Experto A

Basic Operation

Start the simulator. After a few seconds, you should see a small window appear similar to the one below.

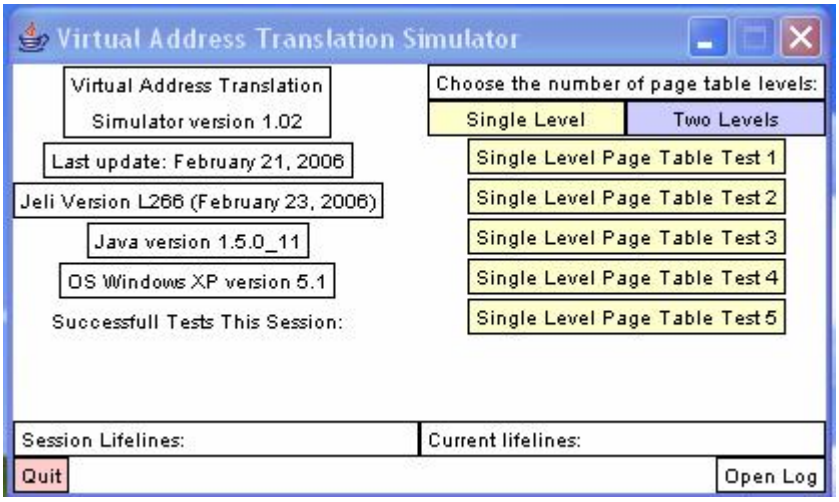


Figure 1: The main simulator window.

Figure 2 shows the result of pushing one of the yellow single level page table test buttons. At the top of this window are green **Lifeline** and **Help** buttons that can give guidance in using the simulator. Most windows used by the simulator have a green **Help** button that pops up a message about using the window. The simulator window contains the data for doing an address translation problem. Seven values are given:

1. The number of page table levels, in this case it is 1. The simulator supports only 1 and 2-level page tables.
2. The page size in bytes. For a 2-level page table, this includes both the top level page second level pages (explained in detail later). For single level page tables this is the number of bytes in a page of virtual memory or the number of bytes in a frame of physical memory.
3. Level 2 page table size. This value is 0 for single level page tables.
4. The number of bits in a logical address.
5. The number of bits in a physical address.
6. The number of entries in the translation lookaside buffer (TLB). If no TLB is used, this value will be 0.
7. The number of bytes in a page table entry. For 2-level page tables it is assumed that all page tables have the same width.

Below these values is the physical address of the start of the top level page table. This value is included for completeness, but is usually not needed in running the simulator. Next appear two widgets containing the logical and physical addresses. The value of the logical address is given (in binary) and the value of the physical address is to be determined by the user. Basically, the fields of the logical address are manipulated to produce the fields of the physical address. The widgets are designed for easy manipulation of these bit fields as explained below. Below the logical and physical

address widgets are two yellow buttons used to indicate when the user is done generating the physical address. Push the **Found Physical Address** button when you think you have correctly filled in the physical address or the **Page Fault** is you determined that a page fault has occurred. The **Status** field below this indicates whether the test was done successfully or not. It indicates **In Progress** until one of the above buttons is pushed, and then indicates success or failure. Lastly are a row of buttons for displaying information useful in calculating the physical address. These include displays of the TLB and memory (mainly in the form of page tables), a progress list to guide the user through the steps needed to do a translation, and a binary bitfield calculator for adding and shifting binary values.

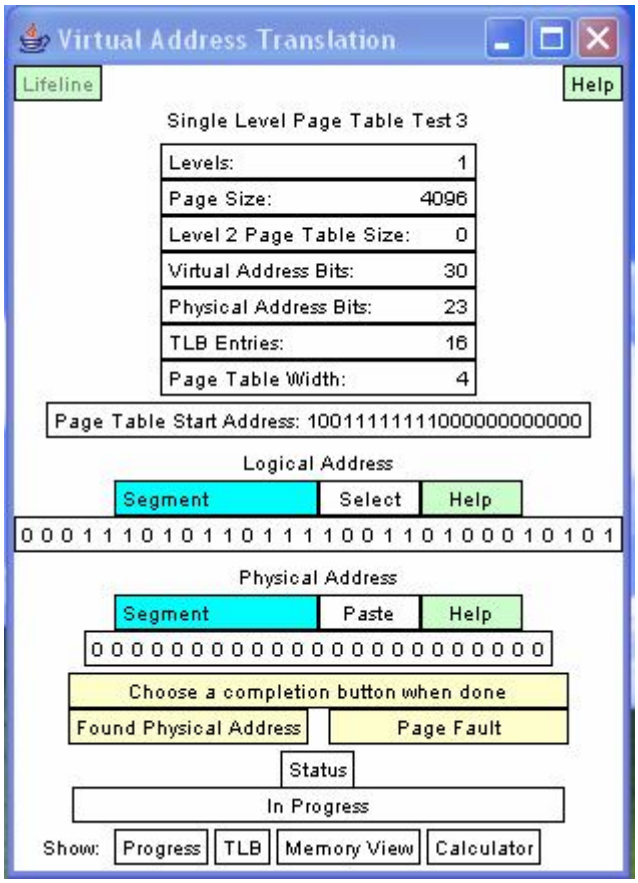


Figure 2: A single level page table test.

The general paradigm for manipulating the binary bit fields is **cut and paste**, or more precisely, **copy and paste** since bit fields are never removed (cut), just copied. The simulator uses a single **clipboard** shared by all entities used by the simulator. Bit fields are moved around by first selecting them by clicking the mouse appropriately. This puts the bit field in the clipboard and then pasting them in the desired place, again by appropriately clicking the mouse. Some fields such as those of the logical address, TLB and memory cannot be changed and so can only be selected. Others, such as those in the physical address only be pasted into.

Experto B

The Logical and Physical Addresses

The Logical and Physical Address widgets manipulate bit strings. The strings can be divided into fields. Each field can be selected (put in the clipboard) or a value can be pasted (replaced by the value in the clipboard). Each of these is done by clicking with the mouse. Which action is performed depends on the mode that is set. Each of the widgets has two rows of information. The top row is a set of buttons controlling the mode of the widget. The possible modes are Segment, Select, and Paste. The current mode is highlighted with a cyan background. An additional **Help** button describes how to use the widget. Since the contents of the logical address cannot be changed by the user, this widget does not have a **Paste** mode. The widget starts in the **Segment** mode that allows it to be divided into fields. For single level page tables, only two fields are allowed, the high order bits representing a page number and the low order bits representing the page offset. Click the mouse between two of the binary digits to segment the logical address. As you move the mouse in the logical address, the **Segment** button shows the sizes of the two fields that would be obtained by clicking the mouse at that time. To copy the offset from the logical address to the physical address, first select the offset field by clicking on the select button to change to Select mode and then clicking on the offset field. In a similar way, segment the physical address into its fields, change to **Paste** mode, and click on the offset field in the physical address to paste the offset into the physical address.

The Progress List

The Progress List is displayed when the **Progress** button is pushed. The Progress List shows the list of steps needed to do the address translation and indicates which of the steps have been completed successfully. The figure 3 shows the state of the progress list after the offset has been correctly copied into the physical address.

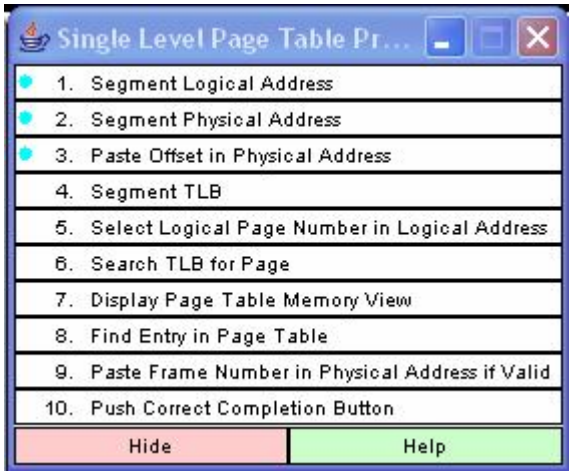


Figure 3: The progress list after the offset has been correctly copied into the physical address.

The TLB

Pushing the **TLB** button at the bottom of Figure 2 displays the TLB as shown in Figure 4a. The TLB must first be segmented into a page number and frame number by clicking in the appropriate column. The result of correctly segmenting the TLB is shown in Figure 4b. The number of bits representing the page number and the number of bits for the frame number are displayed. The TLB also shows the value in the clipboard. Once the TLB is segmented, the simplest way to do a lookup is the put the page number to be looked up in the clipboard and clicking the **Lookup** button in the TLB. If the Logical Address is in **Select** mode, as it would normally be from pasting the offset in the physical address, just click on the page number filed in the Logical Address. This puts in in the clipboard and this value will appear in the TLB clipboard entry. Push the **Lookup** button in the TLB. If the lookup is successful, the corresponding frame number will be highlighted and put in the clipboard. Once this is done, it can be pasted into the physical address simply by clicking on the frame number field of the physical address.

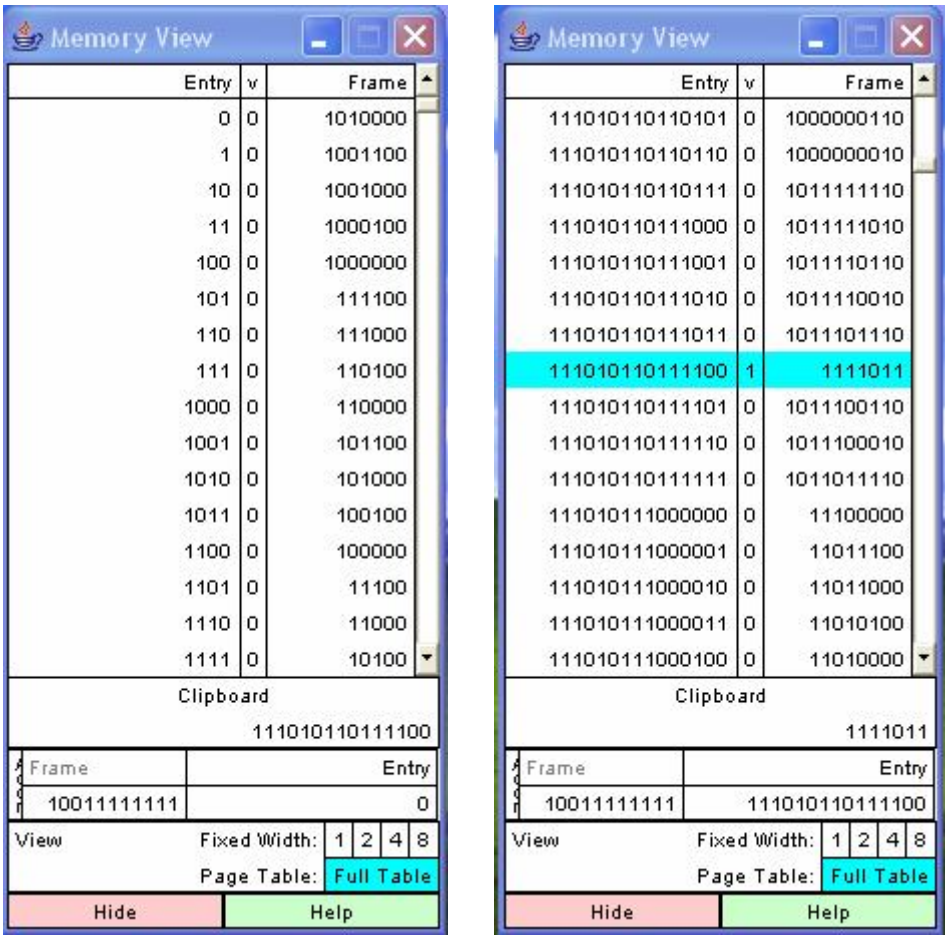
(a) (b)

Figure 4: The TLB, before and after segmenting.

Experto C

The Memory View - Single Level Page Table

This section describes the memory view in the case of a single level page table. Only the page table view of memory is discussed here. If the lookup in the TLB is not successful, a page table lookup is necessary. Display the page table by pushing the **Memory View** button. The default memory view shows the full single level page table, as seen in Figure 5a. The index of the entry appears (in binary) on the left followed by the valid bit and the frame number. Use the scroll bar to scroll through the page table. The starting frame number of the page table is automatically set and shown below the word **Frame** at the bottom of the window. Since the page table may be large, the simplest way to find an entry in the page table is to put its index in the clipboard and push the **Entry** button near the lower right side of the window. The correct value should still be in the clipboard from the TLB lookup. If not, select the page number from the logical address and then click the **Entry** button in the memory view. If the lookup is successful, the corresponding entry is highlighted and the frame number is put in the clipboard. The result of a successful lookup is shown in Figure 5b.



(a) (b)

Figure 5: The Memory View, before and after a successful lookup.

If the valid bit of this entry is clear, it indicates a page fault. Push the yellow **Page Fault** button. Otherwise, the frame number must be pasted into the physical address. Assuming the Physical Address is still in **Paste** mode, just click the left field in the Physical Address. Now click the yellow **Found Physical Address** button. After pushing one of these yellow buttons, the **Status** will indicate success or failure. If you were successful, the test number will appear in the list of successfully completed tests in the menu window shown in Figure 1.

Experto D

Two Level Page Tables

In the two-level page table implementation used here, the top level page table contains a valid bit and the physical frame number of the second level page table. All valid second level page tables must be in physical memory. An access to an address which corresponds to top level page table entry with the valid bit clear corresponds to an invalid memory reference, not a page fault. A page fault can only occur when the second level page table has the corresponding valid bit clear. For the two level tests using a TLB, TLB entries contain a full page number consisting of the top level and second level page numbers and a frame number.

Figure 6 shows the corresponding progress list. The TLB lookup for the two level page table is the same as for the single level page table and is exactly the same if the item is found in the TLB. If not, the top level page table must be searched. Figure 7a shows the top level page table gotten by clicking the **Memory View** button.

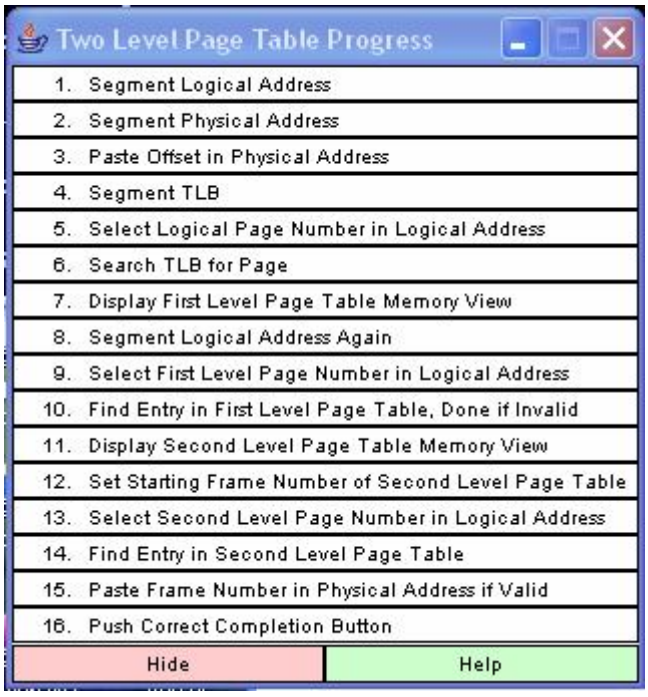
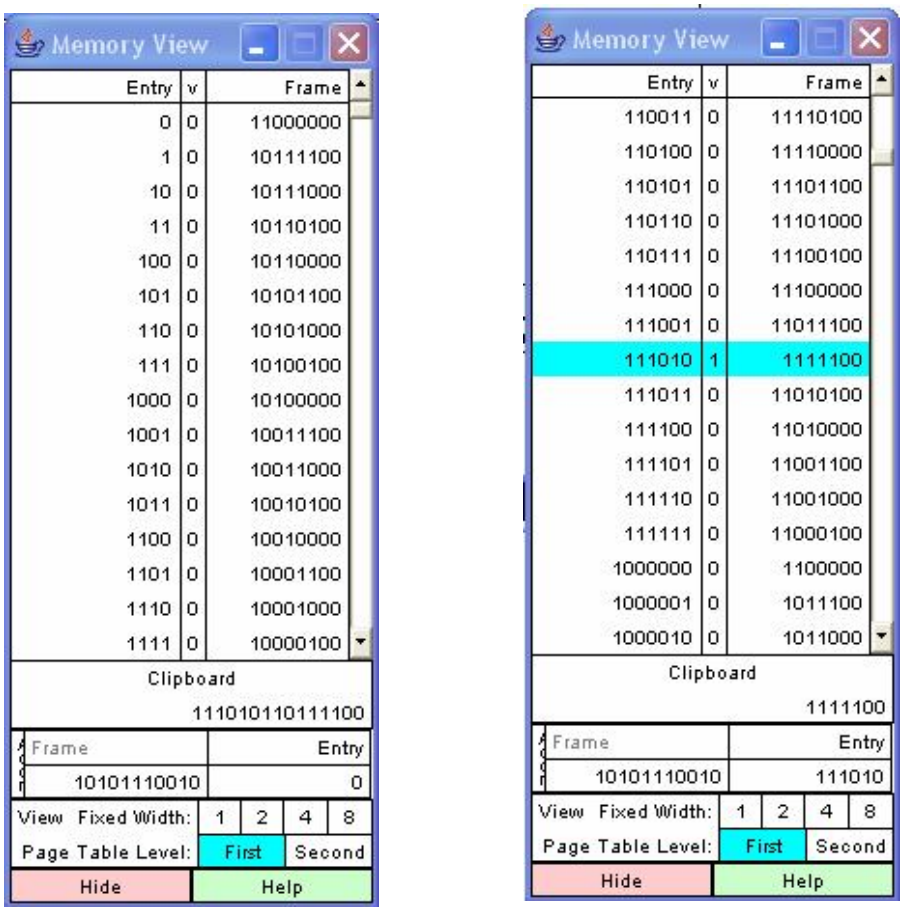


Figure 6: The progress list for a two level page table test.

To look up an entry in this page table, the top level page number must be found. Set the logical address to segment mode to segment the logical address again. Since the second level page number is 9 bits wide, the top level page number contains (30-12-9) = 9 bits. Click on the logical address to segment it into pieces with sizes 9, 9 and 12. Change the logical address to select mode and click on the top level page number. This puts the page number in the clipboard. Click on the **Entry** button in the memory view of Figure 7a to look find this entry in the top level page table. The entry is highlighted as in Figure 7b. If the valid bit were clear, this would be an invalid memory reference. Push the **Invalid Reference** button in Figure 1 to complete the test. Since in this case the valid bit is set, the entry contains the frame number of the second level page table. This frame number has also been put in the clipboard. Push the **Second** button (next to

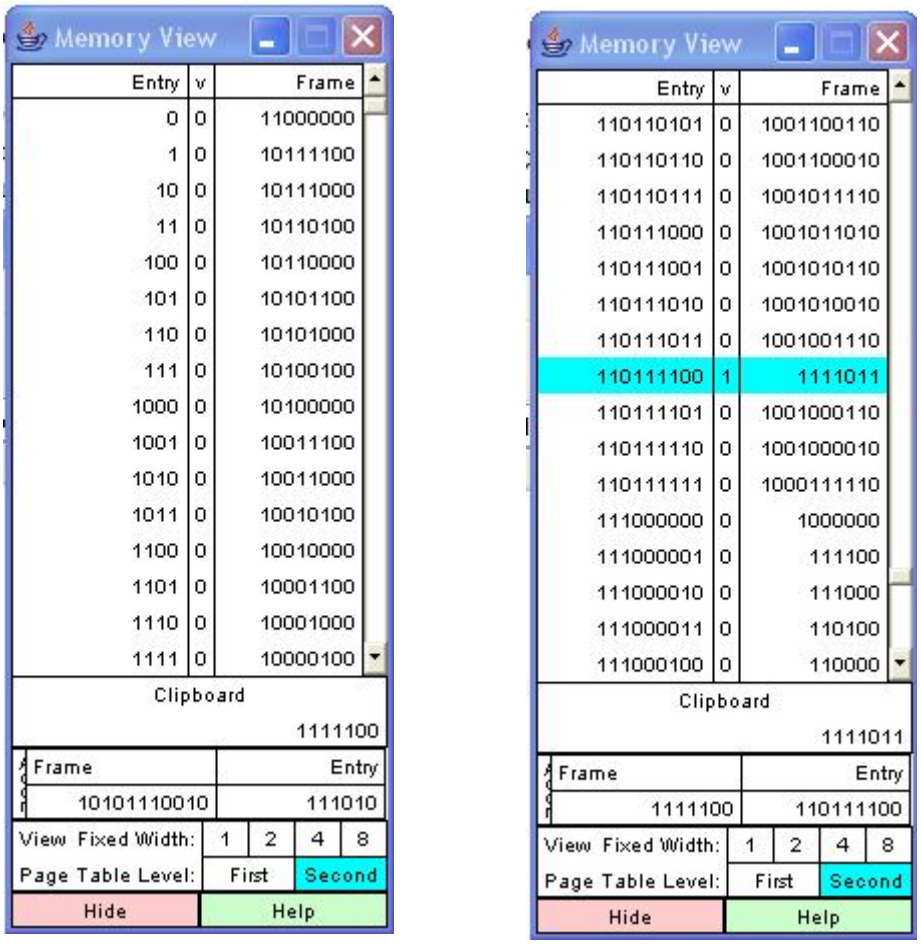
the highlighted **First** button in the memory view of Figure 7b. This displays a second level page table shown in Figure 8a.



(a) (b)

Figure 7: The top level page table view of a two level page table test before a) and after b) a successful lookup.

Click the **Frame** button to set the starting from number of this page table from the value in the clipboard. Now it is time to look up the second level page number in this table. Click on the second level page number field in the logical address to put it in the clipboard. Then click the **Entry** button in the memory view of Figure 8a. The result is shown in Figure 8b. If the valid bit of the highlighted entry were 0, this would indicate a page fault. Click on the yellow **Page Fault** button in Figure 1 to complete the test. Since the valid bit in this case is set, the frame number of the physical address is in this entry. It was also put in the clipboard. Click the frame number field in the physical address to paste the frame number in the physical address. The physical address is now correctly set, so push the yellow **Found Physical Address** button to complete the test. The status should indicate that the correct physical address was found.



(a) (b)

Figure 8: The second level page table view of a two level page table test before a) and after b) a successful lookup.