# Mapping Memory in SourcePoint™

*An Application Note*
*www.arium.com*

## Overview

Modern chip sets (440BX and above) complicate the debugging of Intel Architecture designs. These chip sets hang when a memory access is attempted before the chip set is initialized. This is not an issue during normal operation; but during the debugging process it can cause serious problems.

In-target probes and emulators access memory in order to provide information to the user. Simply viewing the code at the reset vector, before the chip set is initialized, can cause a system hang that requires power cycling the target, the emulator, and sometimes even the host PC.

SourcePoint™ is the only debugger that allows the user to eliminate this problem a priori. By defining the target's memory map in the SourcePoint™ GUI. This memory map can prevent all GUI windows that access memory (memory-based windows) from automatically refreshing when the debugger performs an action that might change the values in memory (e.g., step, go, reset). SourcePoint thus prevents the debugger from accessing memory before the chip set has been properly initialized. This document examines this process and provides suggestions.

## User-Interface

Access to the user-interface of the memory-mapping feature can be found in the Options menu. The first entry in the Options menu is Preferences. See Figure 1.

Clicking on this selection instantiates the Preferences dialog box. There are several tabs along the top of this dialog box. Select the Memory Map tab to change the default options for the memory-mapping feature. Figure 2 shows the default settings of the SourcePoint Memory Map.

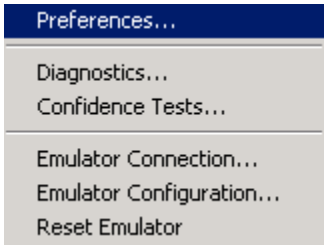



*Figure 1*

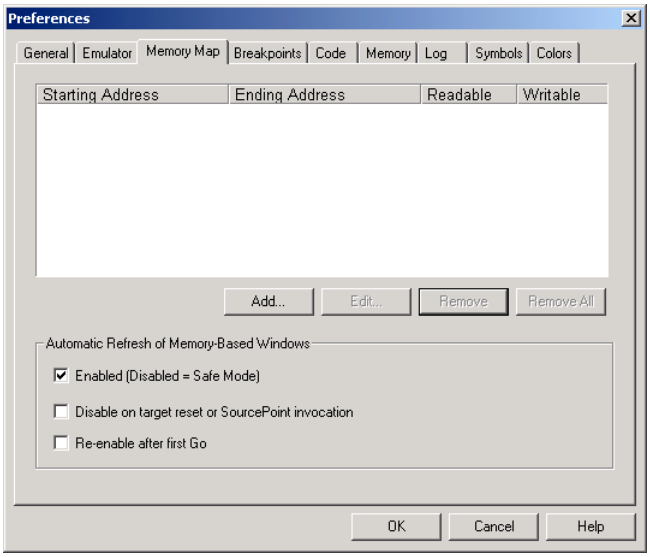
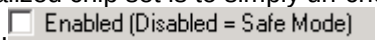



*Figure 2*

The configuration shown in Figure 2 allows the memory-based windows to access any address in memory, valid or not. This setting is the most convenient when the chip set is initialized but causes the most problems when it is not.

The easiest method for preventing accesses to an uninitialized chip set is to simply un-check the box that enables the automatic refresh of memory based windows. 

This setting will allow the debugger to access memory only via the command window. No memory-based window will be allowed to access memory unless the "Refresh" button in that window is pressed. This results in windows that are blank, except for an error message or question marks. See Figures 3 and 4.

The "Refresh" button forces a one-time memory access. If any action occurs that renders the contents of the window invalid (step, go, reset, etc.) then the window will go blank again. In this mode, the "Refresh" button of each memory-based window must be pressed after every step of the processor if the user requires information from that window. This can become tedious, so American Arium added a way of defining the areas of memory where memory accesses are allowed.

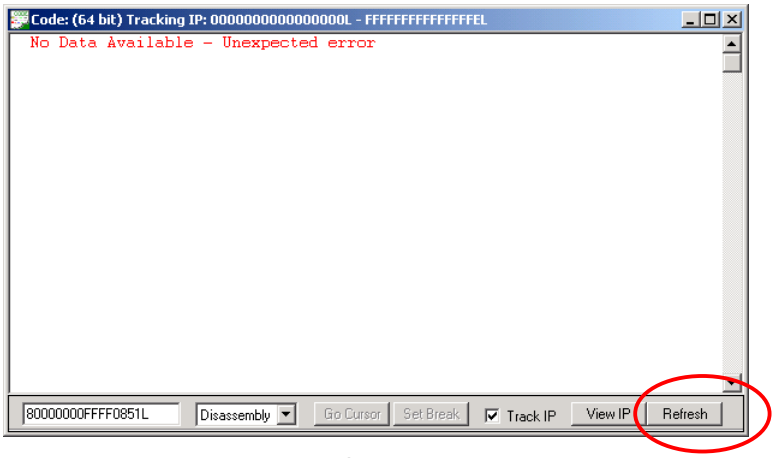



*Figure 3*

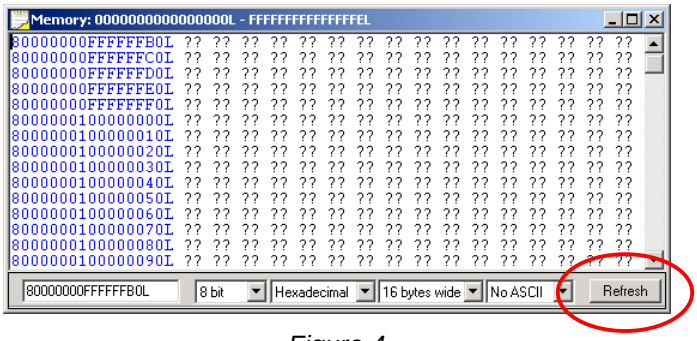
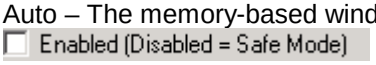



*Figure 4*

## Using the Memory Map

The SourcePoint memory mapping feature allows the user to define ranges of addresses and the read/write rules that SourcePoint is to follow for each range.

There are four columns in the memory map display:

1) **Starting Address** – Beginning of address range
2) **Ending Address** – End of address range
3) **Readable** – Indicates if read accesses are allowed to the address range
   There are three settings:
   a) Always – All attempts to read from this address range are permissible
   b) Auto – The memory-based windows will be able to read from this address range only if 

   is checked.
   c) Never – No reads from this address range are permitted.
4) **Writable** – Indicates if write accesses are allowed to the address range
   There are two settings:
   a) Always – Any write to this address range is permitted
   b) Never – No writes to this address range are permitted

To add an entry to the memory map, click the add button (Figure 5):

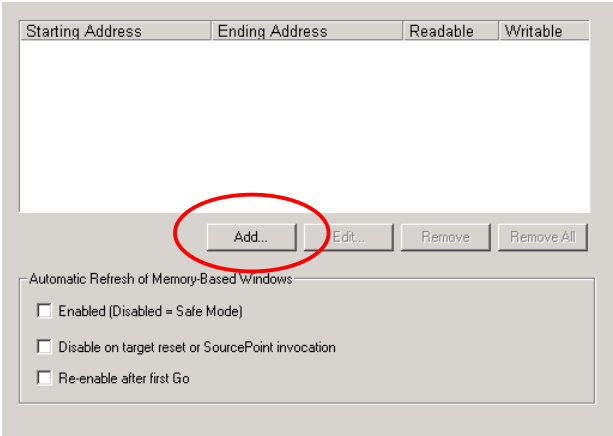



*Figure 5*

This produces the Add Memory Map Entry dialog box (Figure 6), an intuitive interface for defining memory ranges and their settings:

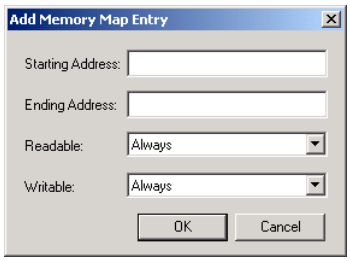



*Figure 6*

The following table shows the effect of various settings on the read/write permissions granted to memory based windows.

| Setting for Address Range | | Auto Refresh Setting | |
|---|---|---|---|
| | | Enabled | Disabled |
| **Read** | Not Mapped | Allowed | Denied |
| | Never | Denied | Denied |
| | Auto | Allowed | Denied |
| | Always | Allowed | Allowed |
| **Write** | Not Mapped | Allowed | Denied |
| | Never | Denied | Denied |
| | Always | Allowed | Allowed |

Note 1: Memory accesses from the Command window obey the same rules as pressing the refresh button in a memory-based window.

Note 2: Entering a number in the address box of a memory-based window has the same effect as pressing the refresh button, except that the starting address shown in the window is changed.

Note 3: When Auto Refresh is disabled, any memory range that is not mapped can be accessed via the refresh button or the Command window.

## Memory Map Suggestions

The settings for the memory map are saved in the project file. Usually the easiest way to set-up the memory map is to use the GUI as previously described and simply close SourcePoint and re-open it. This will make the chosen memory map settings a part of the *wdb.prj* file. An alternative is to save the project under a different name, being careful to load that project when necessary.

There may be times when you will want to modify the memory map before even opening SourcePoint. To do this, simply edit the project file using the following steps:

1) Open the *wdb.prj* file with a simple text editor (Notepad.exe, EditPad, etc.). The *wdb.prj* file can be found in the SourcePoint installation directory (typically C:\Program Files\American Arium\Intel).
2) Search for and delete the [MemoryMap] heading. If there is no heading by this name, go to step 3.
3) Cut and paste the following memory map into the *wdb.prj* file.

```
[MemoryMap]
RefreshWindows=0
Range0=100000000-7fffffff
Range0_Read=Never
Range0_Write=Never
Range1=fff00000-ffffffff
```

Range1_Read=Always
Range1_Write=Never
Range2=0-10000000
Range2_Read=Auto
Range2_Write=Always
*End of map (do not copy this line)*

---

Note 1: Range 0 prevents all accesses above the 4GB boundary

Note 2: Range 1 allows reads but not writes in a common boot code range.

Note 3: Range 2 protects RAM space. The map shown is for a 256M RAM space. The third line from the bottom should be edited to reflect the amount of RAM on the target board.

---

14811 Myford Road, Tustin, CA 92780 Voice: 877-508-3970 toll free within the US, 714-731-1661 outside the US Fax: 714-731-6344 E-mail: info@arium.com
Web www.arium.com