Capítulo 4.

Decisiones de diseño

Una vez establecidos los requisitos, comenzó el proceso de implementación del simulador. En esta etapa debieron tomarse una larga serie de decisiones importantes, encaminadas a conseguir una herramienta muy didáctica, a la vez que altamente funcional.

4.1. Elección del lenguaje de programación

Una de los requisitos básicos del simulador era que fuera una herramienta muy visual, de manera que dentro de las opciones de programación posibles se contemplaron:

- Utilizar *programación web*: pese a las ventajas de portabilidad, distribución y accesibilidad del programa que proporciona la programación web, la velocidad de ejecución que supondría el uso de lenguajes interpretados como Java mermarían la funcionalidad del programa.

- Utilizar un *lenguaje visual* como Visual Basic, Borland Delphi Pascal, Borland C++ Builder, etc… donde se perdía en portabilidad (aplicaciones Windows, aunque también para Linux gracias a Borland Kylix) pero se ganaba en funcionalidad, velocidad de ejecución y eficiencia en el uso de memoria.

Al final la elección se decantó hacia Borland C++ Builder, principalmente por las siguientes características:

- *Basado en C++,* un lenguaje de programación orientado a objetos muy robusto y eficiente.

- *Entorno de desarrollo (IDE) muy completo,* conteniendo componentes que facilitan la tarea de desarrollo.

- *Fácil de modular,* gracias a su naturaleza orientada a objetos.

- *Fácil de depurar,* gracias a las múltiples herramientas que incluye.

4.2. Características del simulador

Como se mencionó en el capítulo 2, son muchos los conceptos involucrados en una jerarquía de memoria, así que teniendo en cuenta que se debía implementar un simulador lo más didáctico y funcional posible, se optó por dividirlo en dos grandes partes:

- *Traducción de direcciones*: Aquí se contemplan todos los conceptos asociados a los mecanismos de traducción de direcciones virtuales a direcciones reales. En el simulador se implementaron 3 mecanismos:
    - *Traducción de direcciones por transformación directa*
    - *Traducción de direcciones por transformación asociativa-directa con TLB*
    - *Traducción de direcciones por transformación asociativa-directa con TLB dividida en datos e instrucciones*
- *Búsqueda de páginas*: Aquí se contemplan todos los conceptos asociados a la búsqueda de páginas y bloques en la jerarquía de memoria. En el simulador se implementa una jerarquía de memoria paginada con caché multinivel que incluye:
    - *Memoria secundaria*
    - *Memoria principal*
    - *Caché de nivel 3*
    - *Caché de nivel 2*
    - *Caché de nivel 1 conjunta o separada en datos e instrucciones*

En esta sección del simulador se contemplan las siguientes estrategias:

- Para el sistema de memoria virtual
    - *Búsqueda*: Paginación por demanda y paginación anticipada cargando la página siguiente a la referenciada.
    - *Colocación:* FIRST FIT
    - *Reemplazamiento:* FIFO (First In – First Out)
- Para los diferentes niveles de caché

- *Colocación:* Mapeado directo, completamente asociativa y asociativa por conjuntos.
- *Búsqueda:* División de bloques en índice (Index) y etiqueta (Tag).
- *Reemplazamiento:* Azar, FIFO, LRU, Clock, LFU, NUR.
- *Coherencia:* Escritura directa y escritura retardada.

Dado que en la búsqueda de páginas también es necesaria la traducción de direcciones, ésta se implementó como un módulo separado dentro del código de la aplicación. Así, cuando el usuario elige la *Búsqueda de páginas*, el simulador efectúa la traducción de direcciones de forma transparente al usuario proporcionándole directamente la dirección real, aunque si tiene dudas sobre el origen de ésta puede dirigirse a la *Traducción de direcciones*.

4.3. Control de la simulación

Otra de las decisiones importantes que se hubo de tomar fue determinar cómo proporcionar al usuario un control de la simulación que fuese funcional y cómodo a la vez.

La solución adoptada fue dividir la simulación en una serie de pasos que mostrasen el estado de la memoria en ese instante destacando las posiciones afectadas, la acción realizada o el suceso ocurrido, y una serie de estadísticas.

Además, se optó por incluir una novedad dentro de los simuladores, la posibilidad de ir tanto hacia adelante como hacia atrás. Este método permite que los usuarios, en el caso de que den un paso hacia adelante y no comprendan lo sucedido, puedan volver atrás uno o varios pasos, para tener una visión global del suceso.

Finalmente, dado que los ficheros de traza pueden ser bastante largos, se incluyó también un modo automático en el que el simulador da pasos a intervalos de tiempo fijos, que además el usuario puede modificar.

Captura 4.1. Control de la simulación

4.4. Asistente

Una de las características principales del simulador es la gran cantidad de parámetros configurables (tamaños de las diferentes memorias, bloques y páginas,

tiempos de acceso, caches habilitadas o deshabilitadas, algoritmos de colocación, reemplazamiento y coherencia, etc…) que le otorgan una elevada funcionalidad.

Sin embargo, esta gran cantidad de parámetros podría llegar a confundir al usuario. Por lo que se buscó una manera de guiar al usuario durante el proceso, optando por crear un asistente que lo ayudara en el proceso de configuración.

Dado que además una vez configurado el simulador, se debían de elegir los ficheros de traza entre múltiples opciones y posteriormente el tipo de simulación a realizar, se dividió al asistente en tres grandes fases:

- *Configurar,* a través de una serie de pasos, guía al usuario en el proceso de configurar todos los parámetros del simulador.

- *Cargar fichero,* aquí el usuario elige el fichero de traza entre los múltiples ejemplos incluidos en el simulador o bien, los propios ficheros creados por él mismo.

- *Simular,* aquí el usuario elige el tipo de simulación a realizar.

Captura 4.2. Asistente

Además se incluye el botón *Resumen configuración*, que muestra en cualquier momento un resumen de la configuración elegida durante el proceso de configuración.

Captura 4.3. Resumen de configuración

4.4.1. Asistente - Configurar

Para guiar al usuario en el proceso de configurar el simulador, se agruparon los parámetros según la siguiente secuencia:

- *Memoria virtual / secundaria*

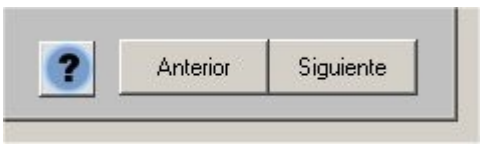
- *Memoria principal*

- *Tamaño de página*

- *Mecanismo de traducción de direcciones*
- *Parámetros de paginación*
- *Tamaño del bloque de caché*
- *Caché de nivel 3* (Tamaño, organización, número de vías, política de reemplazamiento)
- *Caché de nivel 2* (Tamaño, organización, número de vías, política de reemplazamiento)
- *Caché de nivel 1* (Tamaño, organización, número de vías, política de reemplazamiento)

De esta manera, los grupos se van mostrando al usuario en una serie ordenada de pantallas, donde el usuario rellena los parámetros y puede:

- Pasar a la *Siguiente* pantalla: los datos serán validados, si son erróneos se mostrarán al usuario los errores para que los pueda resolver, y si son correctos, se mostrará la nueva pantalla con los datos a rellenar.
- Volver a la pantalla *Anterior*: si el usuario se percata de que ha cometido algún error o ha cambiado de parecer, puede volver a la pantalla anterior y cambiar aquellos parámetros que considere necesarios.
- Solicitar *Ayuda*: en todo momento el usuario tendrá disponible un botón de ayuda sobre el que pulsar para acceder a la sección de la ayuda correspondiente a la pantalla que está viendo en ese instante.



Captura 4.4. Resumen de configuración

Dado que rellenar a mano cada uno de los parámetros podría resultar una tarea tediosa, se incluyó la opción de cargar ficheros de configuración creados a mano por el usuario o incluidos como ejemplo y que contuviesen los parámetros necesarios para la simulación.

4.4.1.1. Ficheros de configuración

Los ficheros de configuración son ficheros de texto con extensión .cfg que contienen todos los parámetros necesarios para realizar la simulación. Estos ficheros pueden cargarse al inicio del asistente, de manera que los parámetros de las

diferentes pantallas son introducidos automáticamente, aunque el usuario tiene la posibilidad de cambiarlos.

La estructura de un fichero de configuración que contenga todos los posibles parámetros sería la siguiente:

```
/Tamaño total de memoria secundaria (en Kbytes)
-1:
/Tiempo acceso a memoria secundaria (en ciclos)
-2:
/Ancho bus Memoria Secundaria-Memoria principal (en bits)
-3:
/Tiempo acceso bus Memoria Secundaria-Memoria principal (en ciclos)
-4:
/Tamaño total memoria principal (en Kbytes)
-5:
/Tamaño página (en Kbytes)
-6:
/Tiempo acceso memoria principal (en ciclos)
-7:
/Traducción direcciones
/(1 - Directa, 2 - Asociativa directa con 1 TLB, 3 - Asociativa directa con 2 TLB)
-8:
/Número de entradas de la TLB
-9:
/Tiempo de acceso a la TLB (en ciclos)
-10:
/Tipo de paginación 1
/(1 - Por demanda, 2 - Anticipada)
-11:
/Tipo de paginación 2
/(1 - Escritura directa, 2 - Escritura retardada)
-12:
/Caché habilitada
/(0 - NO, 1 - SI)
-13:
/Tamaño del bloque de caché (en bytes)
-14:
/Caché de nivel 3 habilitada?
/(0 - NO, 1 - SI)
-15:
/Organización de la cache de nivel 3
/(1 - Mapeado directo, 2 - Completamente asociativa, 3 - Asociativa por conjuntos)
-16:
/Política de reemplazamiento de cache de nivel 3
/(1 - AZAR, 2 - FIFO, 3 - LRU, 4 - LFU, 5 - NUR, 6 - CLOCK)
-17:
/Tamaño de la caché de nivel 3 (en Kbytes)
-18:
/Tiempo de acceso a la caché de nivel 3 (en Ciclos)
-19:
/Ancho bus memoria caché nivel 3 a nivel superior (en Bits)
-20:
/Tiempo acceso bus memoria caché nivel 3 a nivel superior (en Ciclos)
-21:
/Número de vías de la caché de nivel 3 (si es asociativa por conjuntos)
-22:
/Caché de nivel 2 habilitada?
```

```
/(0 - NO, 1 - SI)
-23:
/Organización de la cache de nivel 2
/(1 - Mapeado directo, 2 - Completamente asociativa, 3 - Asociativa por conjuntos)
-24:
/Política de reemplazamiento de cache de nivel 2
/(1 - AZAR, 2 - FIFO, 3 - LRU, 4 - LFU, 5 - NUR, 6 - CLOCK)
-25:
/Tamaño de la caché de nivel 2 (en Kbytes)
-26:
/Tiempo de acceso a la caché de nivel 2 (en Ciclos)
-27:
/Ancho bus memoria caché nivel 2 a nivel superior (en Bits)
-28:
/Tiempo acceso bus memoria caché nivel 2 a nivel superior (en Ciclos)
-29:
/Número de vías de la caché de nivel 2 (si es asociativa por conjuntos)
-30:
/Caché de nivel 1 habilitada?
/(0 - NO, 1 - SI)
-31:
/Tipo de caché de nivel 1
/(1 - Conjunta, 2 - Separada)
-32:
/Organización de la cache de nivel 1
/(1 - Mapeado directo, 2 - Completamente asociativa, 3 - Asociativa por conjuntos)
-33:
/Política de reemplazamiento de cache de nivel 1
/(1 - AZAR, 2 - FIFO, 3 - LRU, 4 - LFU, 5 - NUR, 6 - CLOCK)
-34:
/Tamaño de la caché de nivel 1 (en Kbytes)
-35:
/Tiempo de acceso a la caché de nivel 1 (en Ciclos)
-36:
/Ancho bus memoria caché nivel 1 a nivel superior (en Bits)
-37:
/Tiempo acceso bus memoria caché nivel 1 a nivel superior (en Ciclos)
-38:
/Número de vías de la caché de nivel 1 (si es asociativa por conjuntos)
-39:
```

Ejemplo 4.1. Fichero de configuración

Las líneas que comienzan con '/' se consideran líneas de comentario y no son necesarias para el correcto funcionamiento del fichero.

Cómo se puede observar, son muchos los parámetros que se pueden configurar para una simulación. Sin embargo, no todos son necesarios en algunos casos, ya que pueden existir ficheros de configuración más cortos.

Dentro del simulador se incluyeron una serie de ficheros de configuración de ejemplo para *Traducción de direcciones* y *Búsqueda de páginas*. El contenido de los ficheros puede consultarse en la ayuda del programa.

4.4.2. Asistente – Cargar fichero

Para realizar una buena simulación es necesario tener ficheros de traza que se correspondan con programas reales.

La solución adoptada fue obtenerlos a través del portal de distribución de trazas del PEL (Performance Evaluation Laboratory) de la Bringman Young University en Washington[1].

Este portal se actualiza frecuentemente con nuevas trazas de programas, en su mayor parte benchmarks, ejecutados sobre plataformas Windows y Linux que cubren los diferentes tipos de aplicaciones más utilizadas en computación: compiladores, editores, traductores, bases de datos, programas cad, diseño 3D, servidores web, de correo y ftp, etc…

4.4.2.1 Ficheros de traza

Dado que se trataba de efectuar simulaciones que mostraran las características de diferentes tipos de programas se optó por incluir dentro del simulador las trazas correspondientes a 6 tests del benchmark SPEC2000[2].

	Descripción
Crafty	Programa de ajedrez
Eon cook	Programa de trazado de rayos sobre una tetera
Facerec	Programa de reconocimiento de caras
Gcc	Compilador de C y C++
Mesa	Librería de gráficos 3D
Vortex	Base de datos

Tabla 4.1. Ficheros de traza de ejemplo

De cada uno de estos 6 ficheros se realizaron 5 versiones, para poder así contar con trazas en diferentes formatos de direcciones.

Terminado en	***Direcciones***	***Memoria virtual***
a	32 bits	Hasta 4GB
b	28 bits	Hasta 256MB
c	24 bits	Hasta 16MB
d	20 bits	Hasta 1 MB
e	16 bits	Hasta 64KB

Tabla 4.2. Formatos de direcciones

[1] El portal se encuentra en la dirección web http://tds.cs.byu.edu/tds/

[2] Más información sobre el benchmark puede encontrarse en la dirección web http://www.spec.org [10]

Estos ficheros, que no son más que ficheros de texto con extensión .trd, contienen 100 direcciones virtuales referenciadas por el programa, según la siguiente estructura:

- DIRECCION: Contiene una dirección virtual de X bit que representa la dirección generada por el programa en ejecución.

- TIPO DE ACCESO: Indica el tipo de acceso a memoria, que puede ser:

  - FETCH - Búsqueda de instrucciones.
  - MEMREAD - Lectura de memoria.
  - MEMWRITE - Escritura en memoria.

- TIEMPO: Indica el tiempo (Delta) transcurrido desde la última petición de página en ciclos de reloj.

```
06709ea0    MEMWRITE    103
0776acf8    MEMREAD     263
07781770    FETCH       143
0776aa40    MEMREAD     119
```

Ejemplo 4.2. Extracto de un fichero de traza

Dado que no siempre es posible obtener buenos ficheros de traza que contengan la serie de propiedades que queremos estudiar, se optó por incluir además otro tipo de ficheros, los ficheros de página. Éstos, también son ficheros de texto, aunque con extensión .trp, que siguen la siguiente estructura:

- PÁGINA: Contiene la página referenciada por el programa en ejecución.
- TIPO DE ACCESO: Indica el tipo de acceso, que puede ser:

  - FETCH - Búsqueda de instrucciones.
  - MEMREAD - Lectura de memoria.
  - MEMWRITE - Escritura en memoria.

- DIRECCION: Contiene una dirección virtual de X bit que representa la dirección generada por el programa en ejecución. En caso de no conocerla puede ser 0 siempre.

Estos ficheros pueden ser creados fácilmente por el usuario para estudiar el comportamiento, por ejemplo de páginas adyacentes.

```
0    FETCH       1821
1    MEMREAD     37
2    MEMREAD     1792
3    FETCH       29
4    MEMWRITE    15853
1    MEMREAD     37
```

Ejemplo 4.3. Extracto de un fichero de páginas

4.4.3. Asistente – Simular

Una vez realizada la configuración y elegido el fichero de traza es el momento de comenzar la simulación. En este paso del asistente elegimos el tipo de simulación a realizar.

4.5. Simulación

Cómo se describió en la sección 4.2 la simulación se divide en dos partes:

- *Traducción de direcciones.*
- *Búsqueda de páginas.*

A continuación vamos a pasar a detallar cada una de ellas.

4.5.1. Traducción de direcciones

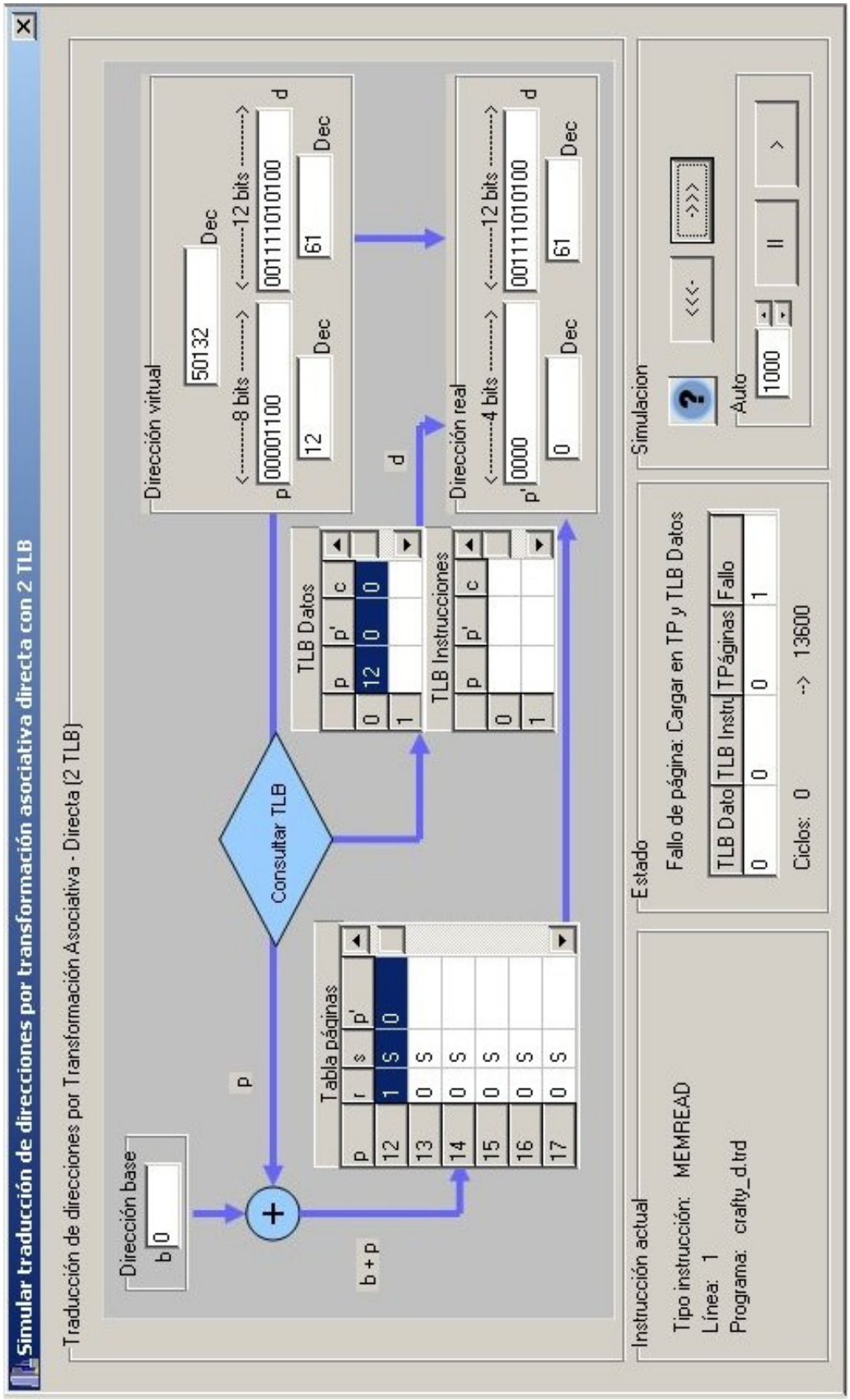
Dependiendo de la configuración elegida, se pueden mostrar tres interfaces diferentes, uno para cada tipo de mecanismo de traducción de direcciones:

- *Traducción de direcciones por transformación directa*
- *Traducción de direcciones por transformación asociativa-directa con TLB*
- *Traducción de direcciones por transformación asociativa-directa con TLB dividida en datos e instrucciones*

Todos ellos comparten los siguientes elementos:

- *Dirección virtual*, tanto en binario como en decimal, dividida en parte p (página dentro de la tabla de páginas) y parte d (desplazamiento dentro de la página).
- *Dirección real,* tanto en binario como en decimal, dividida en parte p' (página en memoria principal) y parte d (desplazamiento dentro de la página).
- *Dirección base* de la tabla de páginas.
- *Tabla de páginas*

A la que los dos últimos tipos de traducción añaden la *TLB* bien conjunta bien dividida en parte de datos y parte de instrucciones.



Captura 4.5. Traducción de direcciones

4.5.2. Búsqueda de páginas

La estructura de la jerarquía de memoria implementada se muestra en la siguiente figura:

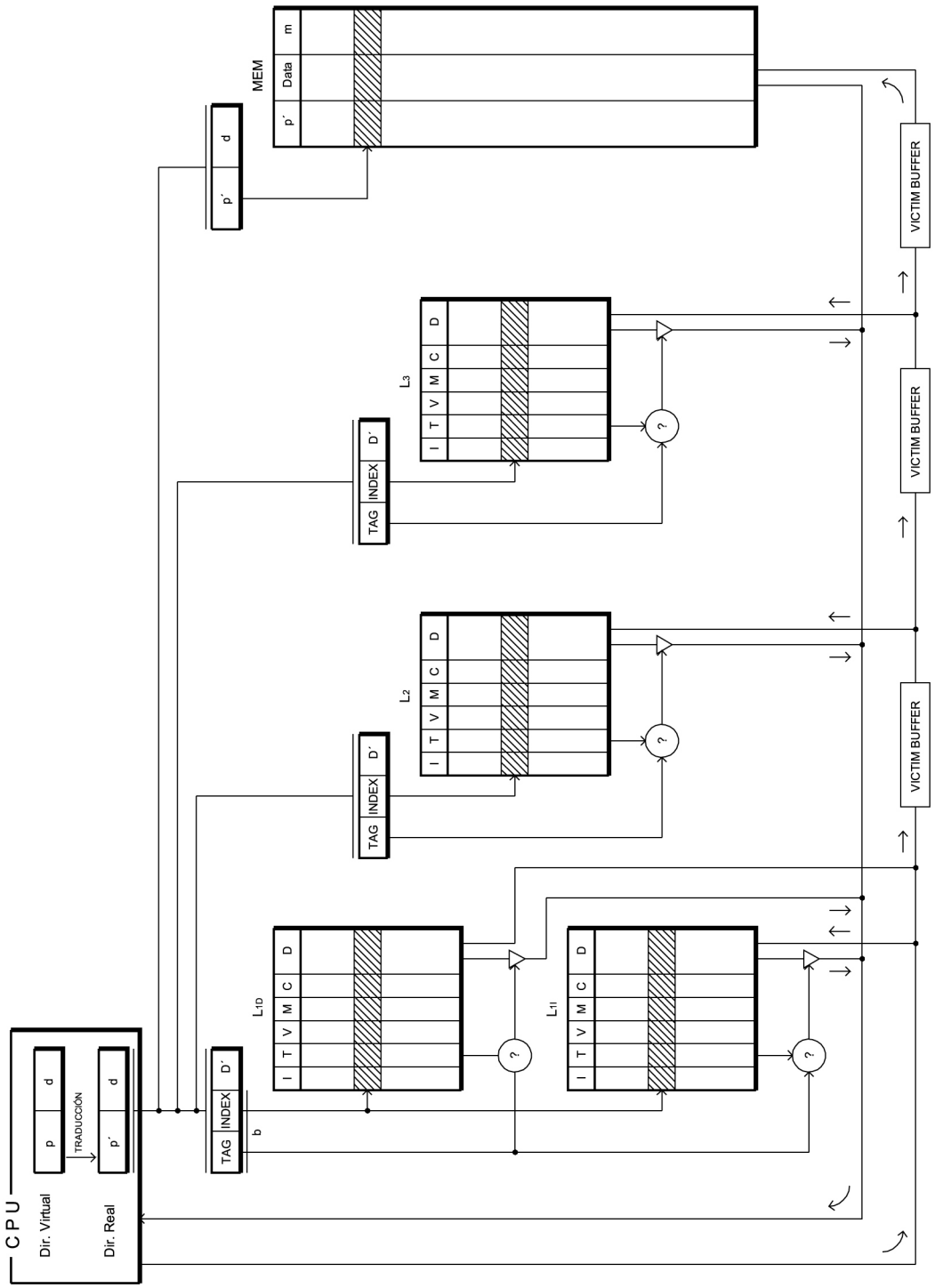


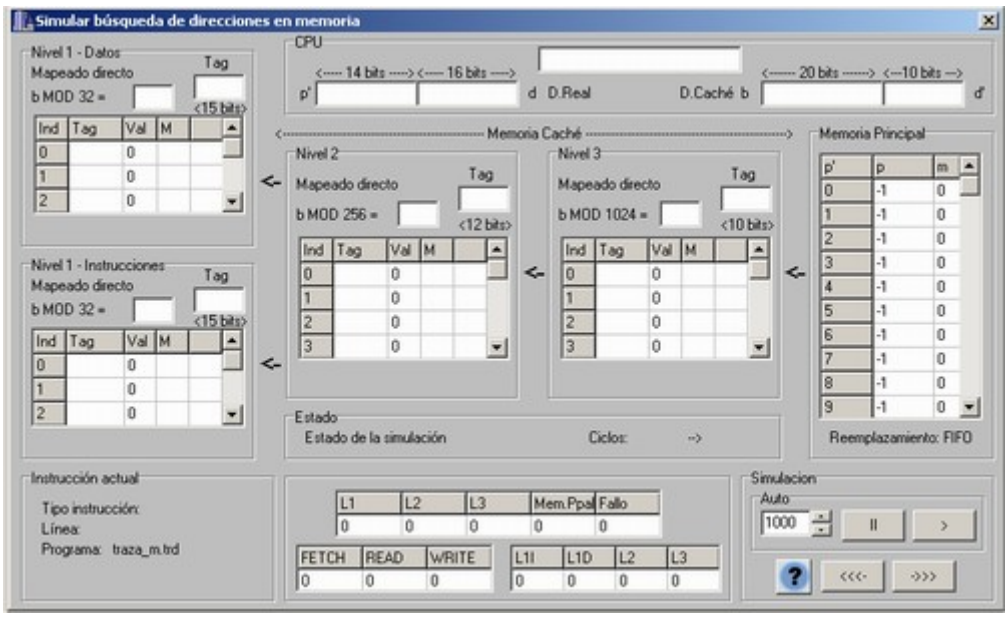
Figura 4.1. Estructura de la jerarquía de memoria implementada

Si elegimos la estrategia de coherencia por escritura retardada debemos tener en cuenta que el grupo de victim buffers de la figura no está presente.

Dependiendo de la configuración elegida, puede llegar a tener los siguientes componentes:

- *Memoria secundaria*
- *Memoria principal*
- *Caché de nivel 3*
- *Caché de nivel 2*
- *Caché de nivel 1 conjunta o separada en datos e instrucciones*
- *CPU*

Habilitando todos los niveles de caché la pantalla mostrada sería esta:



Captura 4.6. Búsqueda de páginas

Donde se puede observar la parte correspondiente a la CPU que incluye:

- Dirección real, en binario y en decimal dividida en parte p (página real) y parte d (desplazamiento dentro de la página).
- Dirección caché, en decimal dividida en parte b (bloque) y d' (desplazamiento dentro del bloque).

4.6. Estrategias

El simulador tiene implementadas múltiples estrategias para el sistema de memoria virtual y caché, a continuación vamos a detallar su implementación.

4.6.1. Estrategias de administración del sistema de memoria virtual

El simulador implementa las siguientes estrategias para el sistema de memoria virtual:

- *Estrategias de búsqueda*: Paginación por demanda y paginación anticipada cargando la página siguiente a la referenciada.

- *Estrategias de colocación:* FIRST FIT, la página nueva que entra a memoria principal se coloca en la primera posición libre.

- *Estrategias de reemplazamiento:* FIFO (First In – First Out), en caso de que todas las posiciones de memoria principal estén llenas se reemplaza la primera página que entró.

4.6.2. Estrategias de administración de la memoria caché

Para la memoria caché se implementan el siguiente conjunto de estrategias:

- *Estrategias de colocación:*

    - *Mapeado directo: el cálculo de la posición se muestra sobre la memoria.*

    - *Completamente asociativa*

    - *Asociativa por conjuntos: el cálculo del conjunto se muestra sobre la memoria.*

Captura 4.7. Cálculo de la posición o conjunto

- *Estrategias de búsqueda:* las direcciones de bloque se dividen en index y tag, mostrándose el tag buscado en un recuadro por encima de la memoria.

Captura 4.8. Etiqueta (Tag) de la caché

- *Estrategias de reemplazamiento:*

  - *AZAR:* El bloque a reemplazar se elige al azar.

  - *FIFO (First In – First Out):* El bloque a reemplazar es aquel que lleva más tiempo en memoria.

    Se implementa usando un contador de 0 a Num bloques, de manera que los nuevos bloques entran a 0, aumentando el contador en 1 del resto de los bloques. Así el bloque con el contador más alto es el que lleva más tiempo en memoria, por lo que es el elegido para ser reemplazado.

  - *LRU (Least Recently Used):* El bloque a reemplazar es aquel que lleva más tiempo en memoria sin ser usado.

    Se implementa usando un contador de 0 a Num bloques, de manera que los nuevos bloques entran a 0, aumentando el contador en 1 del resto de los bloques. Además cuando un bloque es referenciado su contador se pone de nuevo a 0 y se incrementan en 1 aquellos contadores que sean menores. Así el bloque con el contador más alto es el que lleva más tiempo en memoria sin ser usado, por lo que es el elegido para ser reemplazado.

  - *CLOCK aproximación a LRU:* El bloque a reemplazar es aquel que lleva más tiempo en memoria sin ser usado.

    Se implementa sin necesidad de contador, tan sólo se necesita un puntero que determina la posición siguiente a ser reemplazada. La posición actual de este puntero se muestra en un recuadro bajo la memoria.

Captura 4.9. Puntero de la política de reemplazamiento Clock

El puntero avanza una posición, y si esa posición tiene el bit de modificado a 1, indicando que ha sido modificada recientemente,

lo pone a 0 y avanza a la siguiente posición. Si tiene el bit a 0 esa es el bloque elegido para ser reemplazado.

- *LFU (Least Frequently Used):* El bloque a reemplazar es aquel que ha sido usado el menor número de veces.

  Se implementa utilizando un contador que se inicializa a 0 cuando el bloque entra en memoria y se incrementa en 1 cada vez que el bloque es referenciado.

- *NUR (Not Used Recently):* El bloque a reemplazar es aquel que no ha sido usado recientemente. Para determinar esta situación se basa en dos factores: si el bloque ha sido referenciado y si ha sido modificado, dando lugar a cuatro posibles situaciones:

  - 00 (0) - Página no referenciada y no modificada
  - 01 (1) - Página no referenciada pero modificada
  - 10 (2) - Página referenciada pero no modificada
  - 11 (3) - Página referenciada y modificada

  Para implementarlo se utiliza un contador que se inicializa a 0 cuando el bloque entra en memoria por un acceso a instrucciones FETCH o una lectura de memoria MEMREAD, y a 1 si entra por una escritura en memoria MEMWRITE. Posteriormente, si el bloque es referenciado pasa de 0 a 2 o de 1 a 3, y si es modificado, de 2 a 3. Así el bloque elegido para ser reemplazado es el que tiene el contador más bajo.

- *Estrategias de coherencia:*

  - *Write Through o Escritura directa:* Una escritura en memoria actualiza tanto la copia en memoria principal como la copia en memoria caché.

    Cuando un bloque es referenciado por una escritura a memoria MEMWRITE se modifican los bits de modificado de los bloques de todos los niveles de caché.

  - *Write Back o Escritura retardada*: Una escritura en memoria actualiza sólo la copia en caché, mientras que la copia de la memoria principal se actualizará cuando el bloque de caché sea reemplazado.

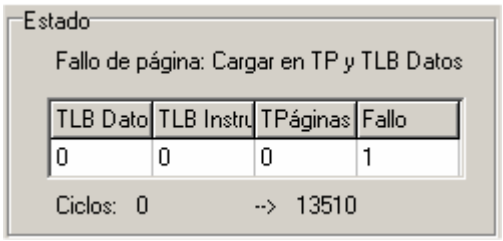
    Cuando un bloque es referenciado por una escritura MEMWRITE el bit de modificado de ese nivel de caché pasa a 1, y cuando sea reemplazado se modificará el de la memoria principal.

4.7. Estadísticas

Según el tipo de simulación elegido se muestran las siguientes estadísticas.

4.7.1. Estadísticas de traducción de direcciones

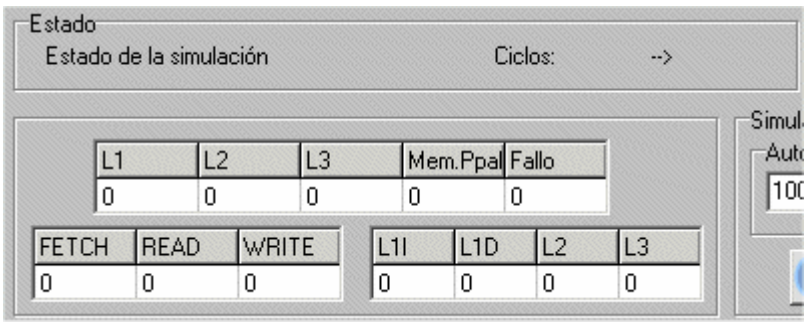
- Número de fallos de página.
- Número de aciertos en la tabla de páginas.
- Número de aciertos en la TLB conjunta o en la TLB de datos y la TLB de instrucciones.



Captura 4.10. Estado, estadísticas y ciclos de traducción de direcciones

4.7.2. Estadísticas de búsqueda de páginas

- Número de aciertos en los diferentes niveles de la jerarquía.
- Número de reemplazamientos en los diferentes niveles de la jerarquía.
- Número de instrucciones de cada tipo.



Captura 4.11. Estado, estadísticas y ciclos de búsqueda de páginas

4.8. Ciclos

Junto a las estadísticas y el estado de la máquina se muestran los ciclos que transcurren en cada paso de la simulación, mostrando los ciclos transcurridos actualmente frente a los del último paso.

4.8.1. Ciclos en traducción de direcciones

- *Traducción de direcciones por transformación directa*

- *Cargar a memoria principal*
  Ciclos = Tacceso memoria principal (consulta tabla páginas) + Tacceso memoria secundaria + Tacceso bus + Tacceso memoria principal

- *Acierto en tabla de páginas*
  Ciclos = Tacceso memoria principal (consulta tabla páginas)

- *Traducción de direcciones por transformación asociativa-directa con TLB*

  - *Cargar a memoria principal*
    Ciclos = TaccesoTLB + TAcceso memoria principal + Tacceso memoria secundaria + Tacceso bus + Tacceso memoria principal

  - *Acierto en tabla de páginas*
    Ciclos = TaccesoTLB + TAcceso memoria principal (consulta tabla páginas)

  - *Acierto en TLB*
    Ciclos = TaccesoTLB

4.8.2. Ciclos búsqueda de páginas

- *Cargar a memoria principal*
  Ciclos = TAcceso memoria principal + Tacceso memoria secundaria + + Tacceso bus

- *Acierto en caché de nivel 1*
  Ciclos = TAcceso cache L1

- *Acierto en caché de nivel 2*
  Ciclos = Tacceso cache L1 + Tacceso cache L2 + Tacceso bus L1-L2 + + Tacceso cache L1

- *Acierto en caché de nivel 3*
  Ciclos = Tacceso cache L1+ Tacceso cache L2 + Tacceso cache L3 + + Tacceso bus L3-L2 + Tacceso cache L2 + Tacceso bus L2-L1 + + Tacceso cache L1
- *Acierto en memoria principal*
  Ciclos = Tacceso cache L1+ Tacceso cache L2 + Tacceso cache L3 + + Tacceso memoria principal + Tacceso bus MemPpal-L3 + + Tacceso cache L3 +Tacceso bus L3-L2 + Tacceso cache L2+ + Tacceso bus L2-L1 + Tacceso cache L1

Además, cuando usamos la escritura retardada se añade una penalización por el tiempo necesario para actualizar los diferentes niveles de caché, que en el peor de los casos es:

Ciclos = Tacceso cache L1+ Tacceso cache L2 + Tacceso cache L3 + + Tacceso memoria principal