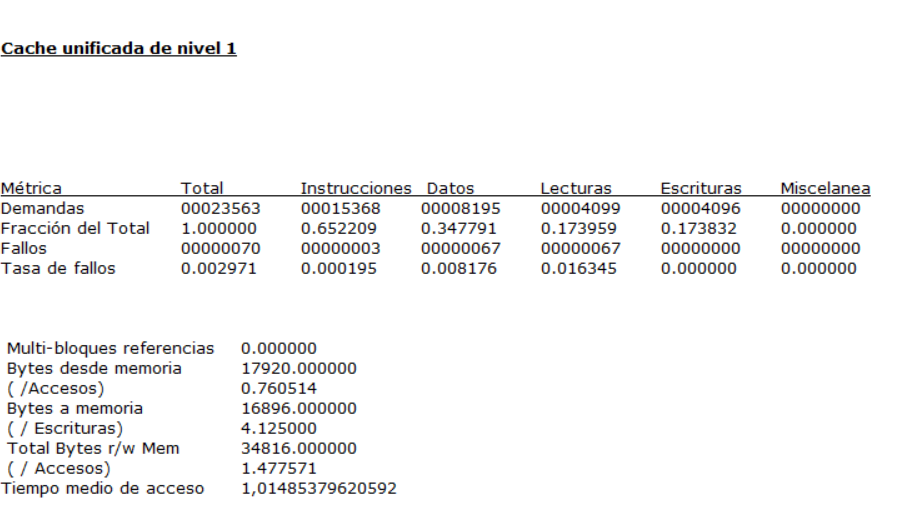
4K Direct mapping



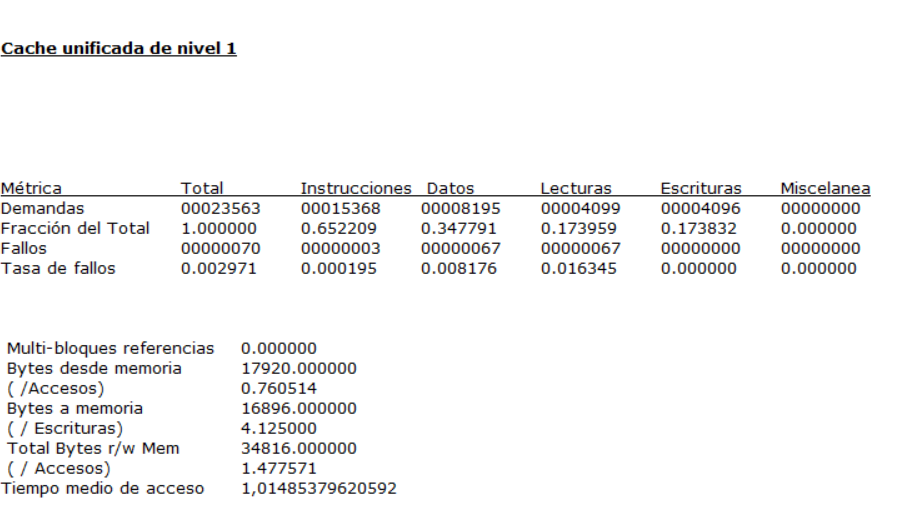
4K 2-way



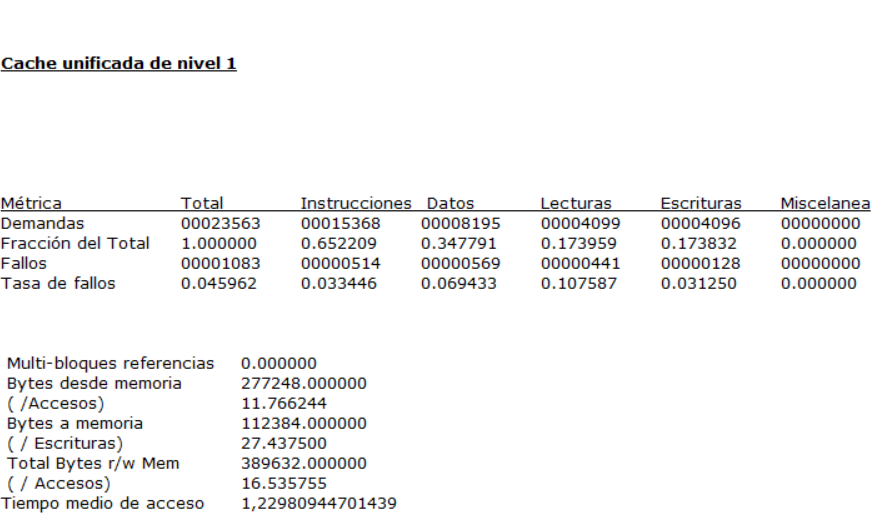
4K 4-way



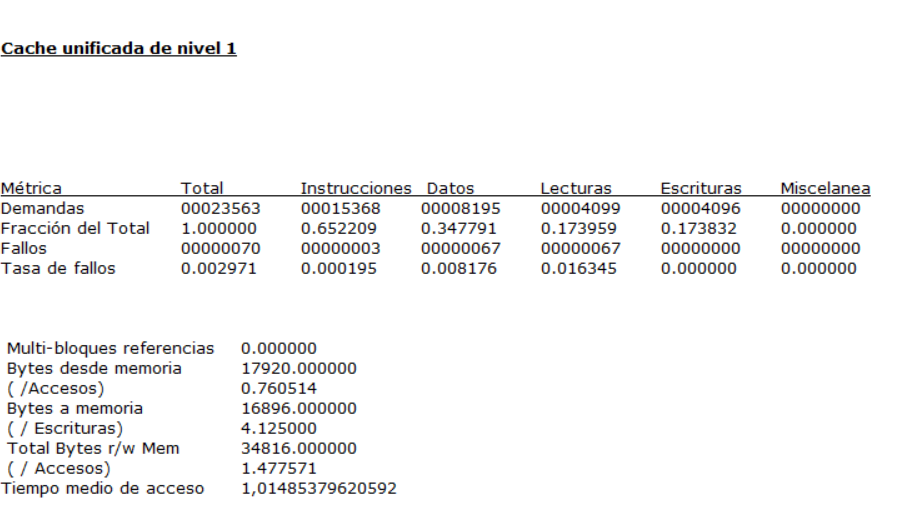
4K fully aso



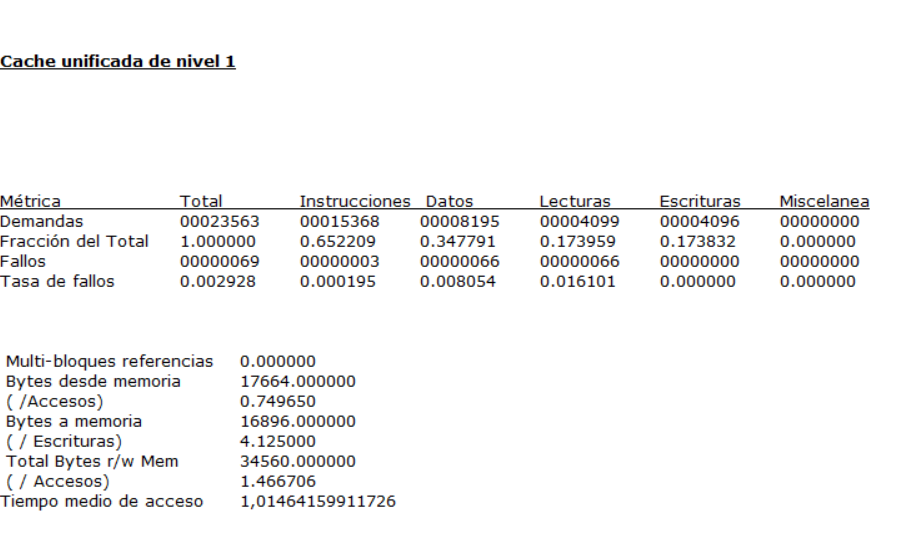
8K Direct mapping



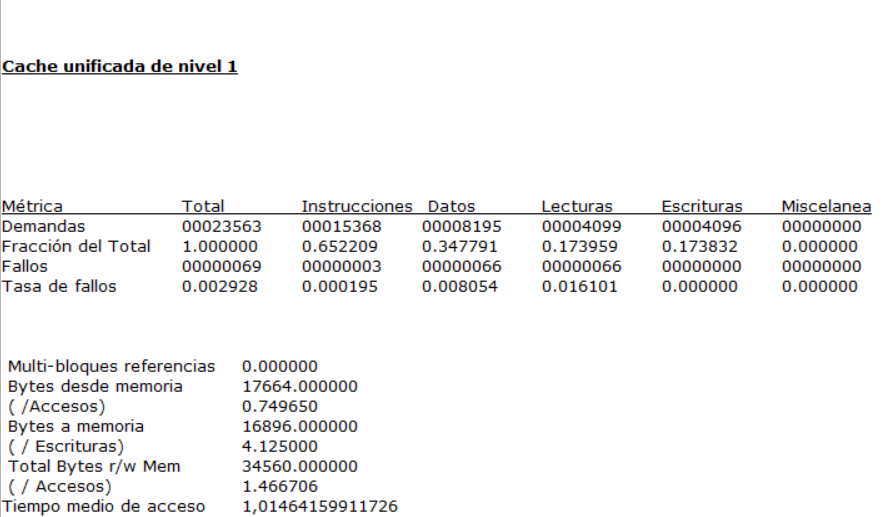
8K 2-way



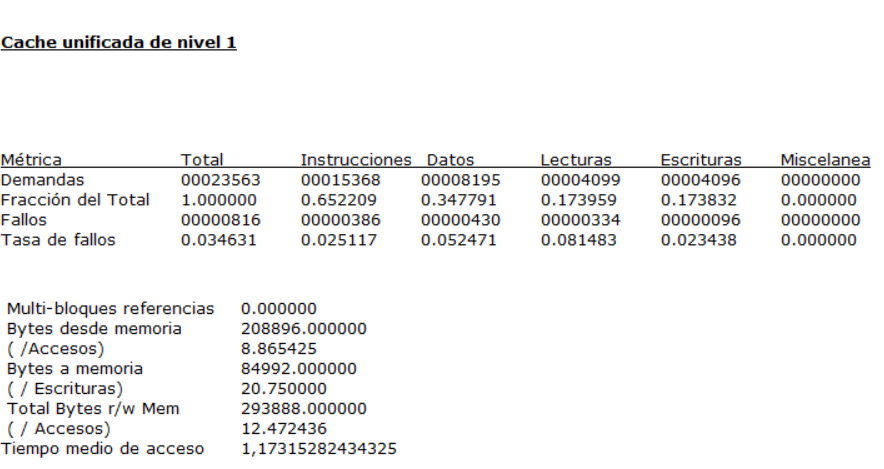
8K 4-way



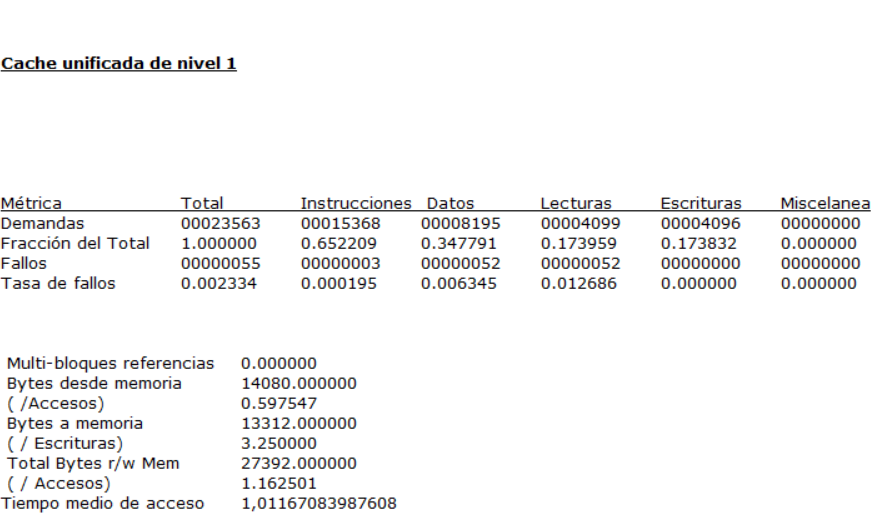
8K fully aso



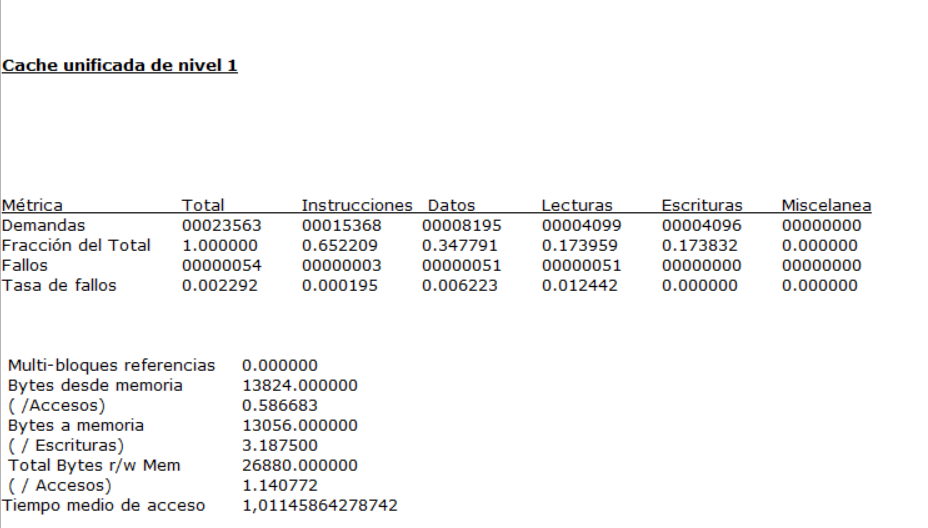
16K Direct mapping



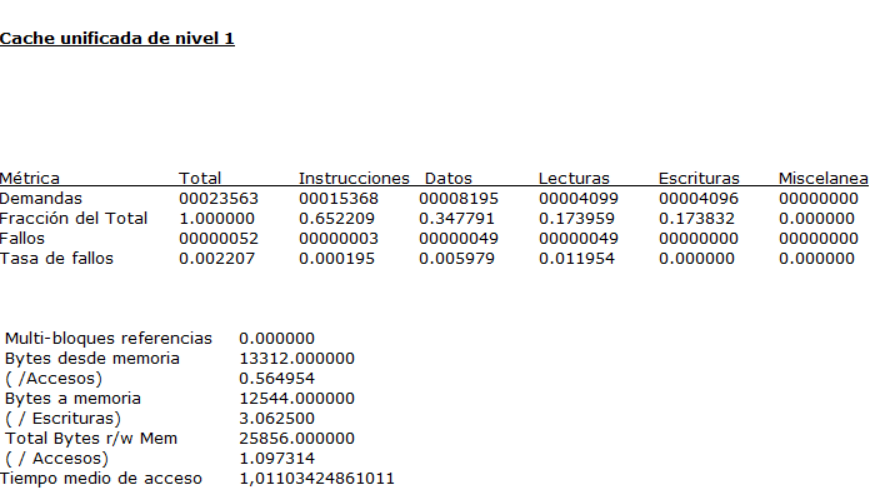
16K 2-way



16K 4-way

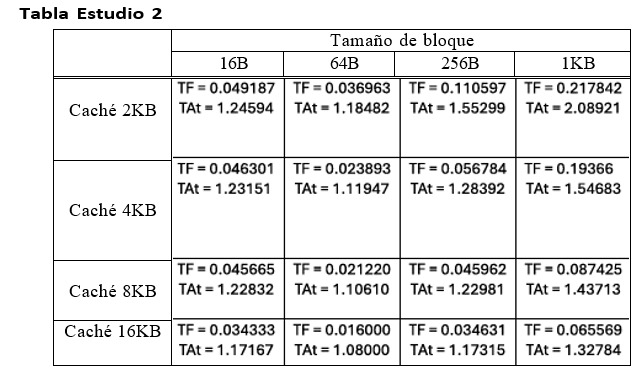


16K fully aso

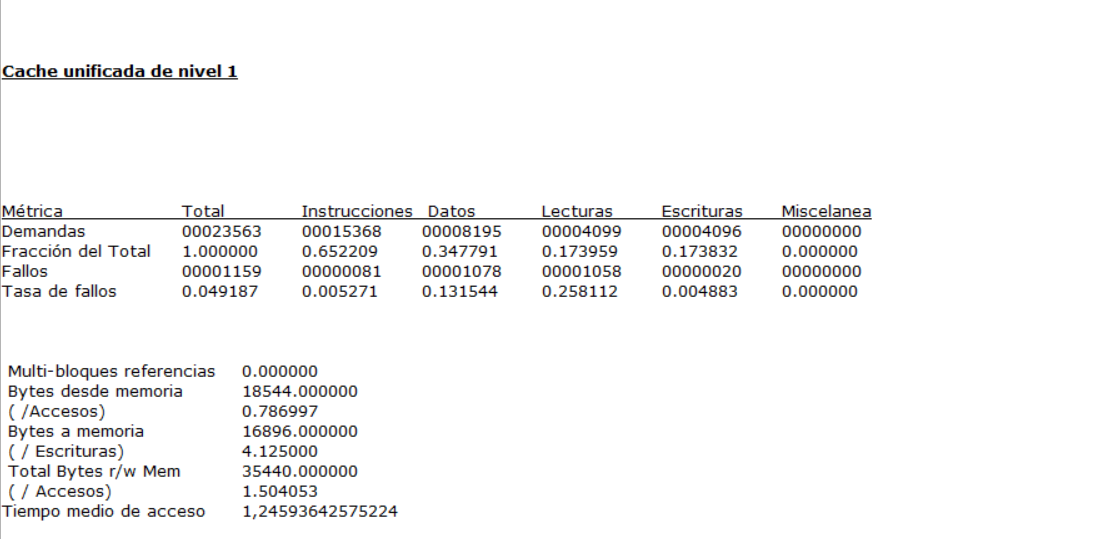


**Estudio 2. Tasa de fallos medio. Tamaño bloque -Tamaño caché**

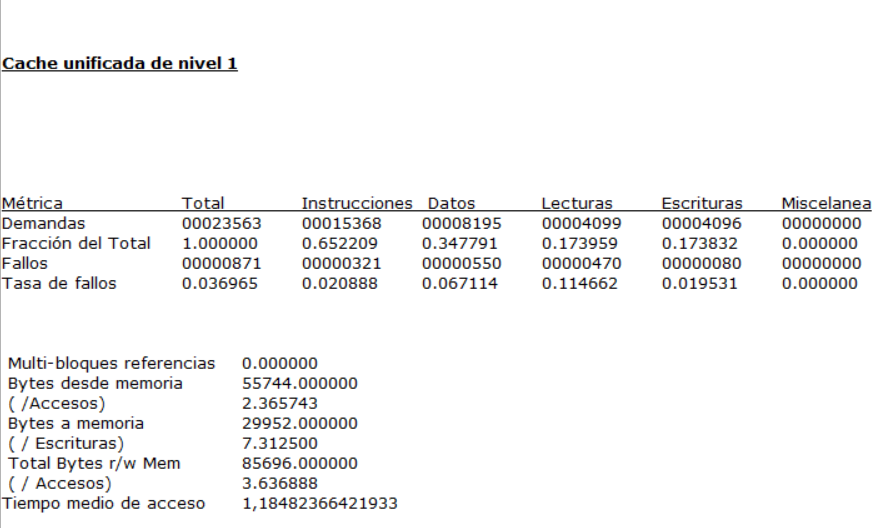
Direct Mapping (el número 1), LRU (Least recently used)



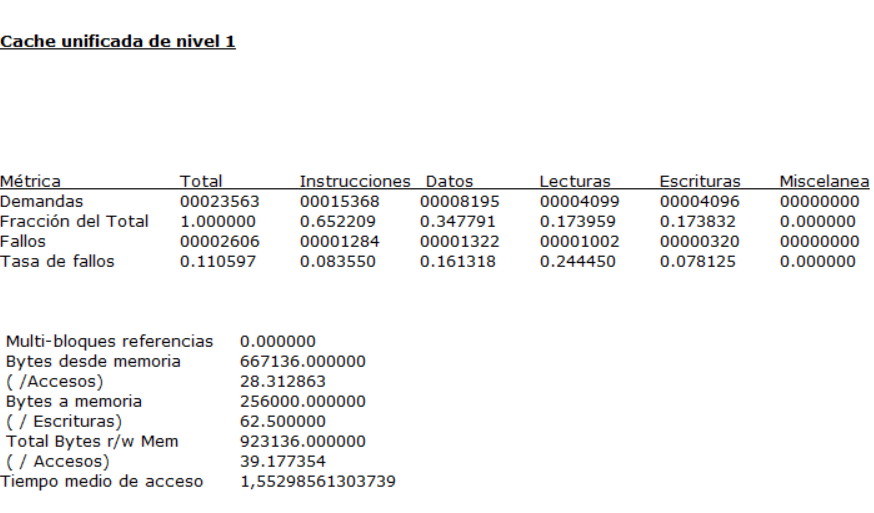
2KB - 16B



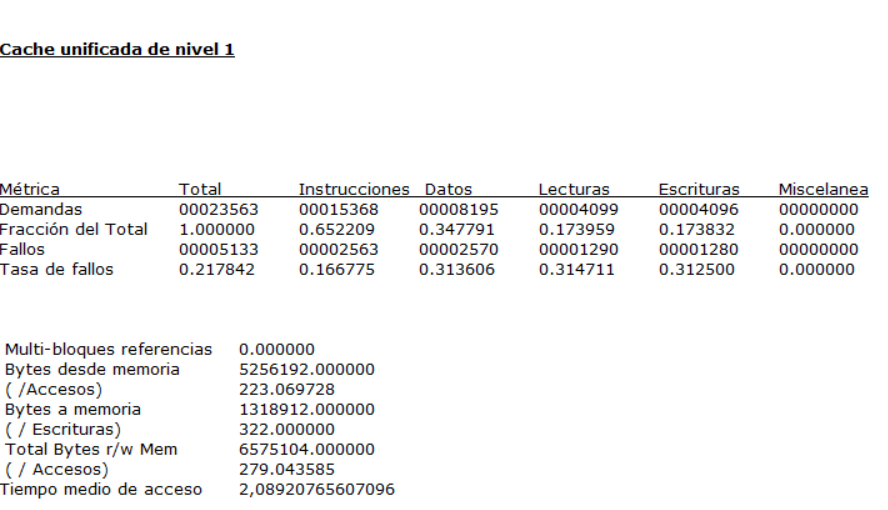
2KB - 64B



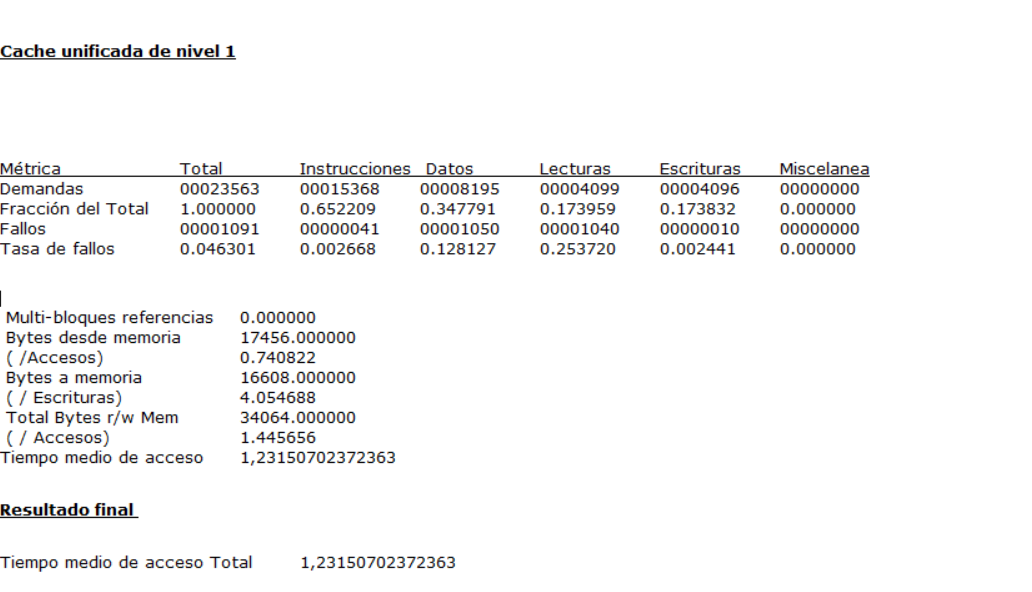
2KB - 256B



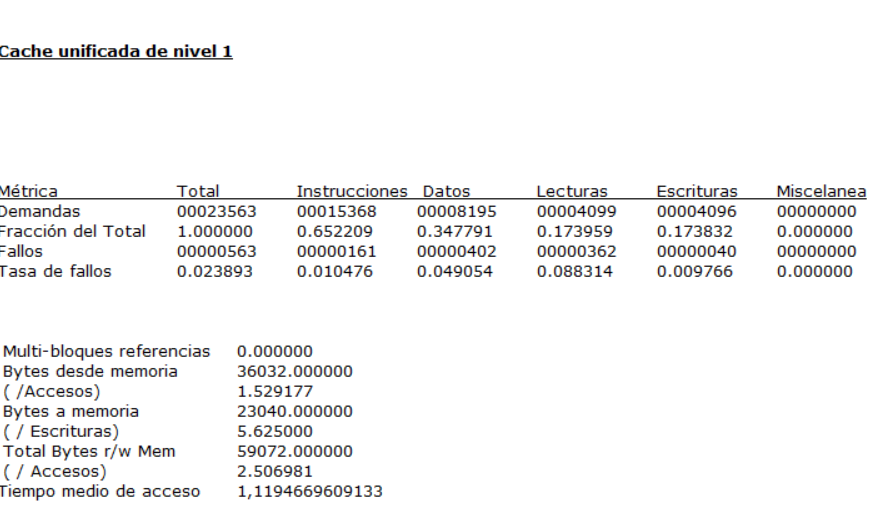
2KB - 1KB (1024B)

****

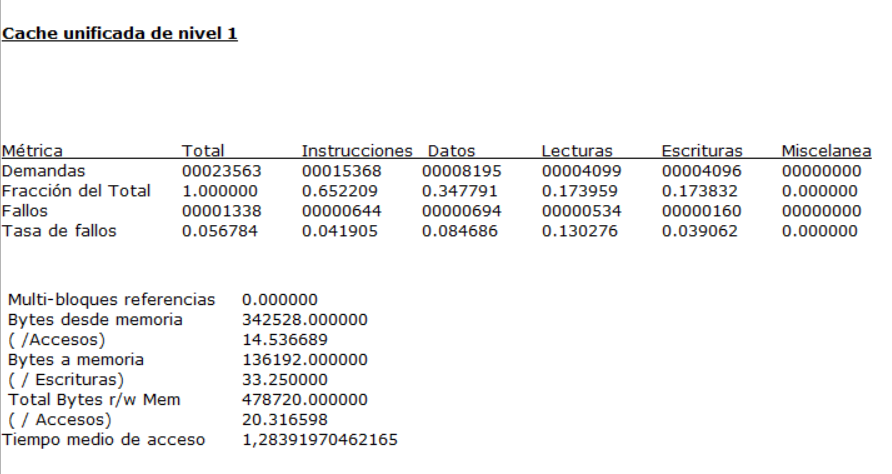
4KB - 16B



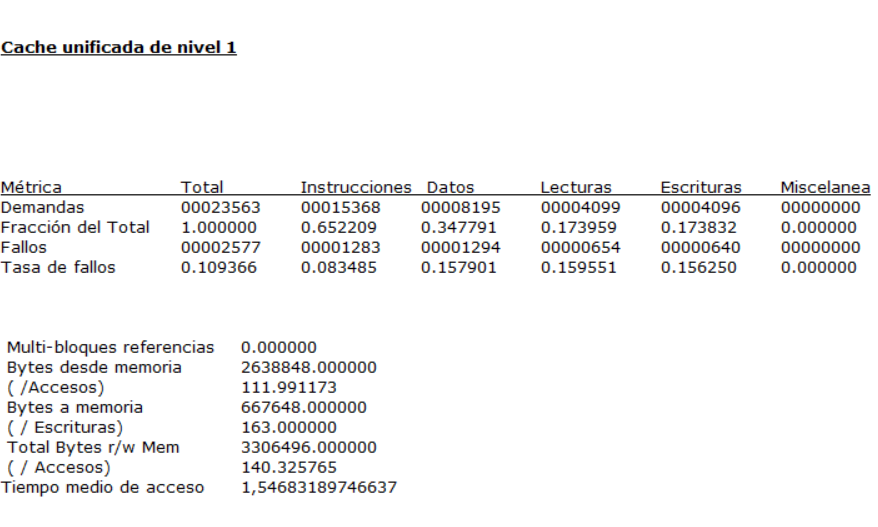
4KB - 64B



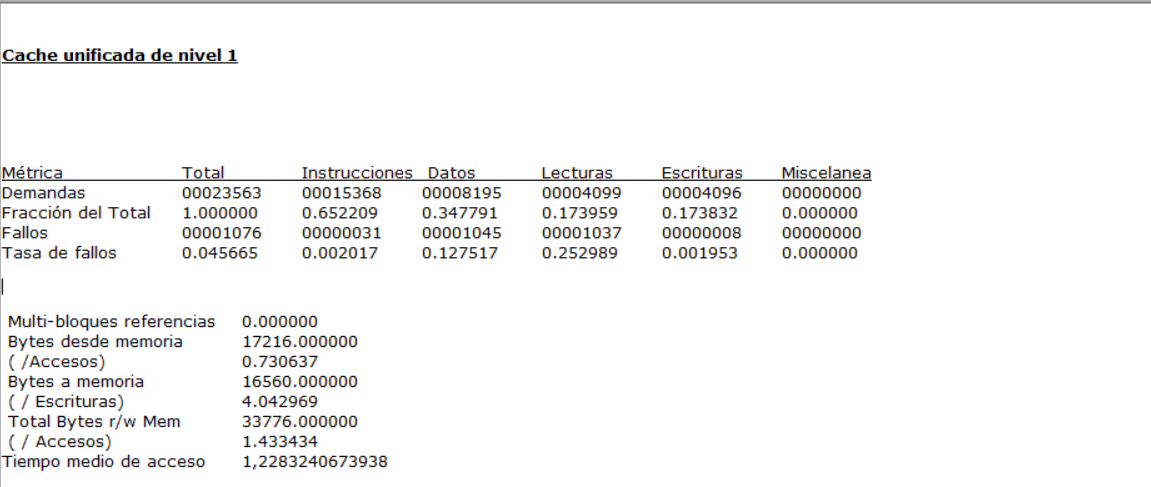
4KB - 256B



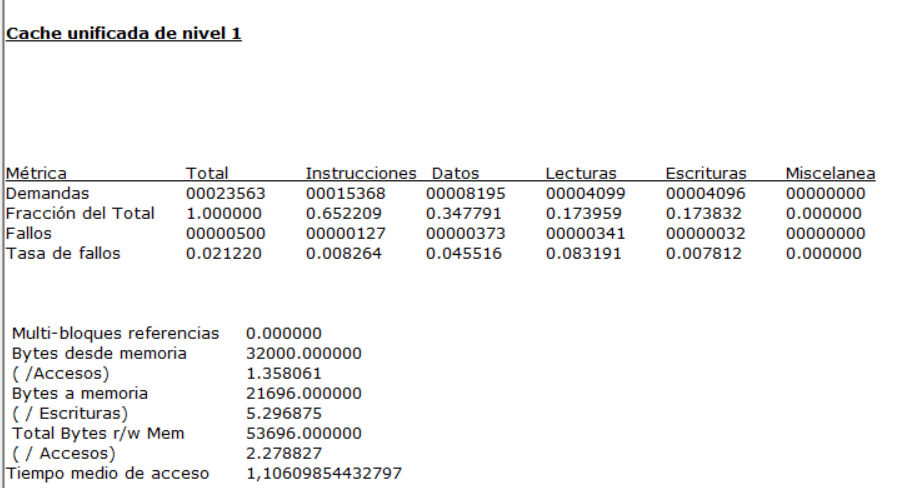
4KB - 1KB (1024B)

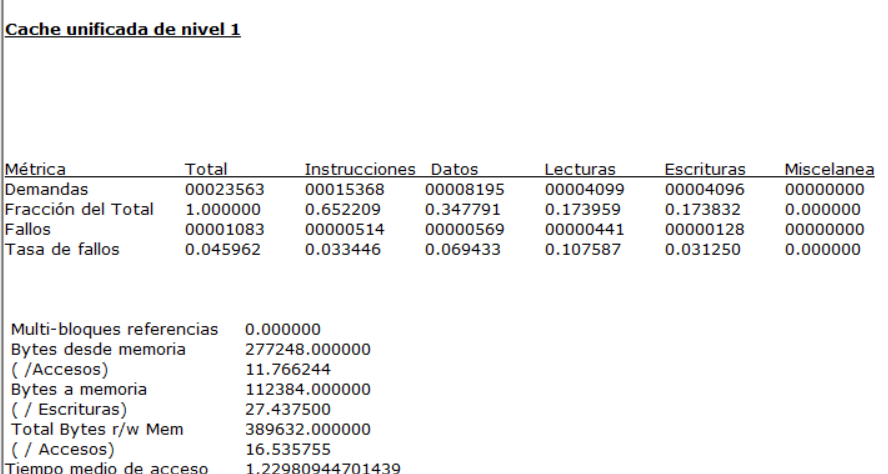


8KB - 16B

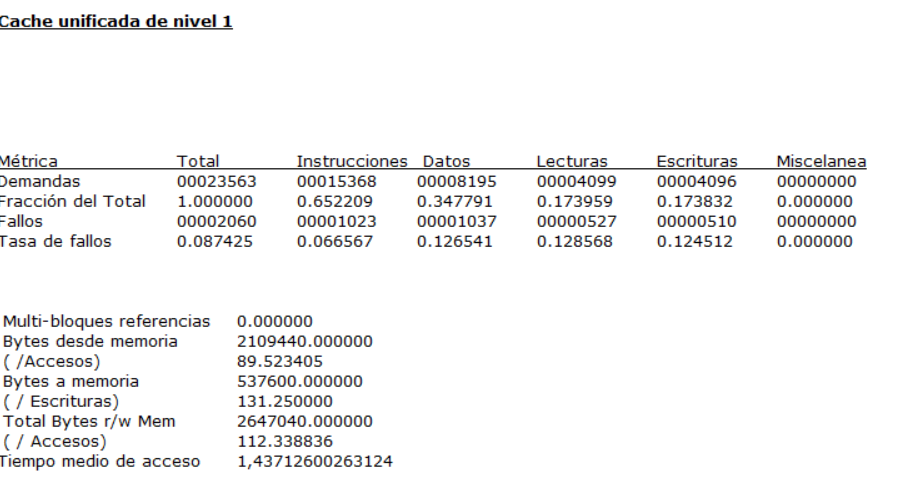


8KB - 64B

8KB - 256B



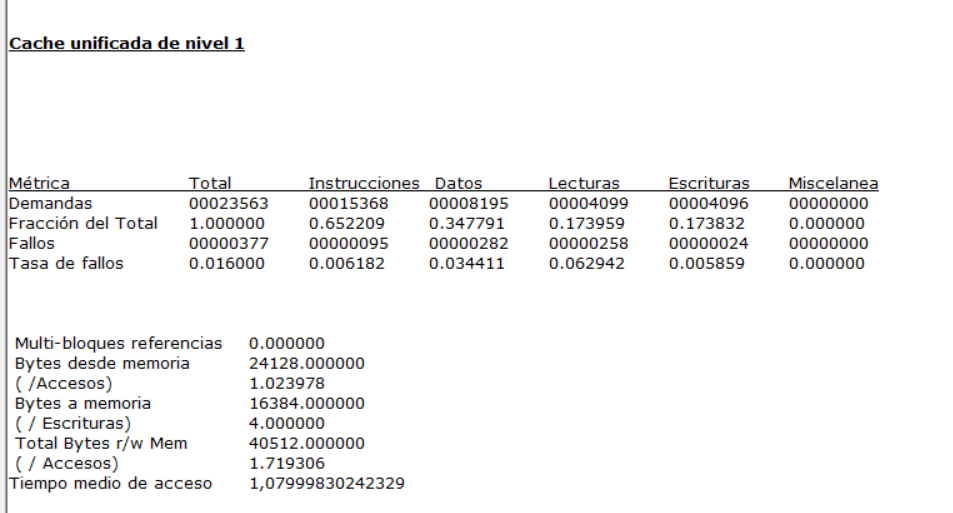
8KB - 1KB(1024B)



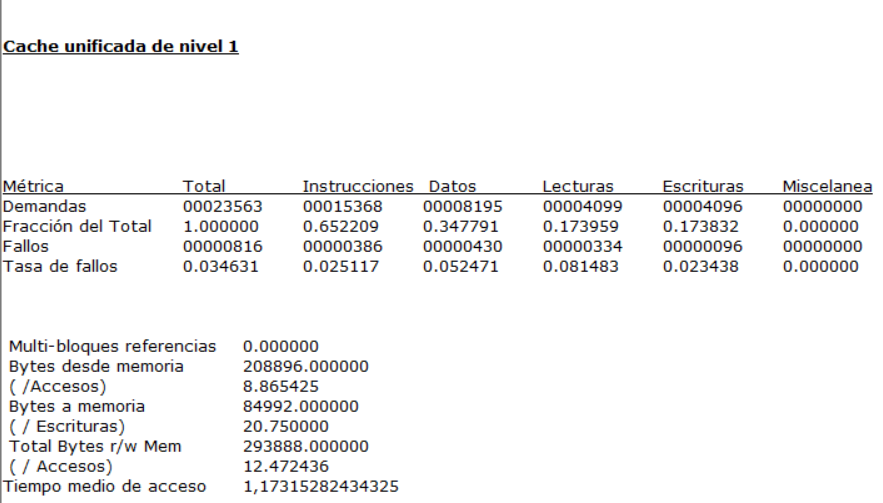
16KB - 16B



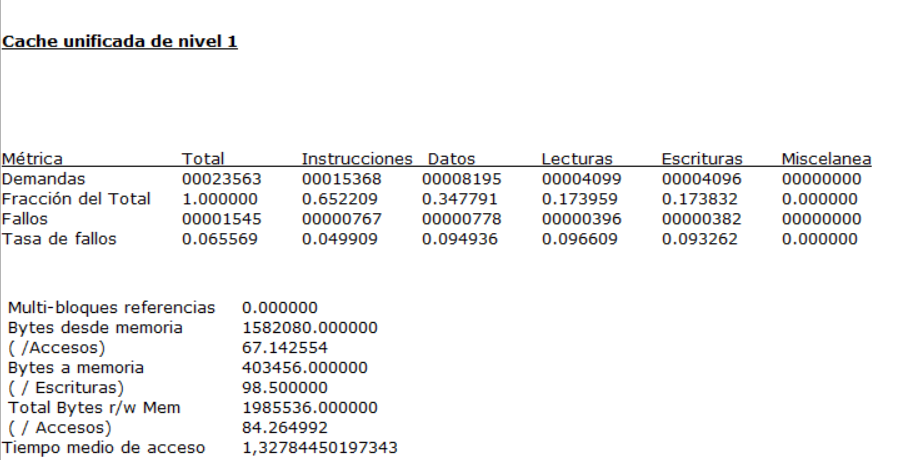
16KB-64B



16KB-256B



16KB-1KB(1024B)



Tomado de los resultados del estudio 1 no podemos obtener ninguna conclusión en relación al grado de asociatividad ya que debemos saber el impacto en los accesos a caché. Explicación detallada:  
Considerando solo los datos de miss rate. Aunque un mayor grado de asociatividad reduce el miss rate, el tiempo promedio de acceso depende tanto del miss rate como del tiempo de acceso a la caché:

Aumentar la asociatividad puede reducir el miss rate.

Pero el tiempo de acceso a la caché también **puede aumentar debido a la complejidad adicional en el hardware**.

Por lo tanto, no podemos concluir si aumentar la asociatividad mejorará o empeorará el tiempo promedio de acceso sin saber cómo afecta al tiempo de acceso a la caché.

Tomado de los resultados del estudio 2 podemos ver que incrementar el tamaño del bloque puede mejorar la miss rate porque la localidad espacial está aprovechada.

Explicación detallada:

**Bloques pequeños (16B):**

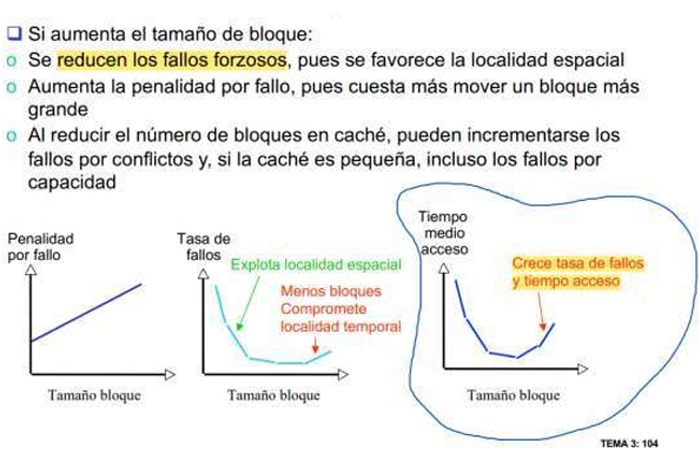
* Para un tamaño de caché dado, el *miss rate* es más bajo con bloques pequeños. Por ejemplo:
  + Con 2KB: *miss rate* = **0.049187**.
  + Con 4KB: *miss rate* = **0.046301**.

**Bloques medianos (64B a 256B):**

* El *miss rate* comienza a aumentar a medida que el tamaño del bloque crece. Ejemplo:
  + Para 2KB y 64B: *miss rate* = **0.036963**.
  + Para 2KB y 256B: *miss rate* = **0.110597**.

**Bloques grandes (1KB):**

* Se van reduciendo los miss rates a medida que se amplía el tamaño de la caché



**Extra para calcular los fields en RISC-V. ¡¡¡¡ESTUDIAR POR TU CUENTA!!!!**

**Desglose de direcciones en la caché**

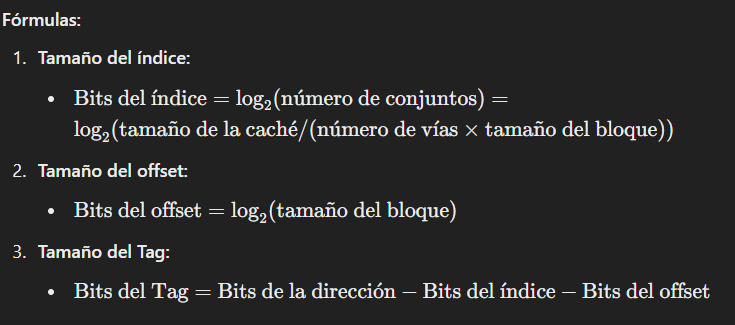
En sistemas de memoria caché, una dirección de memoria se divide en tres partes principales:

1. **Tag Field (Etiqueta):**
   * Se utiliza para identificar el bloque específico de memoria en un conjunto o línea de la caché.
   * Ayuda a resolver colisiones cuando múltiples bloques de memoria pueden mapearse a la misma línea.
2. **Index Field (Índice):**
   * Determina a qué conjunto o línea de la caché pertenece la dirección.
   * El tamaño del índice depende de la cantidad de conjuntos en la caché.
3. **Block Offset (Desplazamiento):**
   * Identifica la posición específica dentro de un bloque de datos (por ejemplo, una palabra o byte).
   * El tamaño del desplazamiento depende del tamaño del bloque de caché.

### **Cómo se calcula el campo Tag en RISC-V**

#### **Suposiciones típicas:**

* **Tamaño de caché = 2^N bytes**
* **Tamaño de bloque = 2^B bytes**
* **Número de líneas de caché = 2^L líneas**

****