

PRÁCTICA 5: JERARQUÍA DE MEMORIA I

ARQUITECTURA DE COMPUTADORES. 2º CURSO

1. OBJETIVOS

La presente práctica pretende ilustrar la estructura interna que poseen la memoria caché y la memoria principal de un computador, así como la decodificación de la dirección de acceso a las mismas. De igual forma se pretende trabajar con una traza básica de acceso a la jerarquía de memoria para comprobar el funcionamiento de las diversas políticas de ubicación.

2. PREPARACIÓN

Antes de acudir a la sesión de laboratorio el alumno debe:

- 1) Leer y asimilar los contenidos teóricos de la memoria caché (políticas de ubicación/localización).
- 2) Realizar el cuestionario que se encuentra a continuación de forma manuscrita. Este cuestionario deberá ser entregado al profesor al comienzo de la sesión.

3. DURANTE LA SESIÓN

3.1. Manejo básico del simulador con accesos a memoria

El simulador utilizado en el primer bloque temático (VisualMips32) también dispone de un módulo para poder visualizar los accesos a memoria.

Para acceder a él, debemos activar la opción “Vistas”→”Acceso a memoria” (tal como se indica en la imagen).

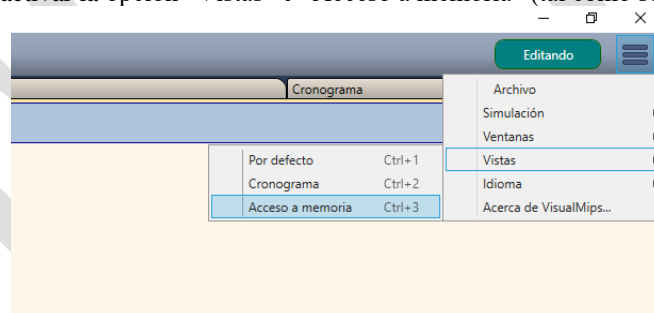


Figura 1. Acceso a las vistas predefinidas

Una vez seleccionemos la opción, la pantalla se dividirá añadiendo tres ventanas nuevas: una ventana para poder visualizar la memoria caché de datos, otra para la memoria caché de instrucciones, y una última para seguir la traza de accesos. Pueden observarse a continuación:

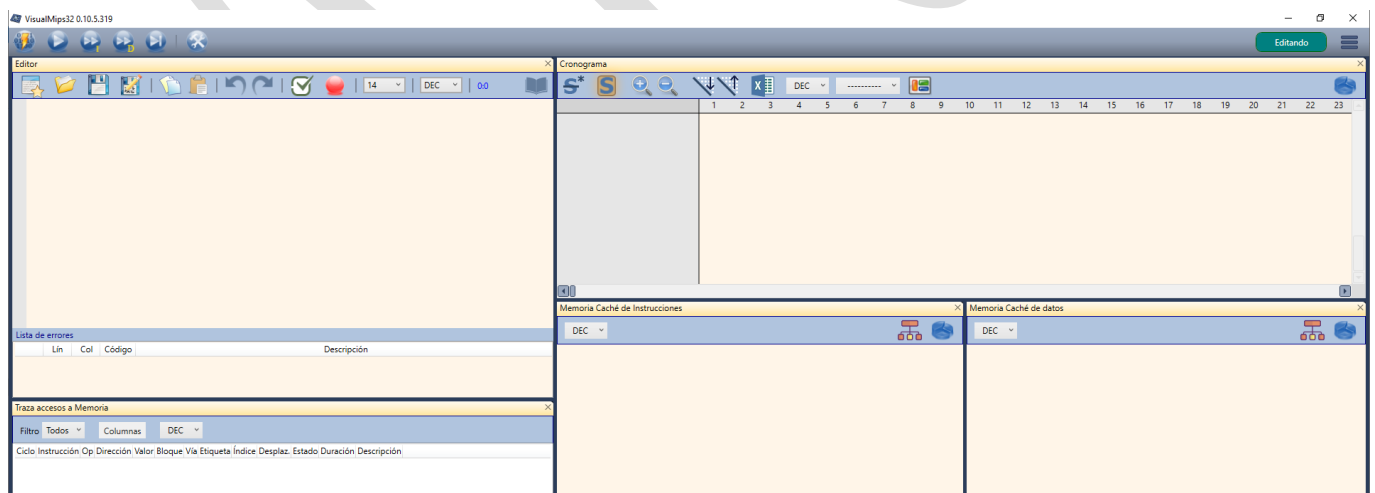


Figura 2. Vista predeterminada para los Accesos a memoria.

La configuración del sistema de memoria podemos realizarla de dos formas:

- A través de la pestaña 'Memoria' de la ventana "Propiedades": Estas opciones son las que el simulador utilizará por defecto (Figura 3).
- Incluyendo directivas de configuración (sección '.config') en nuestro programa: permiten simular un programa concreto utilizando una configuración modificada respecto de la configuración base especificada en la ventana "Propiedades". Puede ver la lista de directivas de configuración del simulador en la Guía Rápida que aparece en la ventana "Editor" (Figura 4).

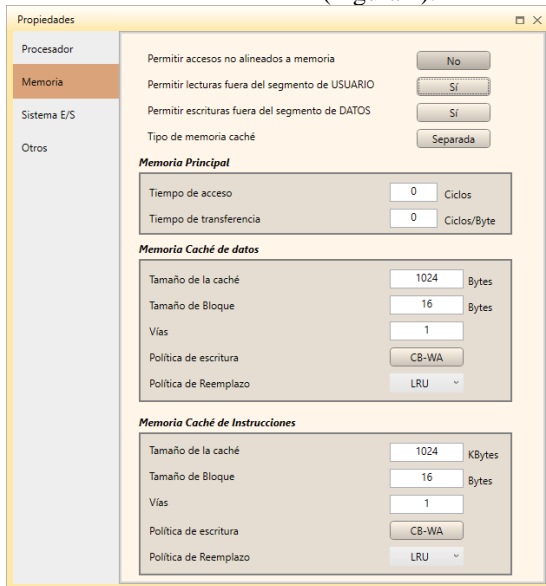


Figura 3. Configuración mediante la ventana "Propiedades"

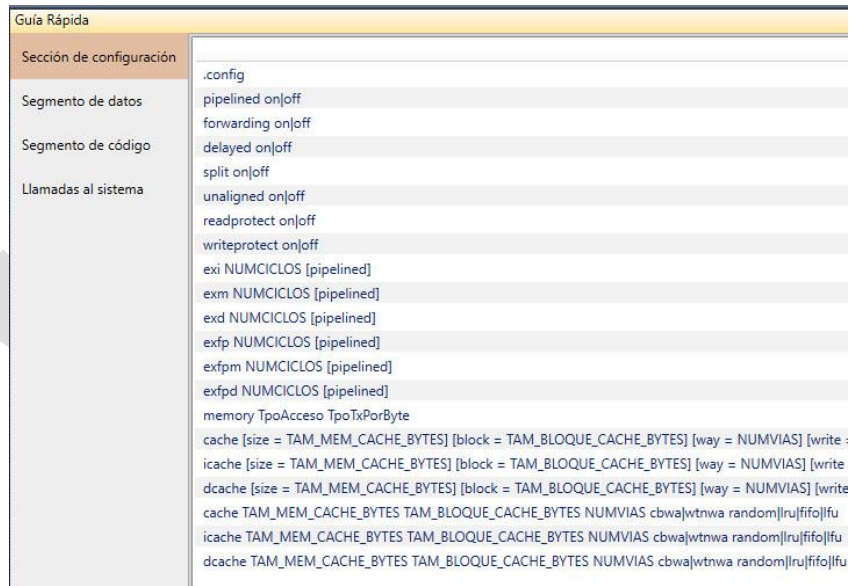


Figura 4. Lista de directivas de configuración según la Guía Rápida del programa

Desde ambas opciones puede configurar:

- Tiempos de acceso y transferencias** de la memoria principal (utilizados para el cálculo del tiempo medio de acceso a la jerarquía de memoria)
- Tamaño de caché, tamaño de bloque, número de vías, políticas de escritura y sustitución para cada una de las dos cachés:** Un par de apuntes al respecto:
 - Para que la caché sea de mapeado (o correspondencia) directo, deberemos seleccionar 1 vía.
 - Para que la caché sea totalmente asociativa, debemos seleccionar tantas vías como líneas tenga: para ello, dividimos su tamaño entre el tamaño de bloque (que nos indicaría el número de líneas); y ese mismo valor lo utilizamos para indicar el número de vías. Podemos ahorrarnos este cálculo si, en el desplegable, seleccionamos el valor máximo que nos permita.
[En esta sesión no haremos uso de políticas de escritura y sustitución, así que puede dejarlas tal cual]
- Permisos:** cerciórese que su sistema permite lecturas y escrituras fuera de los segmentos de usuario y datos, respectivamente. Esto nos facilitará la tarea de poder indicar direcciones de acceso a memoria sin incurrir en zonas protegidas del sistema.

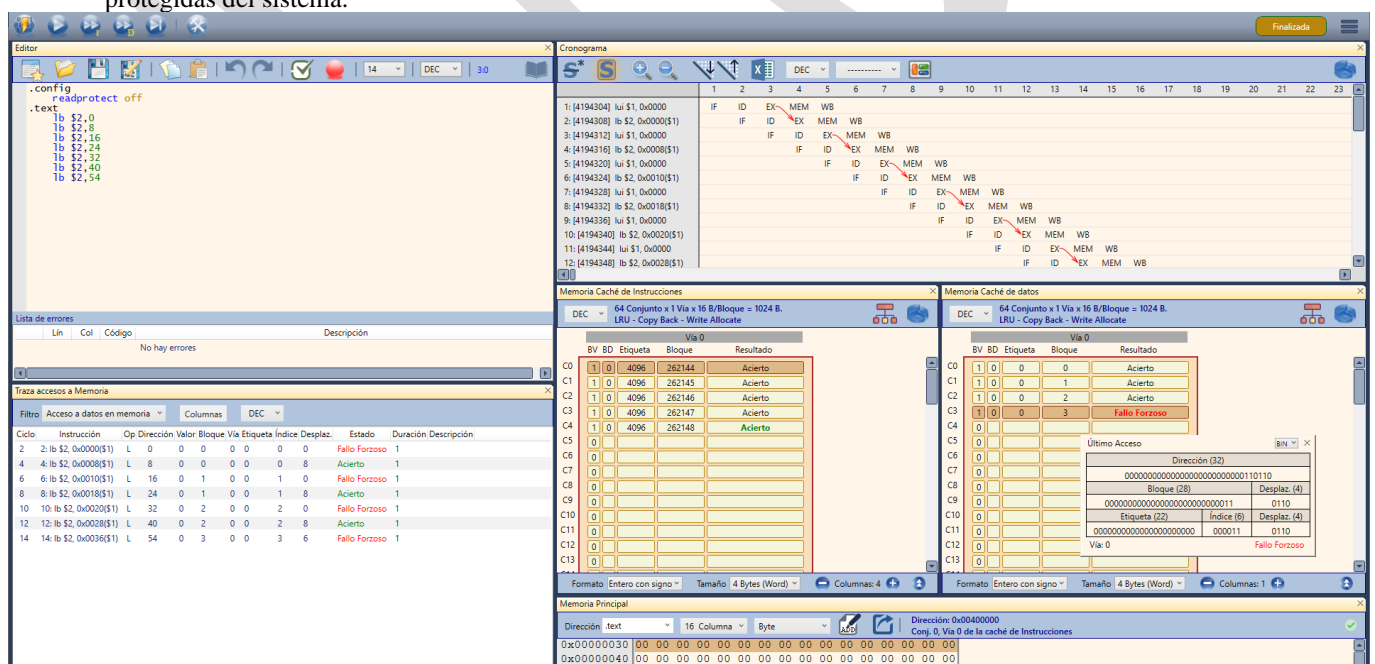


Figura 5. Vista 'Accesos a memoria' tras la ejecución del programa.

Una vez que tenga configurado su sistema, puede realizar accesos a memoria para comprobar el funcionamiento de su jerarquía. Para facilitar la tarea, puede filtrar en la ventana de traza por tipos de acceso: esto es útil porque, en esta primera sesión, únicamente trabajaremos con accesos a datos. En la figura 5 puede ver un ejemplo de ejecución:

3.2. Repaso del cálculo a mano para accesos a memoria

3.2.1. Mecanismo 1: Interpretando la dirección

Haciendo uso de la interpretación de cada uno de los campos de la dirección del procesador, se puede obtener la diferenciación entre el índice (de línea o conjunto), el desplazamiento y la etiqueta; además de la dirección o número de bloque de memoria principal.

Para ello, hay que, previamente, interpretar las direcciones:

- Para MP:
 - Dirección: tamaño del bus de direcciones
 - Desplazamiento (parte menos significativa de la dirección): tamaño de bloque.
 - Número (o dirección) de bloque: resto de bits de la dirección (parte más significativa).

Ejemplo: bus de direcciones de 16 líneas, bloque de 16 Bytes.

- $16\text{Bytes} = 2^4 \text{ Bytes} \rightarrow 4 \text{ bits de desplazamiento.}$

Número/Dirección de bloque (16-4 = 12 bits)	Desplazamiento (4 bits)
---	-------------------------

- Para MC:
 - Dirección: tamaño del bus de direcciones
 - Desplazamiento (parte menos significativa de la dirección): tamaño de bloque.
 - Índice de línea (si Map. Directo): tamaño de caché / tamaño de bloque
 - Índice de conjunto (si Asoc. Conjuntos): tamaño de caché / (tamaño de bloque * num. vías)
 - Etiqueta: resto de la dirección.
 - Ejemplo: bus de direcciones de 16 líneas, bloque de 16 Bytes, MC Asoc. Conj. de 2 vías de 256 Bytes.

- $16\text{Bytes} = 2^4 \text{ Bytes} \rightarrow 4 \text{ bits de desplazamiento.}$

- $256\text{Bytes} = 2^8 \rightarrow 2^8 / 2^4 = 2^4 \rightarrow 2^4 / 2^1 = 2^3 \rightarrow 3 \text{ bits de índice de conjunto}$

Etiqueta (16-4-3 = 9 bits)	Índice (3 bits)	Desplazamiento (4 bits)
----------------------------	-----------------	-------------------------

Con ello, una vez tenemos una dirección, se interpreta según los campos anteriores.

Ejemplo: Dirección 0x00A6

En MP:

Número/Dirección de bloque = 0x00A = bloque 10	Desplazamiento = 0x6 = byte 6
--	---

En MC:

Etiqueta = 0b000000001 = valor 1	Índice = 0b010 = conjunto 2	Desplazamiento = 0b0110 = byte 6
--	---------------------------------------	--

CONCLUSIONES:

- Este mecanismo precisa realizar previamente la interpretación de las direcciones.
- Este mecanismo precisa tener la dirección en hexadecimal.
- Es probable que necesitemos pasar a binario.

3.2.2. Mecanismo 2: Cálculo de forma analítica.

- Partiendo de la dirección de memoria en decimal, se divide ésta entre el tamaño de bloque obteniéndose el número de bloque (dirección de bloque) en el cociente y el desplazamiento en el resto.
- Partiendo posteriormente del número de bloque se divide entre el número de conjuntos (número de líneas en caso de mapeado directo) obteniéndose la etiqueta en el cociente y el índice en el resto.

Dirección de Memoria	Tamaño de Bloque	Nº de Conjuntos
	Nº de Bloque	Etiqueta
Desplazamiento	Índice	

Ejemplo: Bloque de 16Bytes, MC Asoc. Conj. de 2 vías de 256Bytes $\rightarrow 256\text{B} / 16 = 16 \text{ líneas}$, $16 / 2 = 8 \text{ conjuntos}$
Dirección 166 (0x00A6 en hexadecimal)

$166 / 16 = 10.375 \rightarrow \text{cociente} = 10 \text{ (bloque de MP)}, \text{resto} = 0.375 * 16 = 6 \text{ (desplazamiento)}$

$10 / 8 = 1.25 \rightarrow \text{cociente} = 1 \text{ (etiqueta)}, \text{resto} = 0.25 * 8 = 2 \text{ (conjunto)}$

CONCLUSIONES:

- Este mecanismo no precisa realizar previamente la interpretación de las direcciones.
- Este mecanismo precisa tener la dirección en decimal.

RECOMENDACIÓN GENERAL:

- Dirección en hexadecimal \rightarrow Mecanismo 1
- Dirección en decimal \rightarrow Mecanismo 2