Getting Started with SMPCache 2.0

1. Introduction

SMPCache is a trace-driven simulator for cache memory systems on symmetric multiprocessors (SMPs) which use bus-based shared memory. This simulator operates on PC systems with Windows, and it offers a Windows typical graphic interface.

Some of the parameters you can study with the simulator are: program locality; influence of the number of processors, cache coherence protocols, schemes for bus arbitration, mapping, replacement policies, cache size (blocks in cache), number of cache sets (for set associative caches), number of words by block (block size), ...

Because of its easy and user-friendly interface, the simulator is recommended for teaching purposes; since it allows to observe, in a clear and graphic way, how the multiprocessor evolves as the execution of the programs goes forward (the memory traces are read).

1.1. Prerequisites

To make sense of the rest of *Getting Started with SMPCache*, you should be familiar with some theoretical considerations concerning cache memory systems, and particularly regarding their use in multiprocessor environments. These concepts are widely discussed in many computer architecture texts (like the William Stallings' *Computer Organization and Architecture*), and we will not mention them here. All operations and algorithms we use are similar to those found in these computer architecture texts. As a consequence, the results obtained with the simulator are very close to the real world.

1.2. Suggestions?

If you have comments about this guide to SMPCache or about the simulator, please contact Miguel A. Vega at mavega@unex.es (Fax: +34-927-257202) or at the following address:

Miguel A. Vega-Rodríguez

Dept. de Informática, Univ. de Extremadura, Escuela Politécnica

Campus Universitario s/n. 10071. Cáceres. España (Spain)

2. Installation

In order to begin the installation you must execute the Setup.exe program included in your copy of SMPCache. Then, follow the directions below and on the screen during the installation process:

1. Exit *all* Windows applications prior to continuing with the installation.
2. When the "Welcome" screen appears, read it. Click **Next** to continue.

3. Enter your name and company. Click **Next** to continue.
4. Select the location in which you want to install SMPCache. Choose the default folder or click **Browse** to select a different location or to enter your own folder name. Click **Next** to continue.
5. Click the type of setup you prefer, then click **Next** to continue.
   a) Typical — installs all program files to the location you selected in step 4. Recommended for most users.
   b) Compact — reduces the disk space required for the installation because it does not install the sample files.
   c) Custom — provides you with options on installing the sample files or help files.
6. If you select the Custom setup, then you must choose the exact components you want to install. Click **Next** to continue.
7. Select the program folder for SMPCache. Choose the default program folder or type a new folder name. You can also select one from the Existing Folders list. Click **Next** to continue.
8. Before starting copying files, the installation process shows you the current settings. If you want to change any settings click **Back**, otherwise click **Next** to begin copying files.
9. SMPCache will now finish being installed. Click **Finish** to complete the installation.

Once installation is complete, a SMPCache group, which includes the application icon, is created. You can then create a shortcut to SMPCache on your desktop.

☞**Remember**: If for any reason you wish to stop the installation, click **Cancel** and the installation of SMPCache will be terminated.

2.1. Uninstalling SMPCache

To uninstall the simulator:

1. Click the Windows **Start** button.
2. Click **Settings** and **Control Panel**.
3. Click **Add/Remove Programs**. The Add/Remove Programs Properties screen appears.
4. From the **Install/Uninstall** list, select **SMPCache** and select **Add/Remove**. After your confirmation, SMPCache will be removed from your computer.

3. Configuration Files

The simulator allows you to select the different choices for configuring a given architecture (see Table 1). The different choices selected may be stored on an ASCII data file

(configuration file, which has the extension ".cfg") for future loading, so the need to make many selections for configuring the same architectural model is avoided.

Processors in SMP	1, 2, 3, 4, 5, 6, 7 or 8
Cache coherence protocols	MSI, MESI or DRAGON
Schemes for bus arbitration	Random, LRU or LFU
Word wide (bits)	8, 16, 32 or 64
Words by block	1, 2, 4, 8, 16, 32, 64, 128, 256, 512 or 1024
Blocks in main memory	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152 or 4194304
Blocks in cache	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 or 2048
Mapping	Direct, Set-Associative or Fully-Associative
Cache sets (for set associative caches)	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 or 2048
Replacement policies	Random, LRU, FIFO or LFU
Writing strategies	Write-Back (for cache coherence protocols)
Cache levels in the memory hierarchy	1
References	To memory words
Maximum block size	8 KB
Maximum main memory size	32 GB
Maximum cache size (excluded labels, block state bits, counts, etc.)	16 MB

Table 1: Architectural characteristics supported by SMPCache.

The configuration files have the format presented in Figure 1. As we can see, the "comments" (indicating what is configured) appear in the even lines, and the choices in the odd ones.

```
Processors in SMP:
3
Cache coherence protocol:
2
Scheme for bus arbitration:
1
Word wide (bits):
64
Words by block:
128
Blocks in main memory:
1024
Blocks in cache:
64
Mapping:
3
Number of cache sets:
0
Replacement policy:
2
Cache levels:
1
Writing strategy:
2
```

Figure 1: Configuration file.

In the case of numerical configurations as the number of processors, the word wide, the words by block,... the numerical configuration itself is written in the file. For the rest of configurations, a numerical code is assigned to each one of the possible options. Table 2 shows the configuration options and their associated numerical codes.

	Possible values	Code
Cache coherence protocol	MSI MESI DRAGON	1 2 3
Scheme for bus arbitration	Random LRU LFU	1 2 3
Mapping	Direct Set-Associative Fully-Associative	1 2 3
Cache sets	NO 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 or 2048	0 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 or 2048
Replacement policy	NO Random LRU FIFO LFU	0 1 2 3 4
Writing strategy	Write-Through Write-Back	1 2

Table 2: Non numerical configuration options and their associated numerical codes.

Although the cache levels and the writing strategy are not configurable, they must also appear in the configuration files, for facilitating possible future extensions.

4. Trace Files

For working with the simulator, it is necessary to use data files with the “calls” and memory addresses demanded by the processors during the execution of a program: the named memory traces. The trace files will allow you to emulate the programs to process for the different processors in your SMP. These ASCII data files (trace files, which have the extension “.prg”) consist of lines, each one has two numbers, separated by only one white space:

Label Value

Where:

- **Label** is a decimal number that identifies the memory access operation type demanded by the processor (CPU), in a given time, according to the instruction program: to capture an instruction (0), to read a memory data (2) or to write a data in memory (3).
- **Value** is an hexadecimal number that indicates the effective address of the memory word to be accessed by the processor (CPU). This address will be

translated by the simulator for locating the word in the memory system block structure.

Figure 2 illustrates an example of trace file to be loaded in some processor of your SMP.

As an example, the part of file of the figure shows a memory trace with 6 instruction captures of a certain program. Three instructions imply data reading, and one asks for writing in memory. In total, those 6 instructions imply 10 memory accesses.

```
0 00001b08
0 00001ca5
2 00007951
0 00001d04
0 00001eb8
2 00007952
0 0000201c
2 00007c71
0 0000201f
3 00007b51
```

Figure 2: Example of trace file.

5. Interface Overview

Once installation is complete (see section 2), you can load SMPCache by clicking on its icon. The first screen you will see is shown in Figure 3, which contains a menu bar, a tool button bar, a configuration panel, and a status bar. We will explain all these components in the following sections.

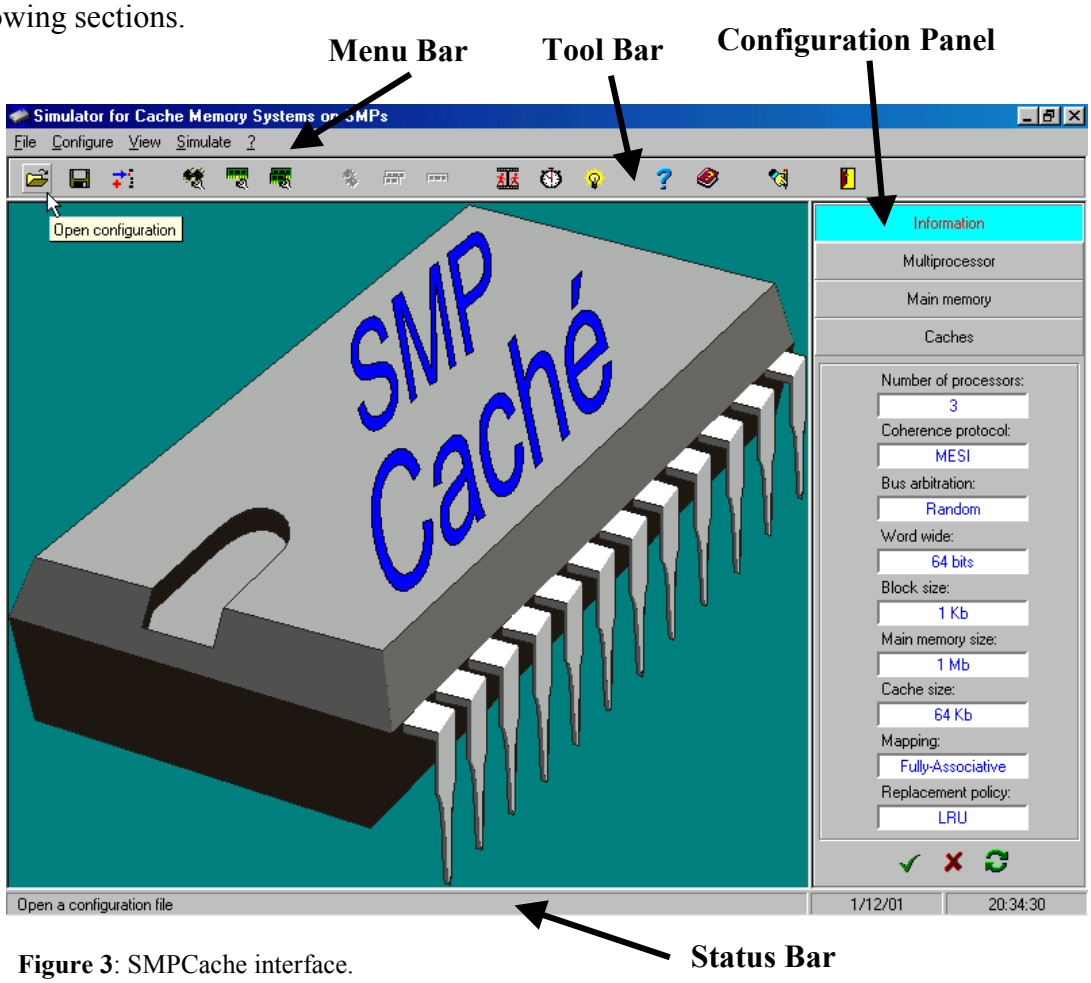


Figure 3: SMPCache interface.

6. Menu Bar

The menu bar is located just above the tool button bar. It contains drop-down menus that list various commands. Commands followed by ellipses (...) indicate that a dialog box opens upon selecting the command. As we can see in Figure 4, the menu bar contains the following menus: File, Configure, View, Simulate, and Help (?).

Figure 4: Menu bar.

6.1. File Menu

The File menu contains the commands shown in Figure 5: Open configuration, Save configuration, Open memory traces, and Exit.

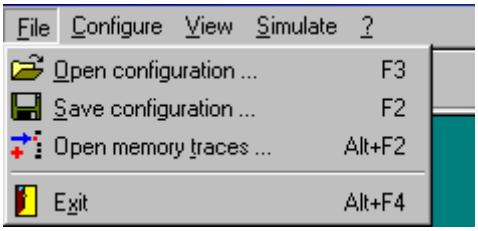


Figure 5: File menu.

6.1.1. Open configuration

This command allows you to open a configuration file (*.cfg), loading the different choices for configuring a given architecture (see Table 1), so the need to make many selections for configuring this architecture is avoided. Figure 6 displays the dialog box opened upon selecting this command.

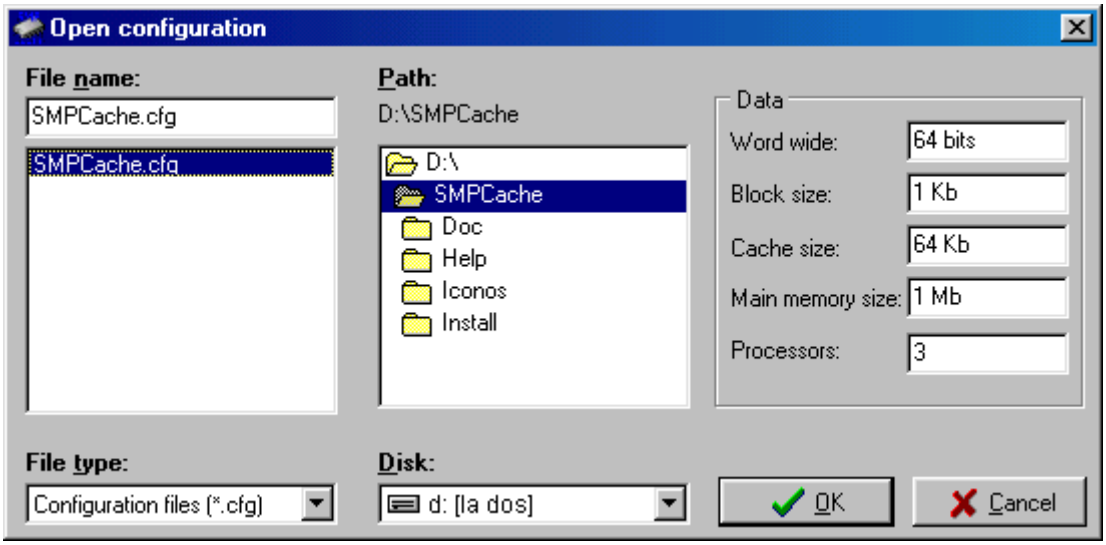


Figure 6: Open configuration.

To list the files of the type you need, make sure that the required Disk and Path are current, and then select the appropriate File type (*.cfg). Then, set the File name to select a configuration file, and click on the OK button.

If you do not remember the name of the configuration file exactly, the Data (word wide, block size, cache size, etc.) shown on the right part of the dialog box can help you.

☞**Remember**: You can also press F3 to select this command, and open a configuration file.

6.1.2. Save configuration

This command allows you to save in a file (*.cfg) the current configuration, for future loading. Figure 7 presents the dialog box opened upon selecting this command.

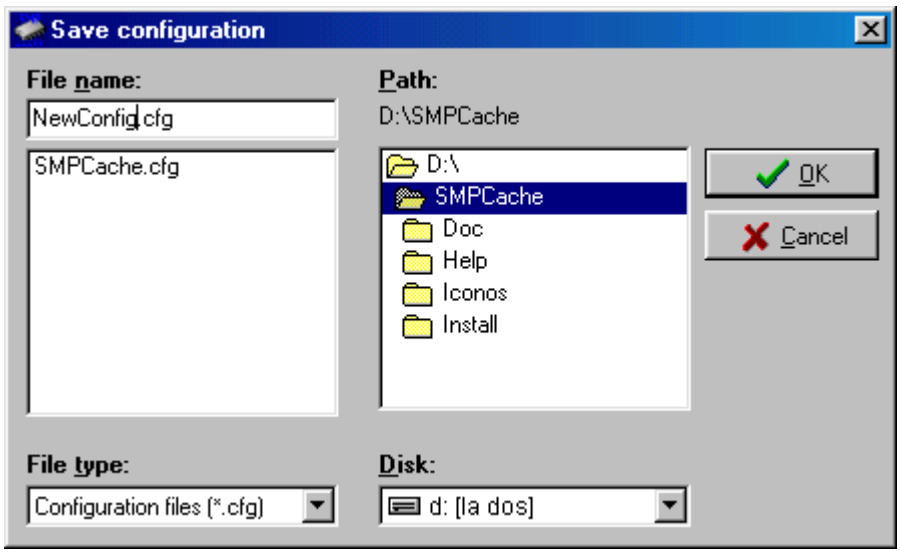


Figure 7: Save configuration.

Indicate the Disk and Path in which to save the current configuration, and then select the appropriate File type (*.cfg). You can double click a folder in the list box to see what is inside of it. Finally, type the File name in which to save the current configuration, and click on the OK button.

☞**Remember**: You can also press F2 to select this command, and save in a file the current configuration.

6.1.3. Open memory traces

This command allows you to open a trace file (*.prg), loading it in some of the active processors in your SMP. Figure 8 shows the dialog box opened upon selecting this command.

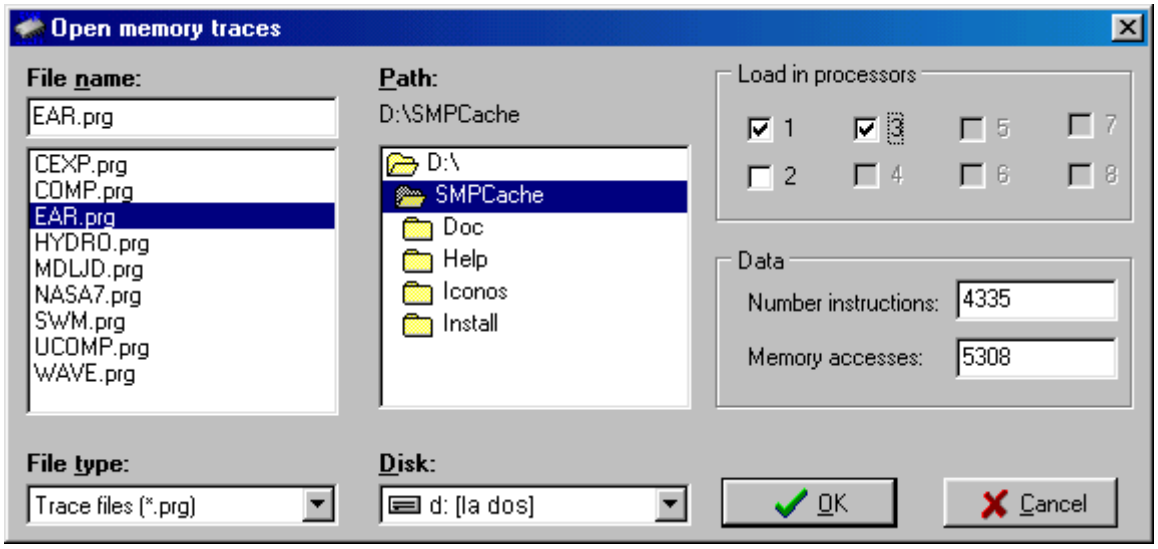


Figure 8: Open memory traces.

To list the files of the type you need, make sure that the required Disk and Path are current, and then select the appropriate File type (*.prg). Then, set the File name to select a trace file, and indicate in what active processors it will be loaded (Load in processors). Finally, click on the OK button.

If you do not remember the exact name of the trace file, the Data (number of instructions, and memory accesses) shown on the right part of the dialog box can help you.

☞**Remember**: You can also press Alt+F2 to select this command, and load memory traces in the active processors.

6.1.4. Exit

This command allows you to exit SMPCache. You will be prompted to confirm that you really want to quit the simulator.

☞**Remember**: You can also press Alt+F4 to select this command, and quit SMPCache.

6.2. Configure Menu

The Configure menu contains the commands shown in Figure 9: Multiprocessor, Main memory, Caches, and Save as initial configuration.

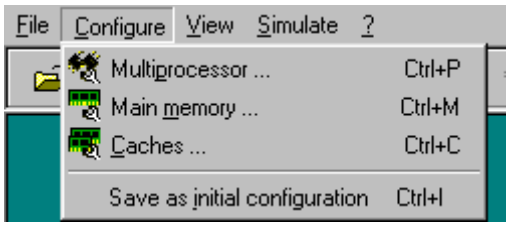


Figure 9: Configure menu.

6.2.1. Multiprocessor

This command allows you to configure general aspects of your multiprocessor. Figure 10 displays the dialog box opened upon selecting this command.

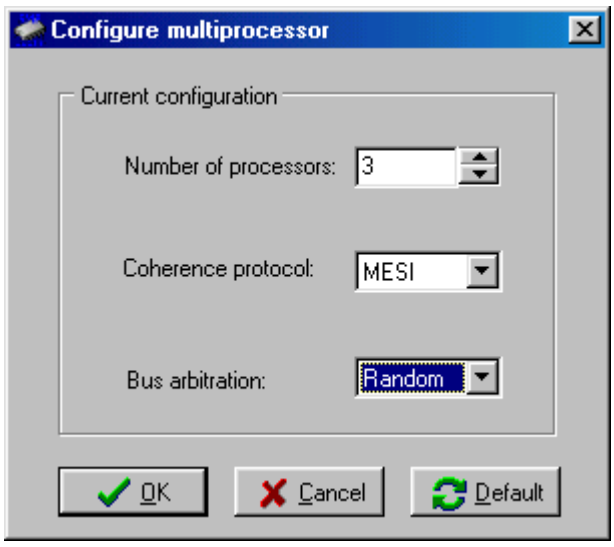


Figure 10: Configure multiprocessor.

In order to update the current configuration you must select the appropriate configuration for the shown parameters (Number of processors, Coherence protocol, and Bus arbitration), and then click on the OK button. The Default button sets the default values in these configuration parameters. Table 3 presents the configuration parameters, their possible values and their default values.

	Possible values	Default value
Number of processors	1, 2, 3, 4, 5, 6, 7 or 8	3
Coherence protocol	MSI, MESI or DRAGON	MESI
Bus arbitration	Random, LRU or LFU	Random

Table 3: Configuration parameters related with general aspects of the multiprocessor.

☞**Remember**: You can also press Ctrl+P to select this command, and configure the general aspects of the multiprocessor.

6.2.2. Main memory

This command allows you to configure the main memory of the SMP. Figure 11 shows the dialog box opened upon selecting this command.

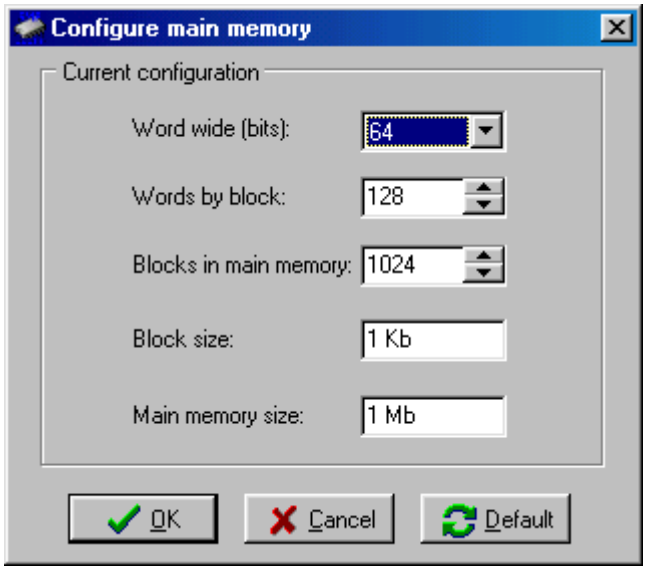


Figure 11: Configure main memory.

In order to update the current configuration you must select the appropriate configuration for the shown parameters (Word wide, Words by block, and Blocks in main memory), and then click on the OK button. The Default button sets the default values in these configuration parameters. The "Block size" and "Main memory size" parameters are computed automatically. Table 4 presents the configuration parameters, their possible values and their default values.

	Possible values	Default value
Word wide (bits)	8, 16, 32 or 64	64
Words by block	1, 2, 4, 8, 16, 32, 64, 128, 256, 512 or 1024	128
Blocks in main memory	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152 or 4194304	1024

Table 4: Configuration parameters related with the main memory of the multiprocessor.

In conclusion, the maximum block size is 8 KB, and the maximum main memory size is 32 GB. By default, the block size is 1 KB, and the main memory size is 1 MB.

☞**Remember**: You can also press Ctrl+M to select this command, and configure the main memory of the multiprocessor.

6.2.3. Caches

This command allows you to configure the caches associated to each processor in the SMP. All the caches will have the same configuration. Figure 12 presents the dialog box opened upon selecting this command.

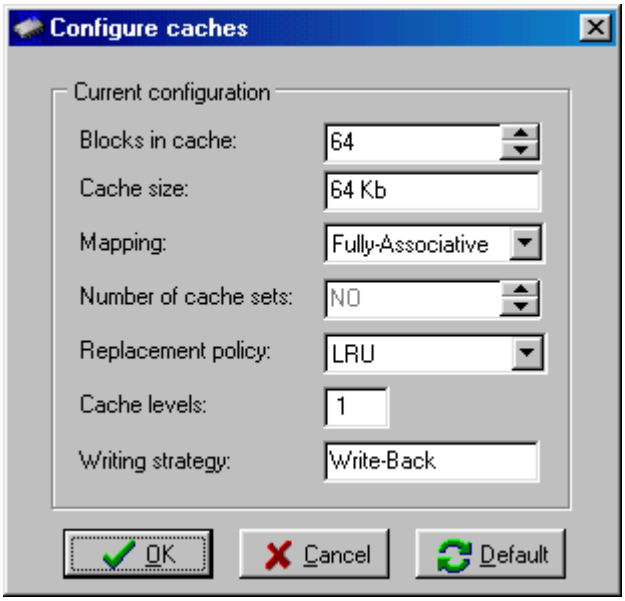


Figure 12: Configure caches.

In order to update the current configuration you must select the appropriate configuration for the shown parameters (Blocks in cache, Mapping, Number of cache sets, and Replacement policy), and then click on the OK button. The Default button sets the default values in these configuration parameters. The "Cache size" parameter is computed automatically, using the Blocks in cache and the Block size (section 6.2.2). The "Cache levels" (1) and "Writing strategy" (Write-Back) parameters are not configurable, but they also appear for facilitating possible future extensions.

All the configuration parameters are related among themselves according to the theoretical models. If you make a choice that contradicts other parameters, the simulator warns you, and blocks the choice. Table 5 shows the configuration parameters, their possible values and their default values.

	Possible values	Default value
Blocks in cache	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 or 2048	64
Mapping	Direct, Set-Associative or Fully-Associative	Fully-Associative
Number of cache sets	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 or 2048	NO
Replacement policy	Random, LRU, FIFO or LFU	LRU

Table 5: Configuration parameters related with the processor caches in the multiprocessor.

In conclusion, the maximum cache size is 16 MB (because the maximum block size is 8 KB, see section 6.2.2). By default, the cache size is 64 KB (excluded labels, block state bits, counts, etc.).

☞**Remember**: You can also press Ctrl+C to select this command, and configure the caches associated to each processor in the multiprocessor.

6.2.4. Save as initial configuration

This command allows you to save the current configuration of the simulator like preset configuration (SMPCache.ini file). That is, with this command you can set a default initial configuration for the simulator.

☞**Remember**: You can also press Ctrl+I to select this command, and save the current configuration of the system as preset configuration.

6.3. Simulate Menu

The Simulate menu contains the commands shown in Figure 13: Step by step, With breakpoint, and Complete execution. This menu includes the commands related with the way of executing the simulations. The simulations can be carried out as a whole (Complete execution) or, as it is usually much more interesting, Step by step, in order to observe the internal operation of the system. For very long traces, breakpoints (With breakpoint) can also be inserted. There are therefore three kinds of simulation (one of them is active), and it is possible to change from one to another without waiting for the end of the simulation.

Figure 13: Simulate menu.

6.3.1. Step by step

If you select this command the simulations are stopped after every memory access. This is the kind of simulation that is set initially by default.

☞**Remember**: You can also press F7 to select this command, and perform a step-by-step simulation.

6.3.2. With breakpoint

If you select this command the simulations are stopped when a specific number of memory accesses is reached.

☞**Remember**: You can also press F5 to select this command, and perform a simulation with breakpoints.

6.3.3. Complete execution

If you select this command the simulations are stopped when all the memory traces are finished.

☞**Remember**: You can also press F9 to select this command, and perform a complete simulation.

6.4. View Menu

The View menu contains the commands shown in Figure 14: Multiprocessor evolution, Cache evolution, and Memory block evolution. In order to be able to choose any command of the View menu, it is necessary to have loaded as minimum one trace file (see section 6.1.3) in some processor.

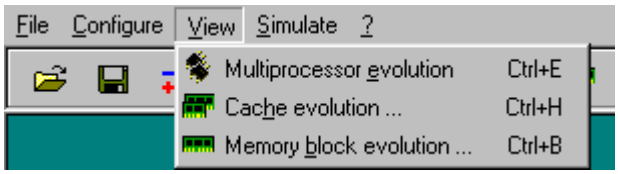


Figure 14: View menu.

6.4.1. Multiprocessor evolution

This command allows you to observe a global vision of the multiprocessor evolution according to the set configuration. Figure 15 presents the window opened upon selecting this command. In this window, the simulator shows you how the system responds to the memory accesses that the programs generate (trace files used for the different processors during the simulation).

Only the processors with memory trace influence the simulation. In Figure 15 we are supposing three processors in the SMP, the processors 1 and 3 have loaded a trace.

In this window, a diagram of the SMP is displayed. In this diagram the bus transactions (BusRd, BusRdX, BusWB and BusUpd), the processor requests (PrRd XXX and PrWr XXX) and the transfers of entire blocks (Block XXX) will be represented. An arrow will indicate the direction of the bus transaction, processor request or block transfer.

The bus transactions will always go from a particular cache to the bus, and although it is not represented, it is supposed that every cache observes every generated transaction (remember that we use snoopy cache coherence protocols). In Figure 15, with MSI cache coherence protocol, the cache 1 generates a BusRdX transaction.

The processor requests will also go in only one direction, from the processor to its cache. We assume the processor issues two kinds of requests, reads (PrRd) and writes (PrWr). Both the instruction captures (label 0, see section 4) and the data readings (label 2) are

considered as PrRd requests. The read or write could be to a memory block that exists in the cache or to one that does not. We represent the cache hits with a red asterisk (*).

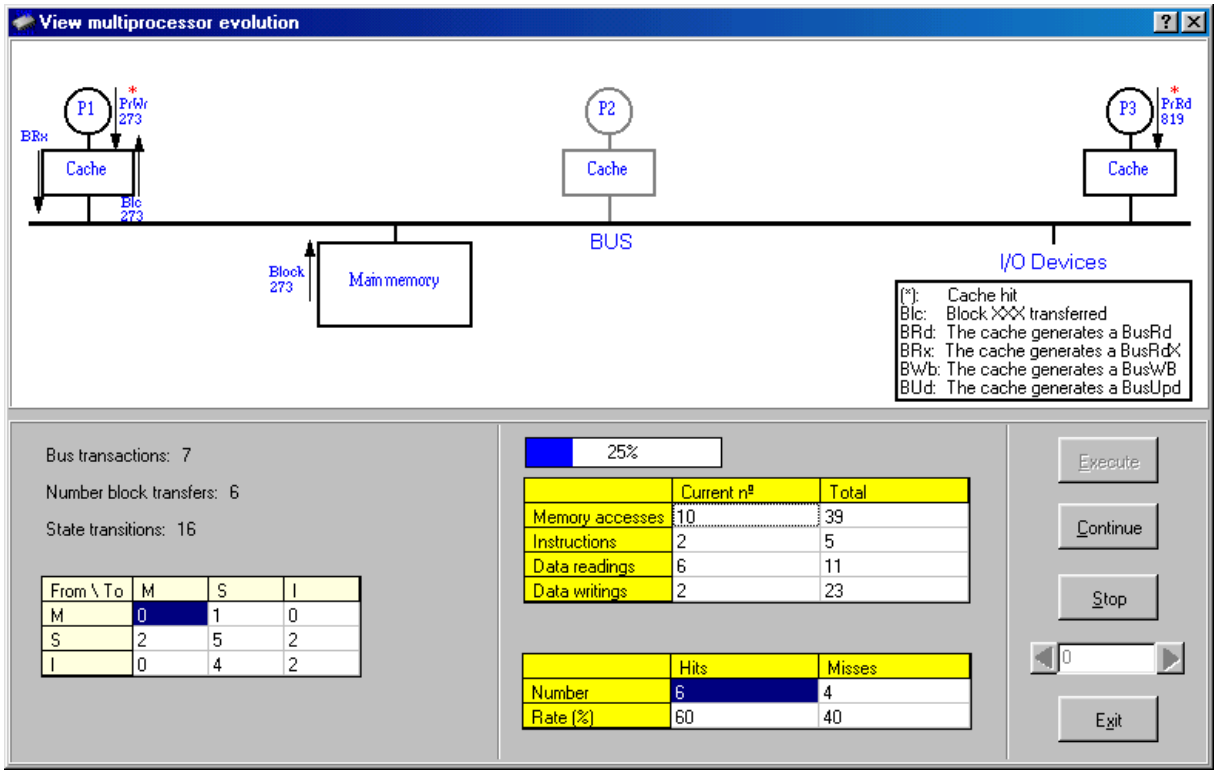


Figure 15: View multiprocessor evolution.

The block transfers will be labelled with the "Block XXX" text. Where XXX will be the block address in main memory. In Figure 15, after the BusRdX transaction, the main memory supplies the memory block 273.

If you use MESI or DRAGON coherence protocol the value for the shared signal is also shown. This signal will be "YES" whether there are any other processors caching the memory block referenced in a bus transaction.

The simulator also shows, in this window, interesting statistical data like:

- Number of bus transactions.
- Number of block transfers on the bus.
- Number of state transitions (it depends on cache coherence protocol).
- Number of state transitions from a particular state to other.
- Global number of memory accesses, and for types: instruction captures, data readings and data writings.
- Number of cache hits and misses, as well as the hit and miss rate.

In order to start the simulation click on the Execute button. You can abort the simulation at any time (for example, in order to correct any architectural detail) clicking on

the Exit button. The numerical edit box allows you to put a breakpoint (the memory access in which the simulation must stop automatically). This edit box can be only used in a simulation with breakpoint. Finally, the Stop button allows you to stop (pause) immediately a simulation with breakpoint or a complete simulation. In order to continue that simulation or a step-by-step simulation click on the Continue button.

☞**Remember**: You can also press Ctrl+E to select this command, and observe a global vision of the multiprocessor evolution.

6.4.2. Cache evolution

This command allows you to observe the evolution of a particular cache according to the set configuration and the memory accesses that the programs generate (trace files). Although only the data of this cache are shown, the relationship among all the multiprocessor elements is taken into account. Figure 16 presents the dialog box opened upon selecting this command.

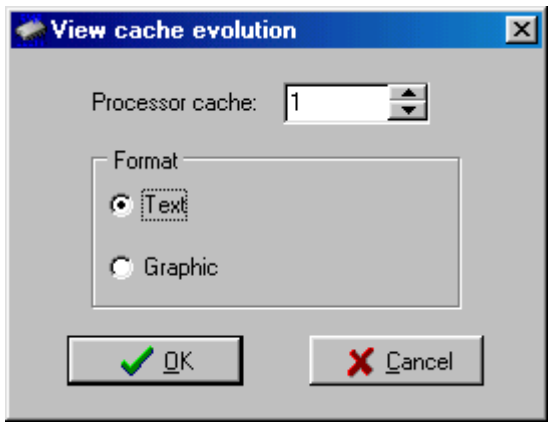


Figure 16: View cache evolution.

You can select the cache to study (it must be a cache whose processor has an associated trace file) and the format in which the data will be shown (text or graphic format), then click on the OK button.

If you choose text format, the window presented in Figure 17 will be opened. In this window, the state of the cache under study is always displayed. This state includes the number of cache (processor), the number of blocks in cache, the state of every memory block within this cache, etc. The main data of the current access to this cache are also presented: access number within the associated trace file, access type, effective address of the memory word to be accessed by the processor, ...

The simulator also displays, in this window, interesting statistical data regarding this specific cache like:

- Number of bus transactions generated by this cache.
- Number of block transfers on the bus.
- Number of state transitions in this cache (it depends on cache coherence protocol).

- Number of state transitions from a particular state to other.
- Global number of memory accesses in this cache, and for types: instruction captures, data readings and data writings.
- Number of cache hits and misses, as well as the hit and miss rate.

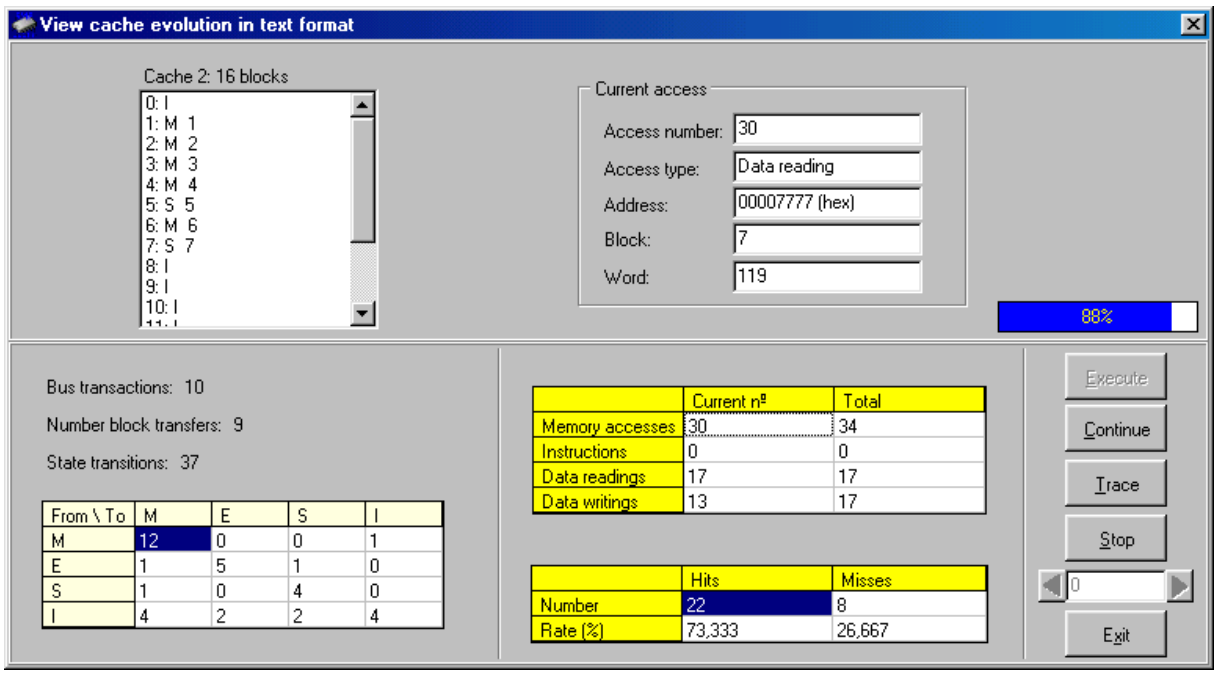


Figure 17: View cache evolution in text format.

In order to start the simulation click on the Execute button. You can abort the simulation at any time (for example, in order to correct any architectural detail) clicking on the Exit button. The numerical edit box allows you to put a breakpoint (the memory access in which the simulation must stop automatically). This edit box can be only used in a simulation with breakpoint. Finally, the Stop button allows you to stop (pause) immediately a simulation with breakpoint or a complete simulation. In order to continue that simulation or a step-by-step simulation click on the Continue button. Figure 18 presents the dialog box opened upon clicking on the Trace button.

In this dialog box, you can obtain detailed information about the memory trace associated with that cache (processor): name of the trace file, memory accesses within the trace, etc. In order to get the main data (access number within the trace file, access type, ...) of any memory access within the trace click on the desired access.

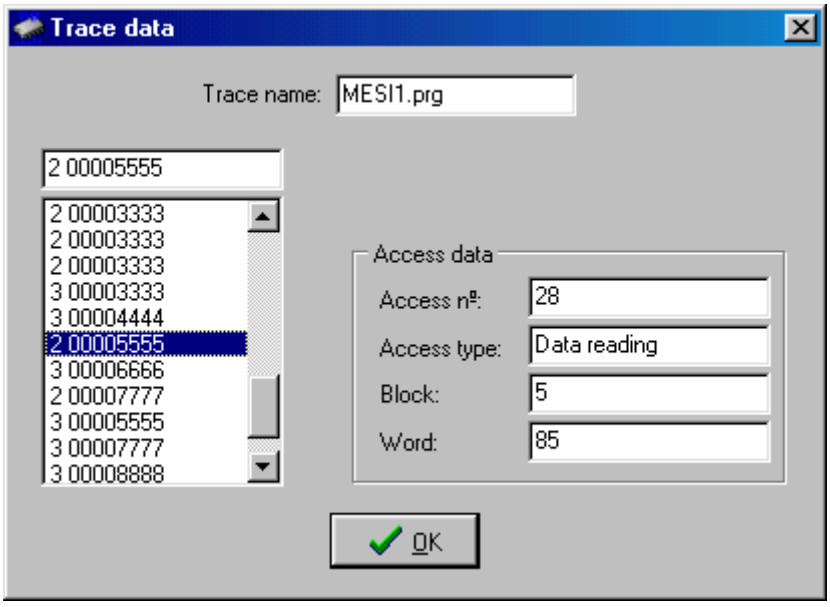


Figure 18: Trace data.

If you choose graphic format in the dialog box of Figure 16, the window presented in Figure 19 will be opened.

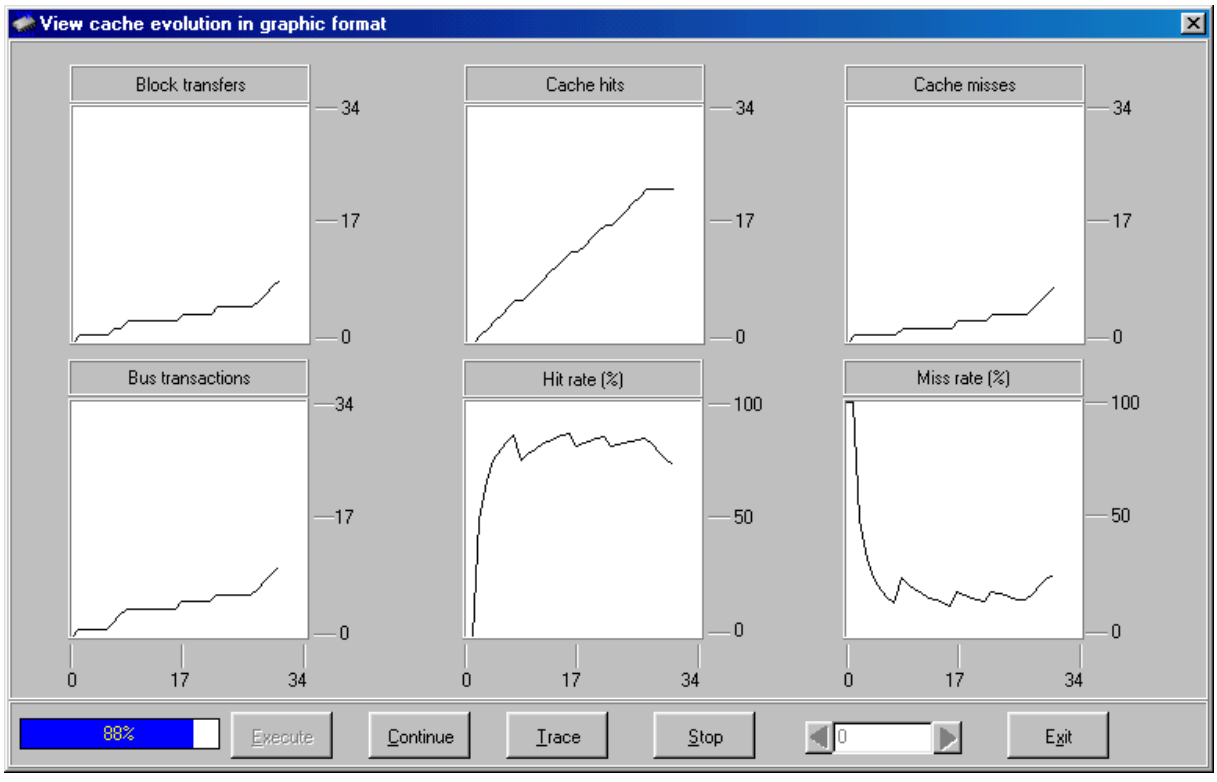


Figure 19: View cache evolution in graphic format.

In this window, the simulator shows, using several graphics, interesting measurements regarding that specific cache like:

- Number of block transfers on the bus.
- Number of bus transactions generated by that cache.
- Number of cache hits and misses, as well as the hit and miss rate.

All these graphs indicate in their X axis the memory access for which that value in their Y axis is obtained.

The buttons in this window have the same use as for the case of text format (Figure 17).

☞**Remember**: You can also press Ctrl+H to select this command, and observe the evolution of a particular cache in text or graphic format.

6.4.3. Memory block evolution

This command allows you to observe the evolution of a specific memory block inside a particular cache or the complete multiprocessor. Although only the data of this block are shown, the relationship among all the multiprocessor elements is taken into account. Figure 20 shows the dialog box opened upon selecting this command.

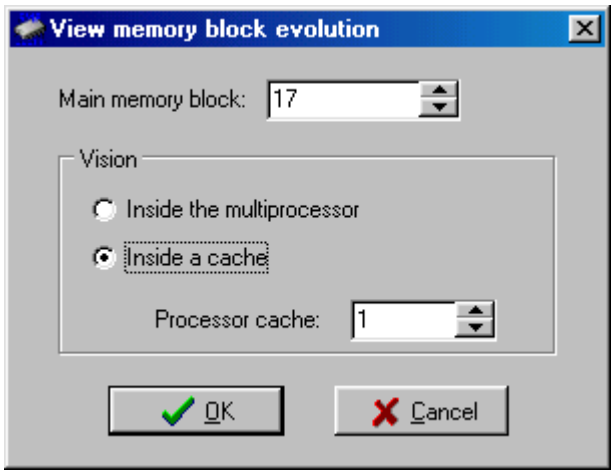


Figure 20: View memory block evolution.

You can select the main memory block to study and the desired vision level: inside the multiprocessor or inside a specific cache (it must be a cache whose processor has an associated trace file). Then click on the OK button.

If you choose the multiprocessor vision, the window presented in Figure 21 will be opened. This window is very similar to the one shown when the “Multiprocessor evolution” command was explained (section 6.4.1). The great difference is that, in this case, the shown data are not the totals of the complete system, but the totals of the memory block under evaluation (in Figure 21, the main memory block 17). Only the data related with that block are shown. In conclusion, you can see the state of that block inside every cache in the multiprocessor, the memory accesses to that block, and for types (instruction captures, data readings and data writings), and so on. For a more detailed explanation see section 6.4.1.

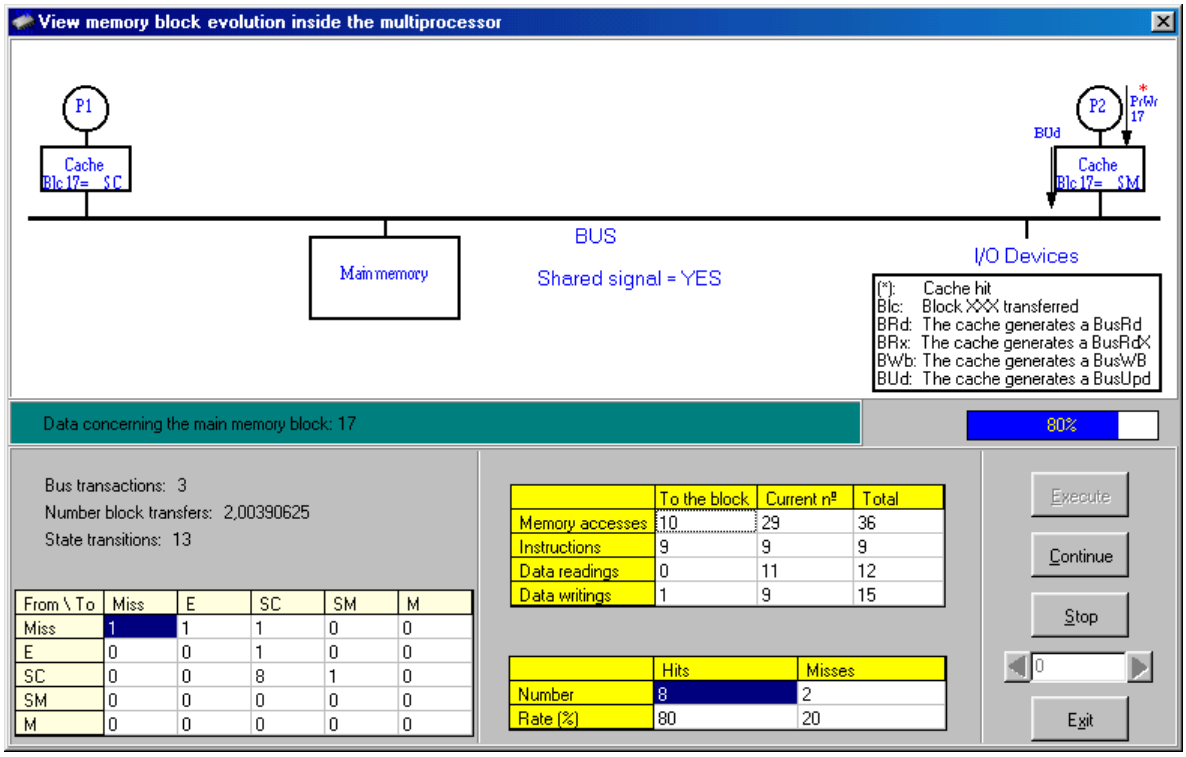


Figure 21: View memory block evolution inside the multiprocessor.

If you choose the cache vision in the dialog box of Figure 20, the window presented in Figure 22 will be opened.

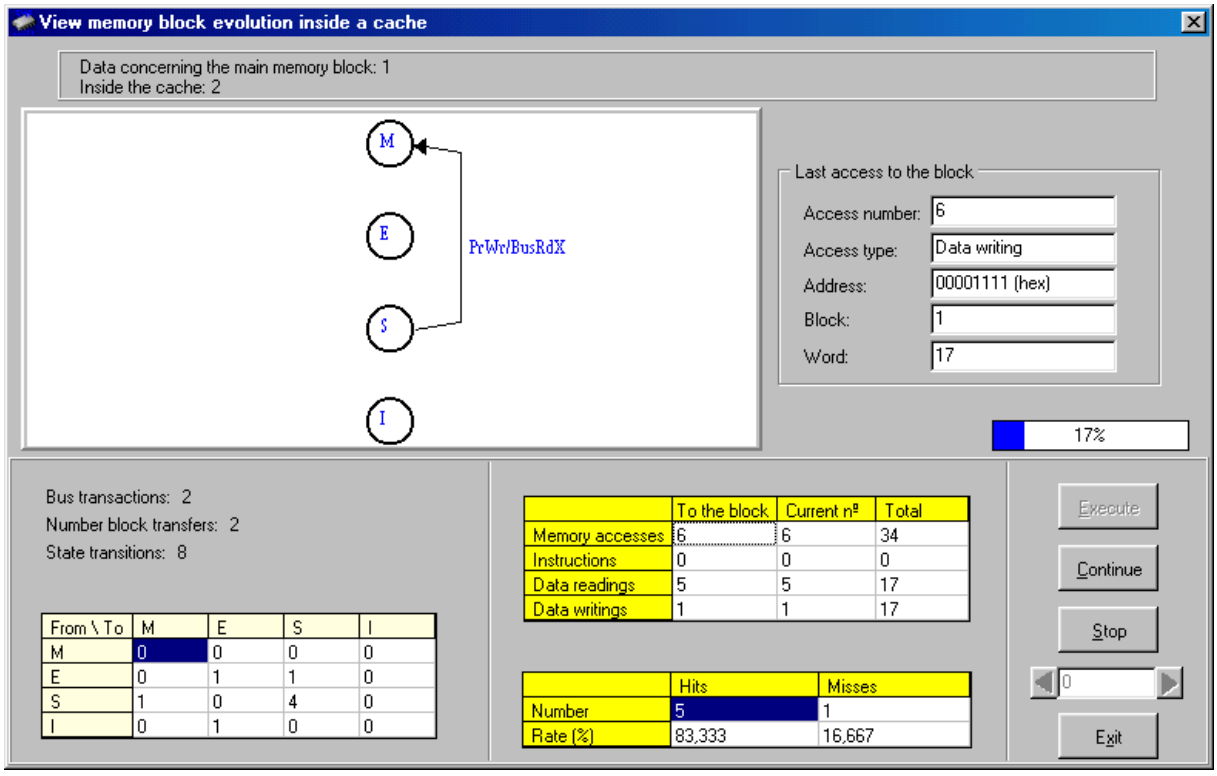


Figure 22: View memory block evolution inside a cache.

In this window, the evolution of the memory block under study inside a particular cache is always displayed. This evolution is represented using the state transition diagram for that block (it depends on cache coherence protocol). The state-transition diagram takes as inputs the current block state and the processor request or observed bus transaction, and produces as output the next state for the cache block. In this diagram, the notation A/B means if you observe from processor-side or bus-side event A, then in addition to state change, generate bus transaction or action B. For example, in Figure 22 for the MESI cache coherence protocol, on a processor write (PrWr) a BusRdX transaction is generated, and upon completion of this transaction the block transitions up to the "M" state.

This window also indicates the main memory block and cache (processor) under study, and the main data of the last access to that block inside that processor (cache): access number within the associated trace file, access type, effective address of the memory word to be accessed by the processor, ...

The rest of data shown in this window have the same explanation as for the multiprocessor vision case (Figure 21). The main difference is that, in this case, the shown data are regarding that memory block and that specific cache.

In order to start the simulation click on the Execute button. You can abort the simulation at any time (for example, in order to correct any architectural detail) clicking on the Exit button. The numerical edit box allows you to put a breakpoint (the memory access in which the simulation must stop automatically). This edit box can be only used in a simulation with breakpoint. Finally, the Stop button allows you to stop (pause) immediately a simulation with breakpoint or a complete simulation. In order to continue that simulation or a step-by-step simulation click on the Continue button.

☞**Remember**: You can also press Ctrl+B to select this command, and observe the evolution of a specific memory block.

6.5. Help (?) Menu

The Help (?) menu contains the commands shown in Figure 23: Using SMPCache, Theoretical concepts, and About.

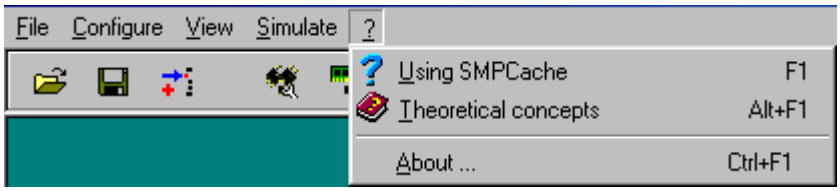


Figure 23: Help (?) menu.

6.5.1. Using SMPCache

This command allows you to obtain help about the use of SMPCache: menus, commands, tool bar, etc.

☞**Remember**: You can also press F1 to select this command, and obtain help about the use of the simulator (menus, commands, ...).

6.5.2. Theoretical concepts

This command allows you to obtain help with the main theoretical concepts about caches in symmetric multiprocessors.

☞**Remember**: You can also press Alt+F1 to select this command, and obtain help with the main theoretical concepts about caches in SMPs.

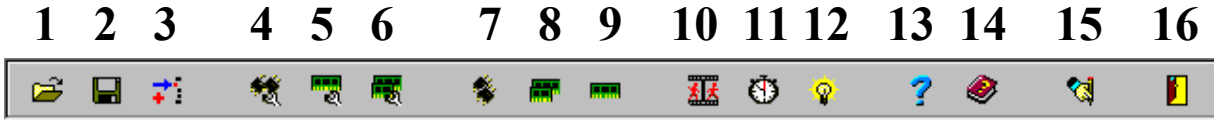
6.5.3. About

This command allows you to obtain general information about SMPCache (authors, version number, etc.).

☞**Remember**: You can also press Ctrl+F1 to select this command, and obtain general information about the simulator.

7. Tool Bar

The tool button bar is located underneath the menu bar (described in the previous sections). The tool bar allows you to quickly access the most common commands in SMPCache (Figure 24).



1. **Open configuration**: see section 6.1.1.
2. **Save configuration**: see section 6.1.2.
3. **Open memory traces**: see section 6.1.3.
4. **Configure multiprocessor**: see section 6.2.1.
5. **Configure main memory**: see section 6.2.2.
6. **Configure caches**: see section 6.2.3.
7. **View multiprocessor evolution**: see section 6.4.1.
8. **View cache evolution**: see section 6.4.2.
9. **View memory block evolution**: see section 6.4.3.
10. **Simulate step by step**: see section 6.3.1.
11. **Simulate with breakpoint**: see section 6.3.2.

12. Simulate complete execution: see section 6.3.3.

13. Using SMPCache: see section 6.5.1.

14. Theoretical concepts: see section 6.5.2.

15. Configure: Show / Hide the configuration panel.

16. Exit: see section 6.1.4.

Figure 24: Tool button bar.

8. Configuration Panel

The configuration panel is located on the right of the application window (Figure 25). This panel allows you to select the different choices for configuring a given architecture. Therefore, it has a similar use to the Configure menu (section 6.2).

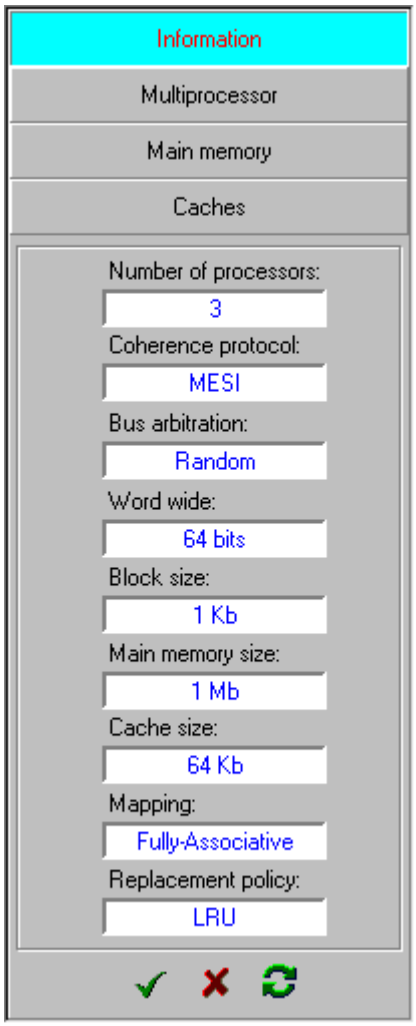


Figure 25: Configuration panel.

The Configuration panel has four tabs: Information, Multiprocessor, Main memory, and Caches. The Information tab shows general information about the current configuration. The rest of tabs are very similar to the commands in the Configure menu.

☞**Remember**: In the Configuration panel, the OK, Cancel and Default buttons are represented by icons at the bottom of the panel.

9. Status Bar

The status bar is located at the bottom of the application window. It displays different information messages. For example, when you place the cursor over a tool button or a menu command, the status bar displays a short description of the functionality of that tool button or menu command.

Furthermore, the status bar always displays the date and time of the system.

10. SMPCache Use Overview

Once you have loaded SMPCache by clicking on its icon, the steps to follow in order to use the simulator are:

1. Configure the simulator:
   a) Using the default initial configuration of the simulator (section 6.2.4).
   b) Modifying the initial configuration by means of the Configure menu (section 6.2) or the Configuration panel (section 8).
   c) Loading the different choices (section 6.1.1) from a previously saved (section 6.1.2) configuration file (section 3).
2. Load the memory traces (trace files, section 4) in the active processors in your SMP, using the "Open memory traces" command (section 6.1.3). Remember that you must use this command for each different trace.
3. Select the way of executing the simulation, using the Simulate menu (section 6.3). Remember that there are three kinds of simulation (Step by step, With breakpoint, and Complete execution), and it is possible to change from one to another without waiting for the end of the simulation. By default, the step-by-step simulation is active.
4. Select the vision level and start the simulation, using the View menu (section 6.4). You can carry out a simulation observing the complete multiprocessor and all the memory blocks ("Multiprocessor evolution" command, section 6.4.1) or only one particular block ("Memory block evolution" command, section 6.4.3, multiprocessor vision). You can also observe a specific cache, and all the memory blocks ("Cache evolution" command, section 6.4.2, in text or graphic format) or only a concrete block ("Memory block evolution" command, section 6.4.3, cache vision).