SourcePoint

Getting Started Guide for the

AAEON UP Xtreme i11

Support for v0000 and v0001 boards

Revision 2.20

Contents

Revision History

Revision Number	Description	Date
1.00	Original document, describes v0000 board support	November 28, 2021
2.00	Added content for new support of v0001 (with the Type-C connector removed) AAEON UP Xtreme i11 board	May 30, 2022
2.10	Updated for WinDbg support and other perfective changes	December 3, 2023
2.20	Update for beta release SourcePoint 7.12.52.	March 31, 2024

Welcome!

Thank you very much for your SourcePoint purchase! We appreciate you acquiring our best-in-class debugger, and hope you enjoy using it. We strive to deliver the most powerful, easy-to-use and polished product as possible. So, please feel free to share your feedback directly at our support site at https://www.asset-intertech.com/support/, or via your favorite social media outlet.

As with any new tool, mastering SourcePoint takes an investment in terms of time and effort. JTAG-based debug is a fairly specialized area, and low-level “on the metal” firmware development on x86 platforms is even more so. So, in your use of the tool, you may encounter behavior that seems non-intuitive or even wrong. You may be encountering a tool corner case, a limitation inherent in JTAG or DCI, or even a bug. If so, try a few different options as may be referenced in the Troubleshooting section of this Guide, and if it persists, give us a call. We are happy to support you.

Boards and Cables

The board covered in this document is the AAEON UP Xtreme i11 board, based upon an Intel Tiger Lake CPU. As of February 2022, to address part availability issues, AAEON redesigned the v0000 boards to remove the on-board 40-pin GPIO Bus header, Intel FPGA Altera Max V, and the USB Type-C port. The v0001 boards support DCI in a similar way as the original v0000 boards, but via the Type-A USB3.0 ports.

And then subsequently, AAEON reverted back to the v0000 design. This is the board that is generally available as of December 2023, and is the preferred platform for debugging with SourcePoint. **The v0001 board may be used for UEFI debugging, but is not suitable for Windows kernel debugging (see below)**. But the good news is that all new purchases as of March 2024 are of the correct v0000 version. It's unlikely that anyone reading this has a v0001 board, but this note is here just in case.

The Tiger Lake boards come in four flavors: Celeron, i3, i5 and i7. As of this writing, the Celeron version is best suited for UEFI and Windows debugging. The Celeron boards support all the latest Intel debug and trace features such as Intel Processor Trace (Intel PT), Intel Trace Hub (ITH), Architectural Event Trace (AET), and others.

WARNING:

Do ***NOT*** plug a regular USB cable into the target and attempt to use DCI. Specialty cables, with VBUS snipped, are required; using a regular USB cable may possibly fry your target, or worse.

For the v0000 boards, the main source to purchase the DCI Type-A/C cable is www.designintools.intel.com. Accessing this site requires a confidential NDA with Intel. This target has its Type-C port enabled for DCI. If you have a debug host with a Type-A port, purchase the part # ITPDCIAMCM1MU (1.0 meter) cable. The longer 1.8-meter ITPDCIAMCM2MU will work as well. If your host has a Type-C port, purchase the ITPDCIC2CD2U1M. Using Type-A/C hubs have been seen to work, but are not warranted. Type A/C adapters have been seen not to work.

Other vendor sources may have the Type-A/C or Type-C/C DCI cables available. Feel free to check out other supplier options, or fab your own.

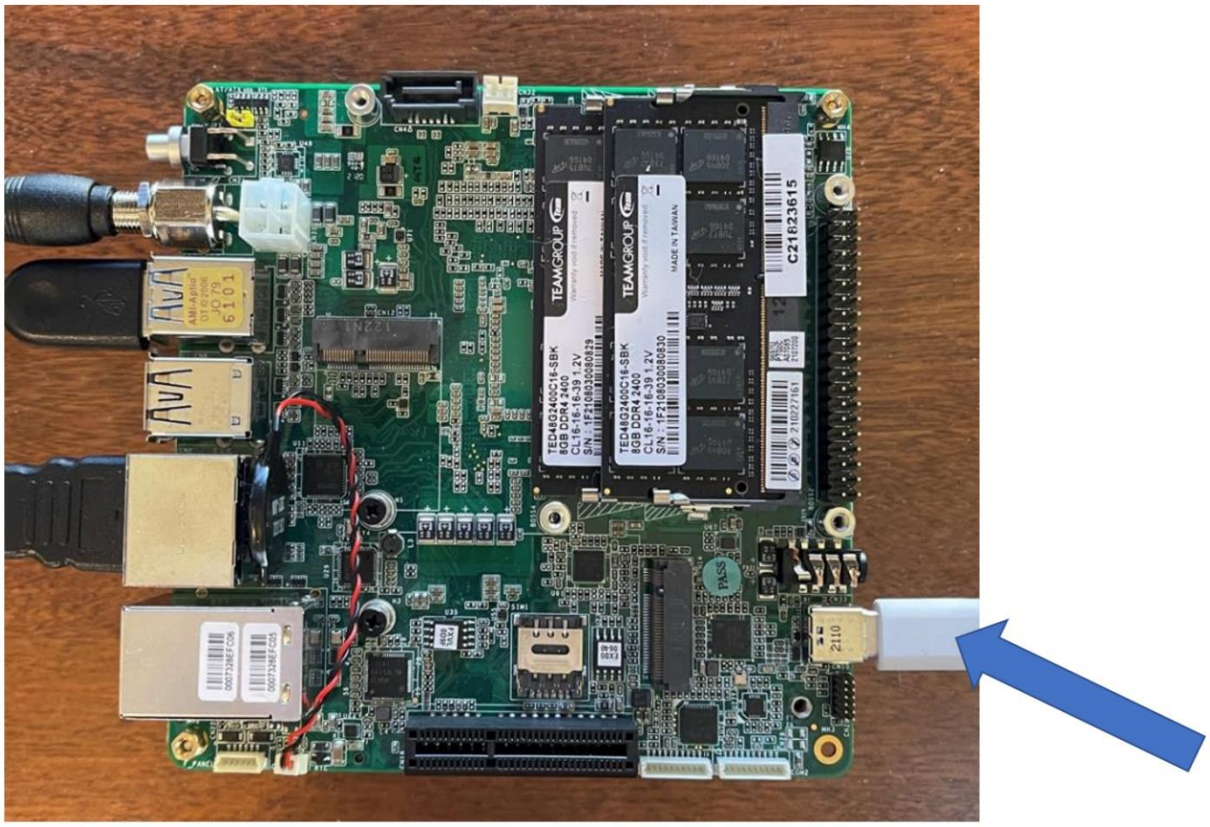
For the v0001 boards, purchase the DCI Type-A/A cable. The good news is that, in addition to www.designintools.intel.com, these cables can be purchased from DataPro: https://www.datapro.net/products/usb-3-0-super-speed-a-a-debugging-cable.html.

BIOS Settings

The AAEON UP Xtreme i11 boards come equipped with an AMI Aptio BIOS that is based upon typical Intel Customer Reference Board (CRB) BIOS.

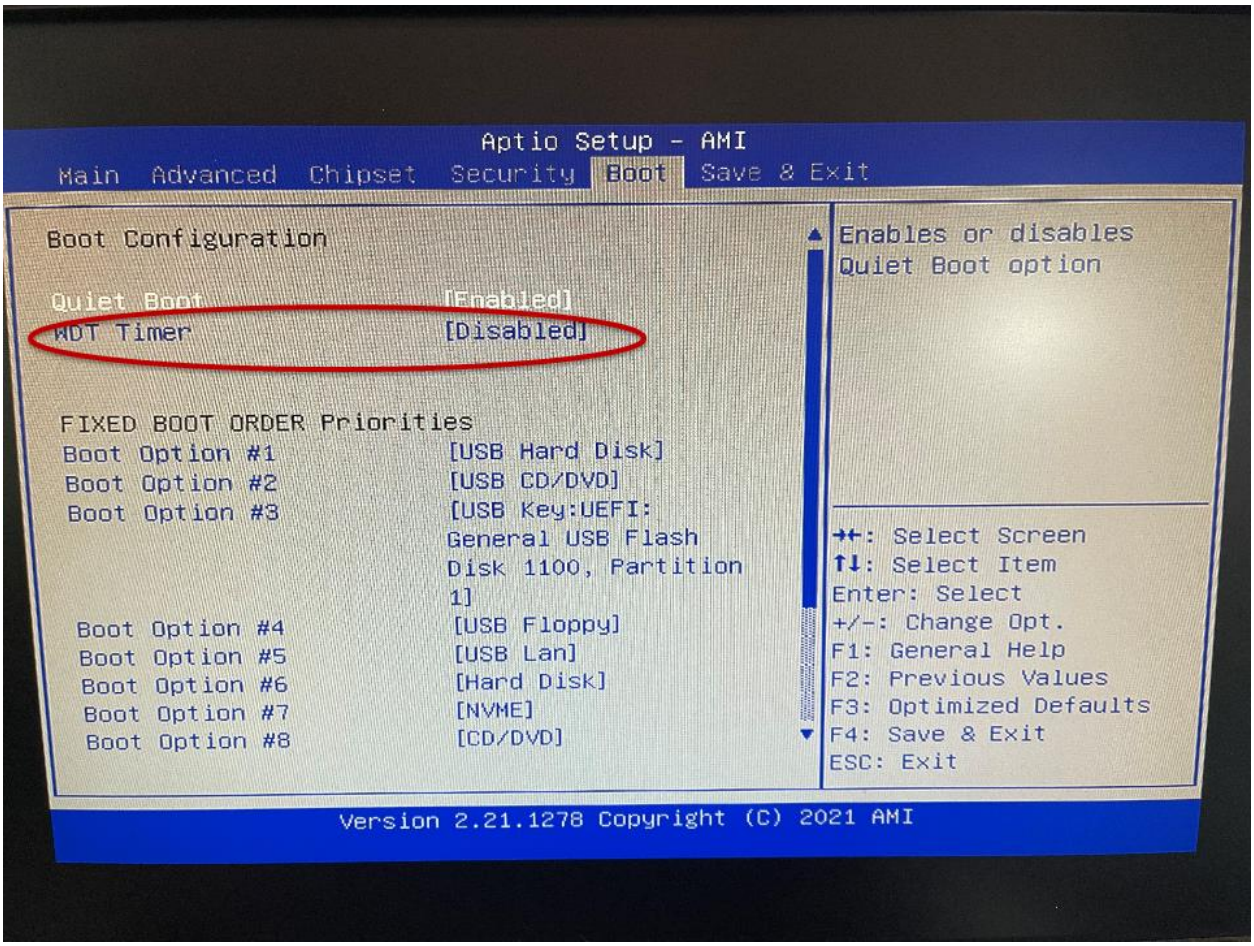
Luckily, the platform comes with all the necessary hardware hooks and firmware straps to support Intel Direct Connect Interface (DCI) out of the box.

For the v0000 board, the USB Type C port on the board is the port of interest:



For the v0001 board, the Type-A/A cable can be plugged into any of the three USB3.0 ports (the blue jacks) available. All three jacks support DCI. The USB2.0 port (the black one) does not support DCI.

There is only one BIOS setting change that must be made for either the v0000 or v0001 boards: to disable the board's Watchdog Timer (WDT). Go into the BIOS Boot menu, and set WDT Timer -> Disabled. If you don't, run-control will be successful, but the board will power-cycle every 30 seconds; putting a real crimp in your debugging!



Power Tip: There is another setting within the CRB Advanced menu to Disable the TCO Timer from being re-enabled by Windows; this is set to Disabled by default when shipped from AAEON, but if you run into issues with run-control stability, you might want to check this:

`CRB Setup > CRB Chipset > PCH-IO Configuration > Enable TCO Timer` must be set to `Disabled`.

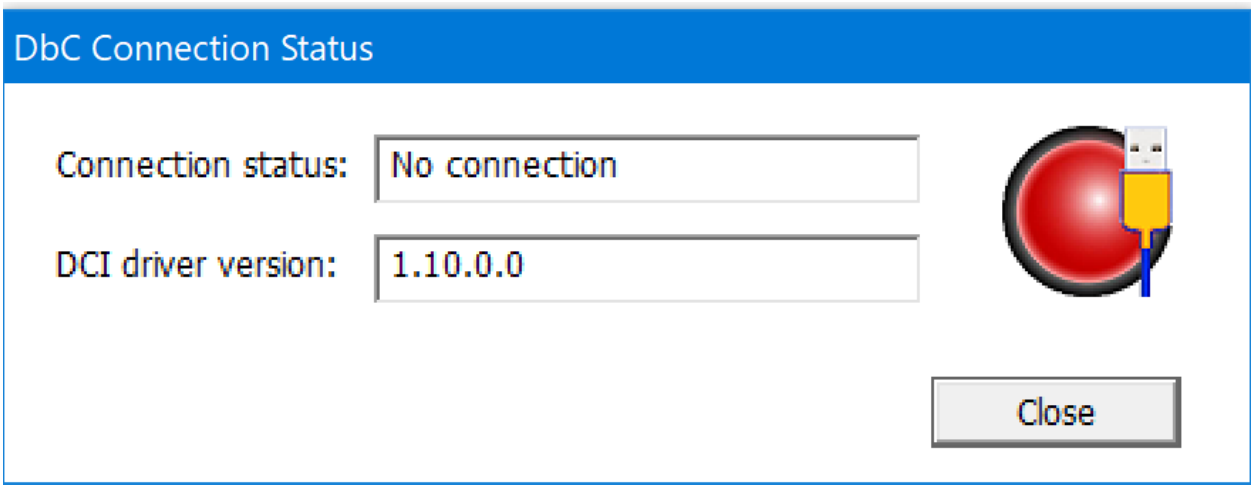
Editor's Note: the password to the CRB Advanced BIOS setup is "upassw0rd" (yes, that is a "zero").

You are now ready to test your connection, and then launch SourcePoint and begin debugging.

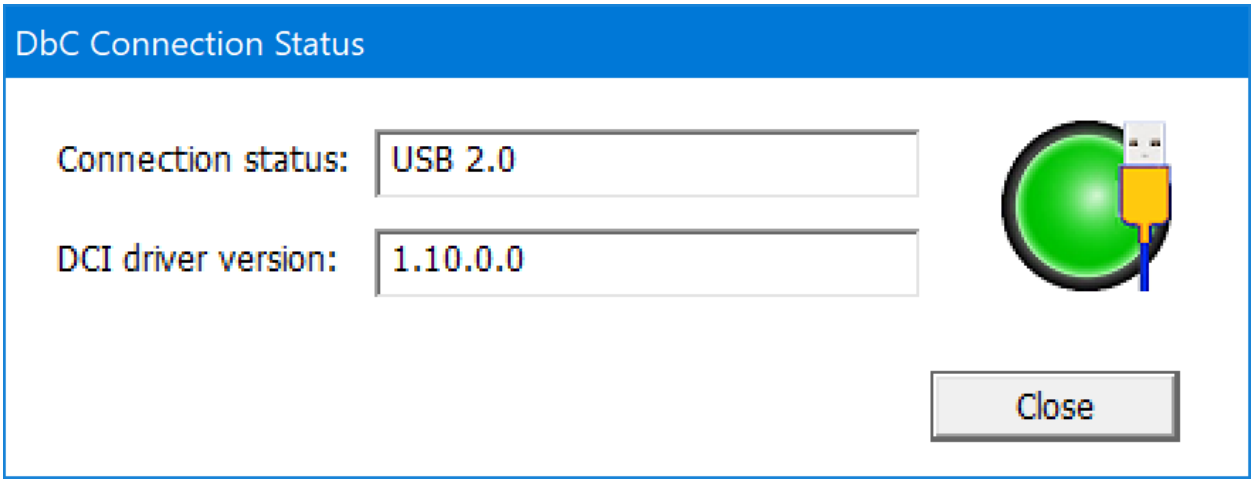
DbCStatus.exe: Red is Bad, Green and Yellow are Good

Luckily, there is a convenient application in the SourcePoint install directory that will tell you that the DCI driver is successfully installed on your computer, and it is possible to make a connection between SourcePoint and the target.

Navigate to `C:\Program Files (x86)\Arium\SourcePoint 7.12.XX` (where XX is your current SourcePoint release), and launch the `DbCStatus.exe`. You should see the red ball, indicating that there is no connection:



For the v0000 board, ensure that the Type-C cable is firmly connected to both the host and target, power up the UP Xtreme i11. In a moment the ball should turn green:

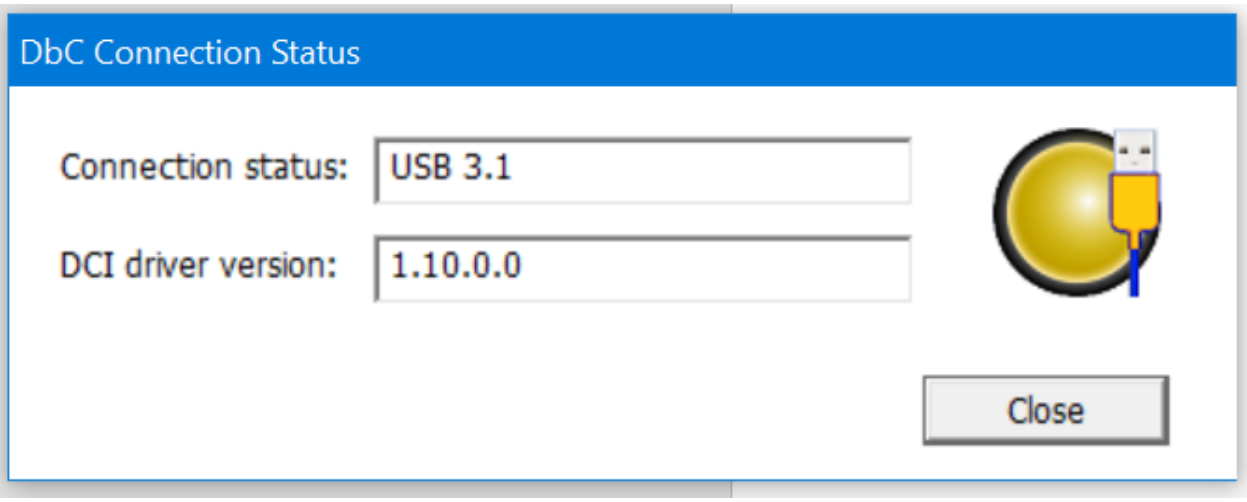


Let the platform boot to the UEFI shell. Congratulations! You have a working DCI connection. It's smooth sailing from here.

Power Tip: If you are using SourcePoint WinDbg for debugging this board, and have Windows already installed, there may be situations where you want to go to BIOS setup

before booting all the way up to Windows. In this case, press F7 after powering on to stop at the BIOS setup.

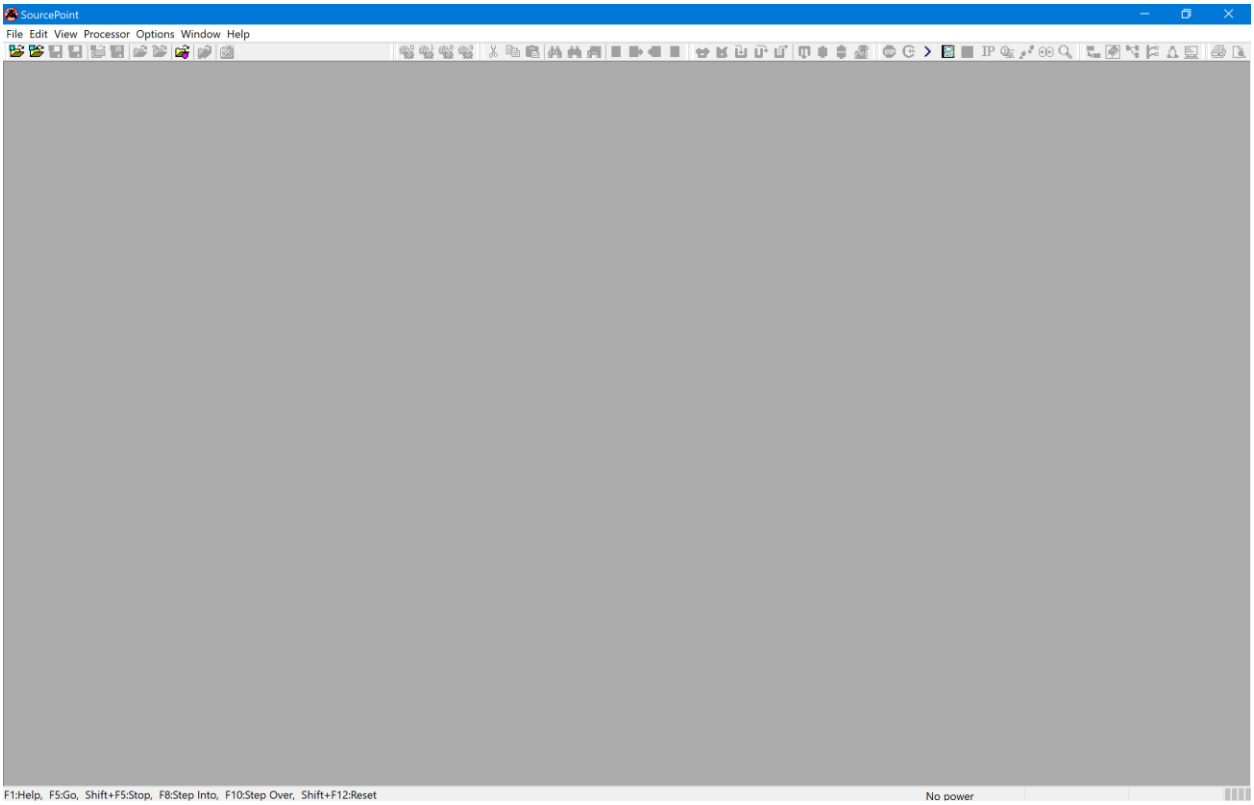
For the v0001 board, the ball will be yellow, versus the green. This is due to a flash firmware strap setting, which changes the target behavior somewhat, as you'll see below; but otherwise does not affect UEFI debugging.



Is the ball still red? Try cycling power on the target (don't just power cycle with the reset button; physically remove the power plug, then plug it back in again). Still red? See our Troubleshooting section.

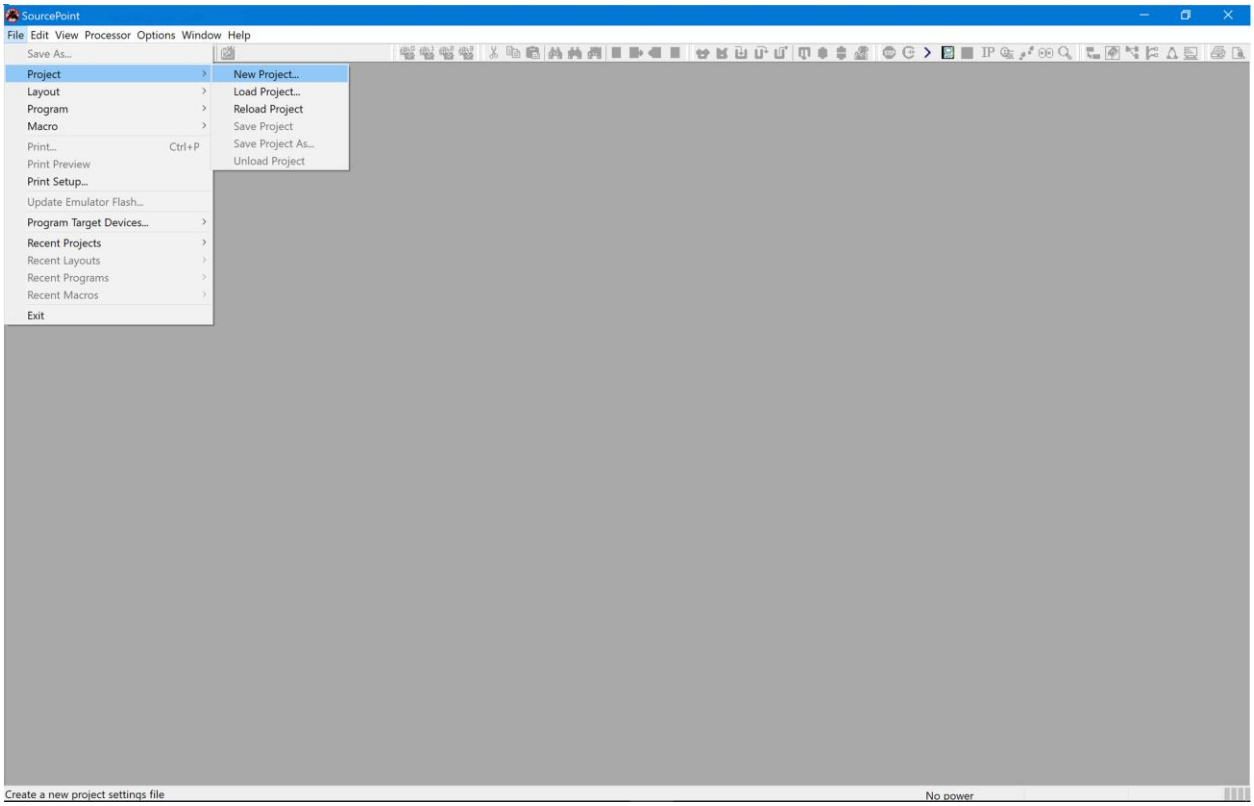
Getting Started with SourcePoint

When you launch SourcePoint for the first time, you will see the main screen, mostly gray:

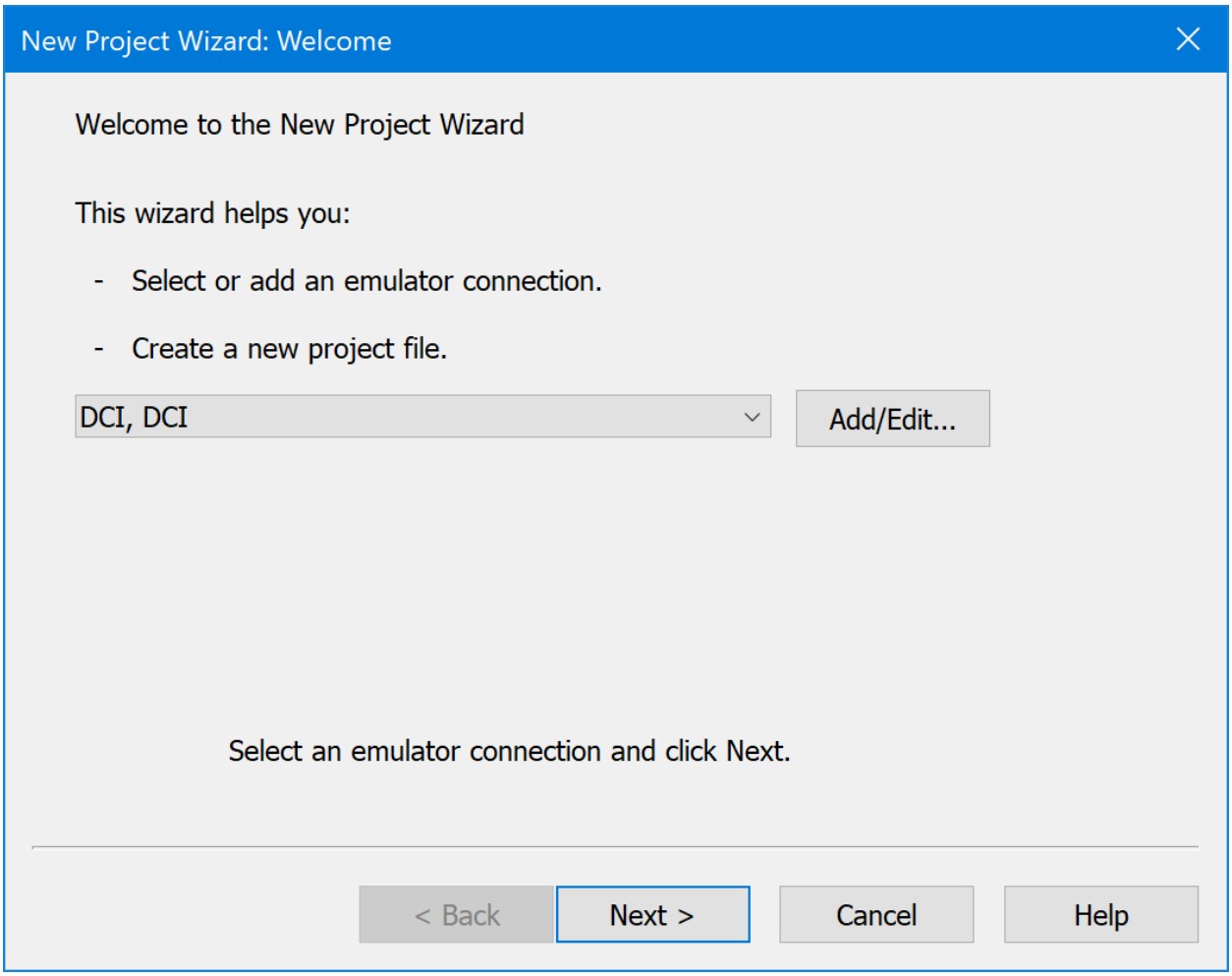


SourcePoint uses Projects (files with suffix .prj) as containers for your debugging session. You can create as many Projects as you want, with all your own preferences saved. Often, once you have the SourcePoint Project configured to your liking, you'll save it and use it repeatedly during your separate debugging sessions. Other users may wish to save a separate Project for each separate debugging session. That's really a matter of user preference and what you're debugging – it's your choice.

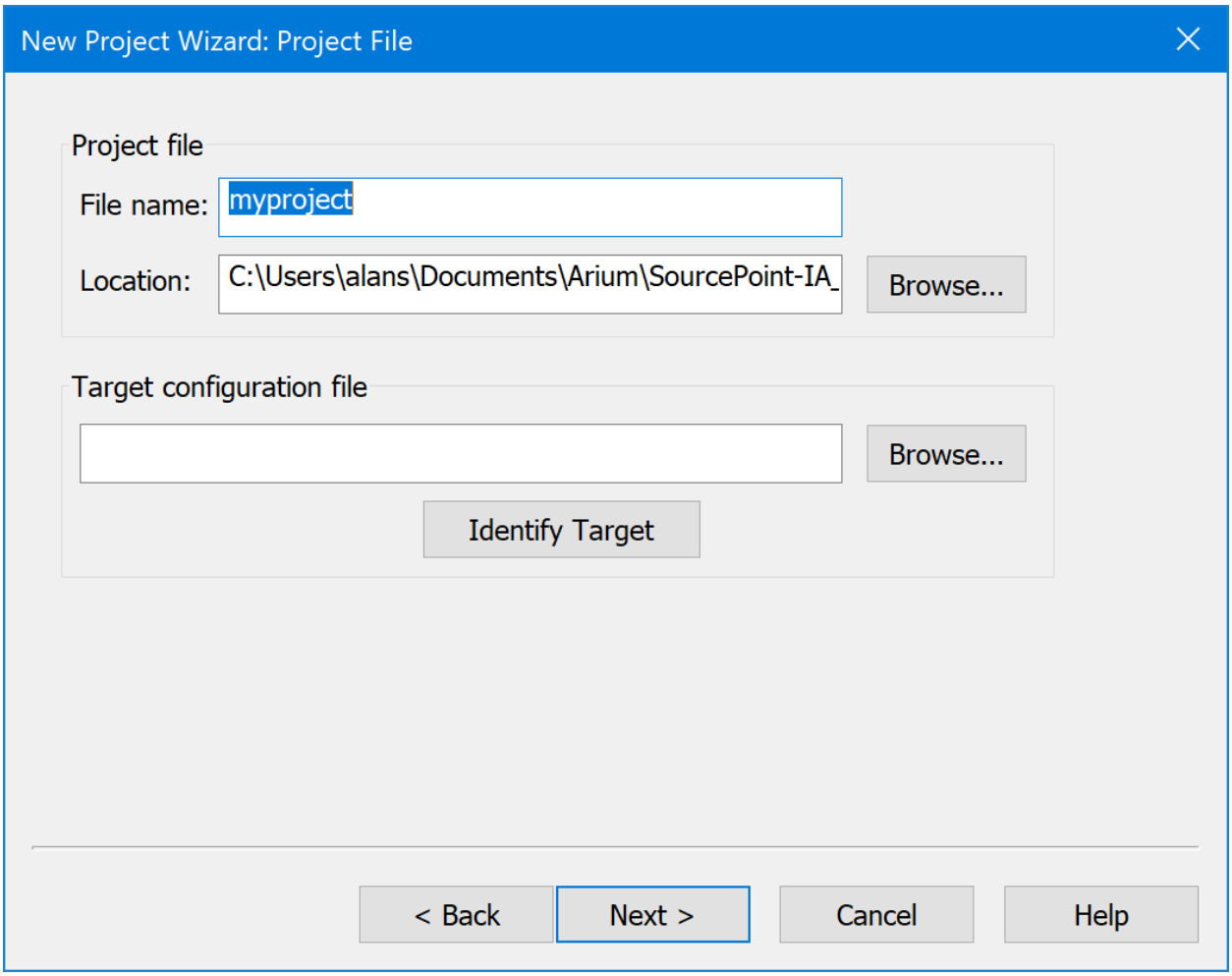
Now it's time to create the Project. Under `File > Project…` select `New Project`:



You'll be presented by the New Project Wizard (NPW). The emulator connection should be via DCI:

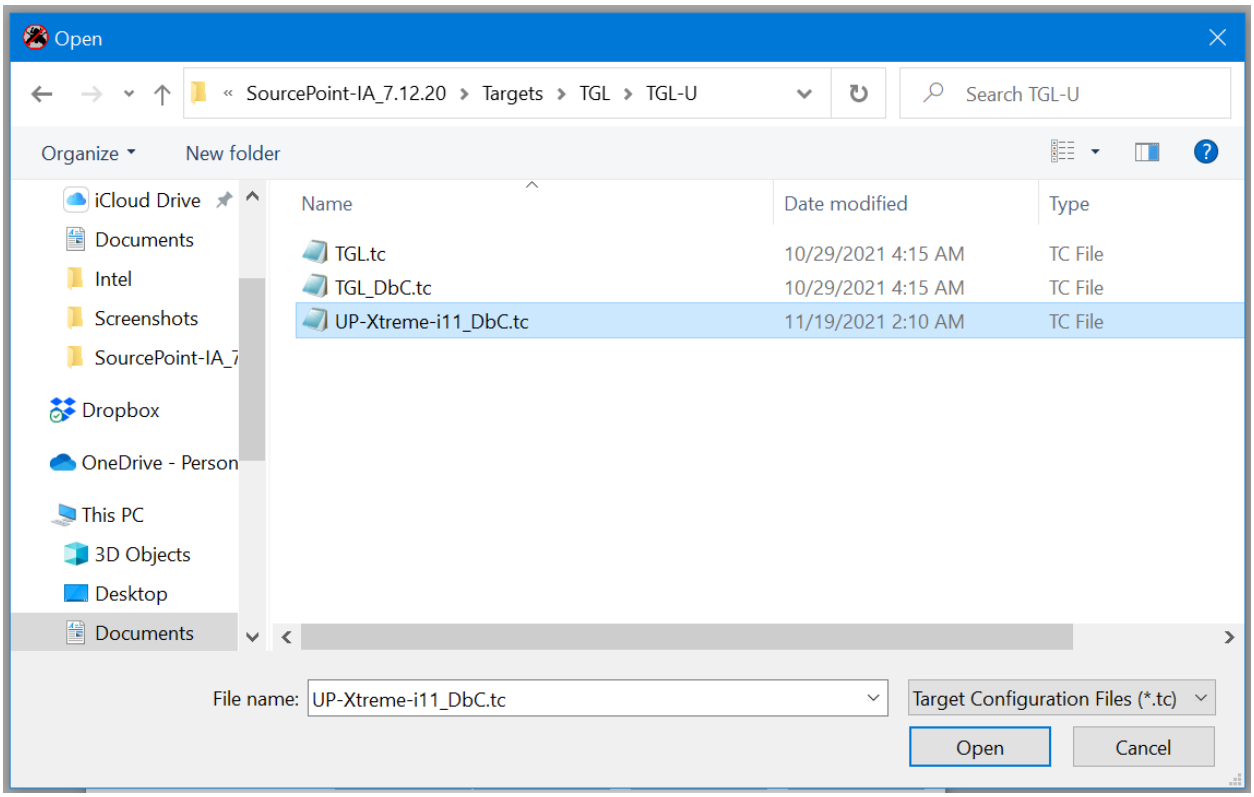


After hitting `Next`, you'll be prompted for the Project Name, Location to store the Project, and the location of the Target Configuration file:

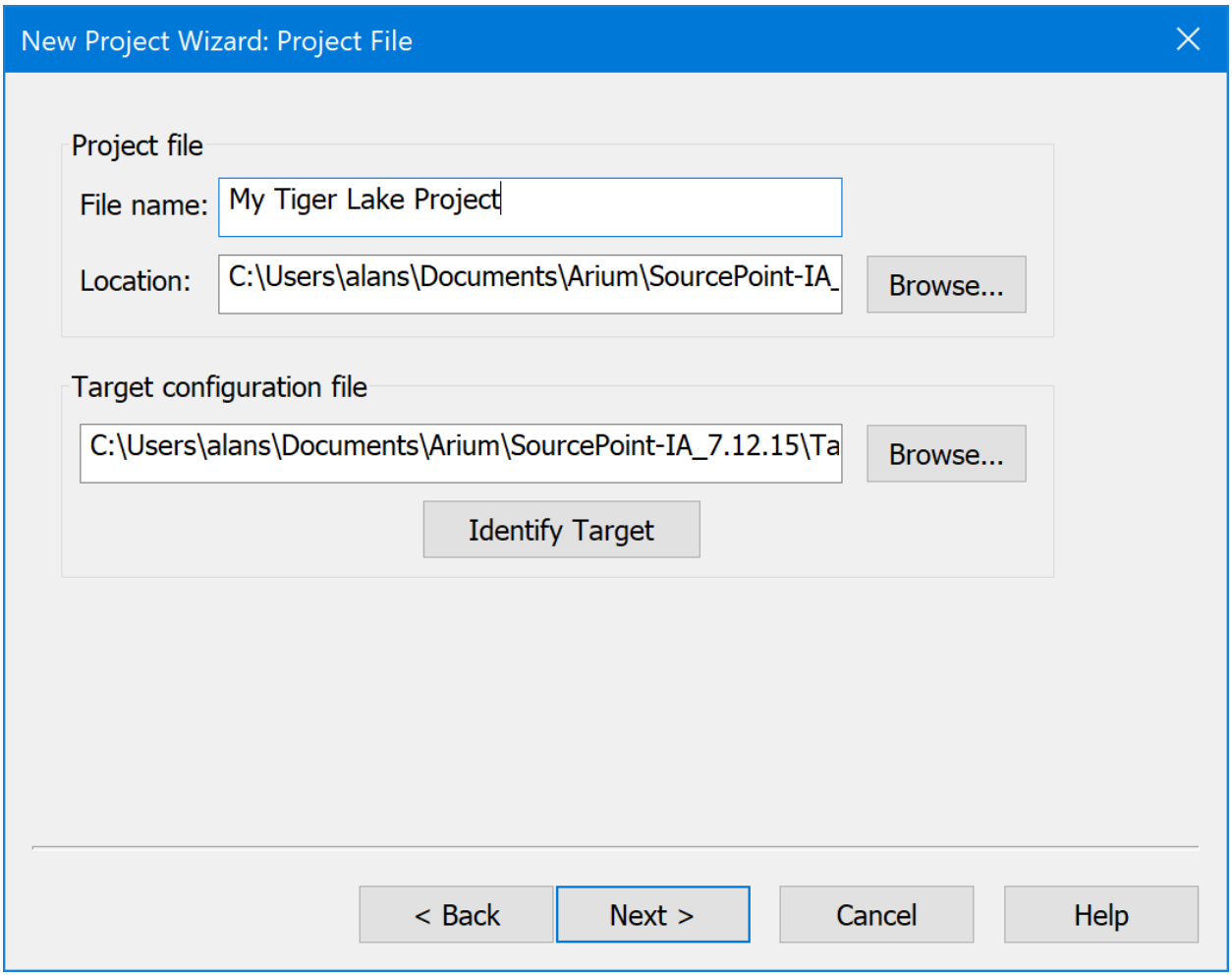


Note that the Target Configuration (TC) files located in `C:\My Documents\Arium\Targets` are used in conjunction with the `jtag-devices.xml` file to define the specific silicon and SourcePoint settings necessary to ensure a successful DCI connection.

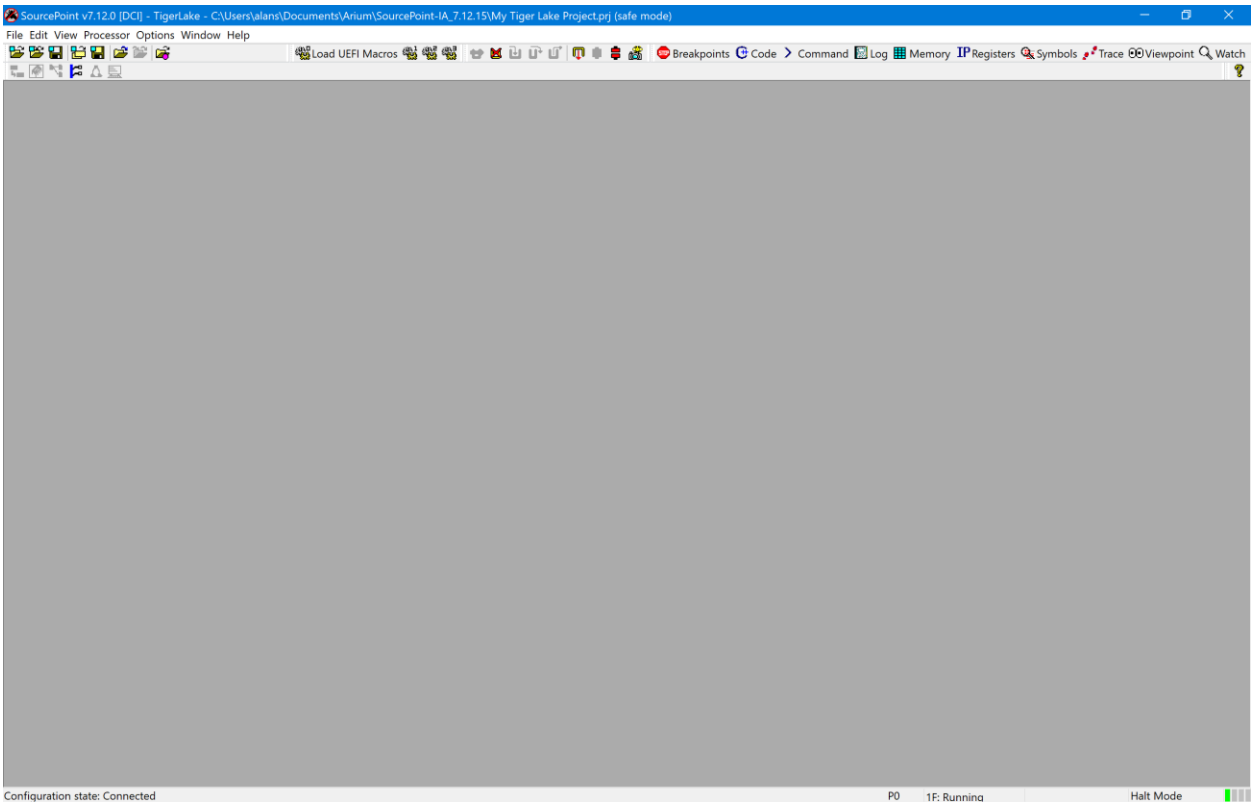
For the UP Xtreme i11 boards, custom TC files have been created, so you shouldn't do an `Identify Target` to automatically select the TC file of interest. Rather, manually select the specific TC file that is customized for this target (`TGL\UP-Xtreme-i11_DbC.tc`) and hit Open:



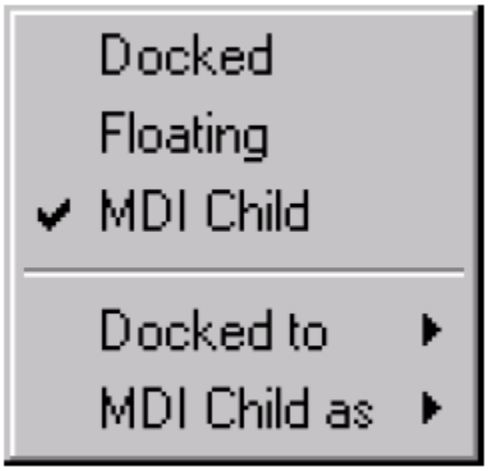
Then, your screen should look something like this:



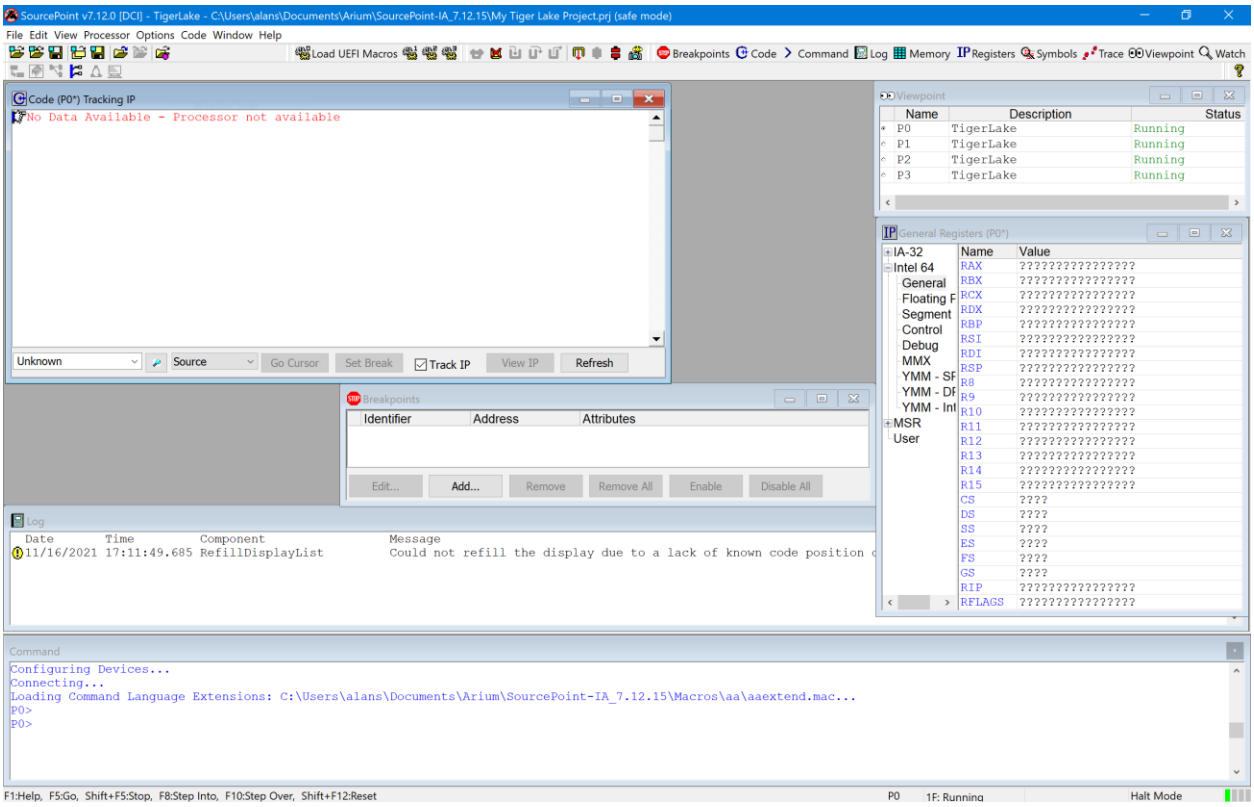
Hit `Next`, then `Finish`, and SourcePoint should successfully connect to the target. You should see "`JtagTest: Successful operation`" followed by "`Configuration state: Connected`" in the Status bar at the bottom left:



Now the fun part begins. Click on the buttons at the top to set up the Viewpoint, Code, Command, Registers and other windows to your own preference. Move the windows around and resize them to take best advantage of your available screen real estate. You can right-click in the title bar of each window to change its type and, for example, to dock the window to the bottom, right side, etc.

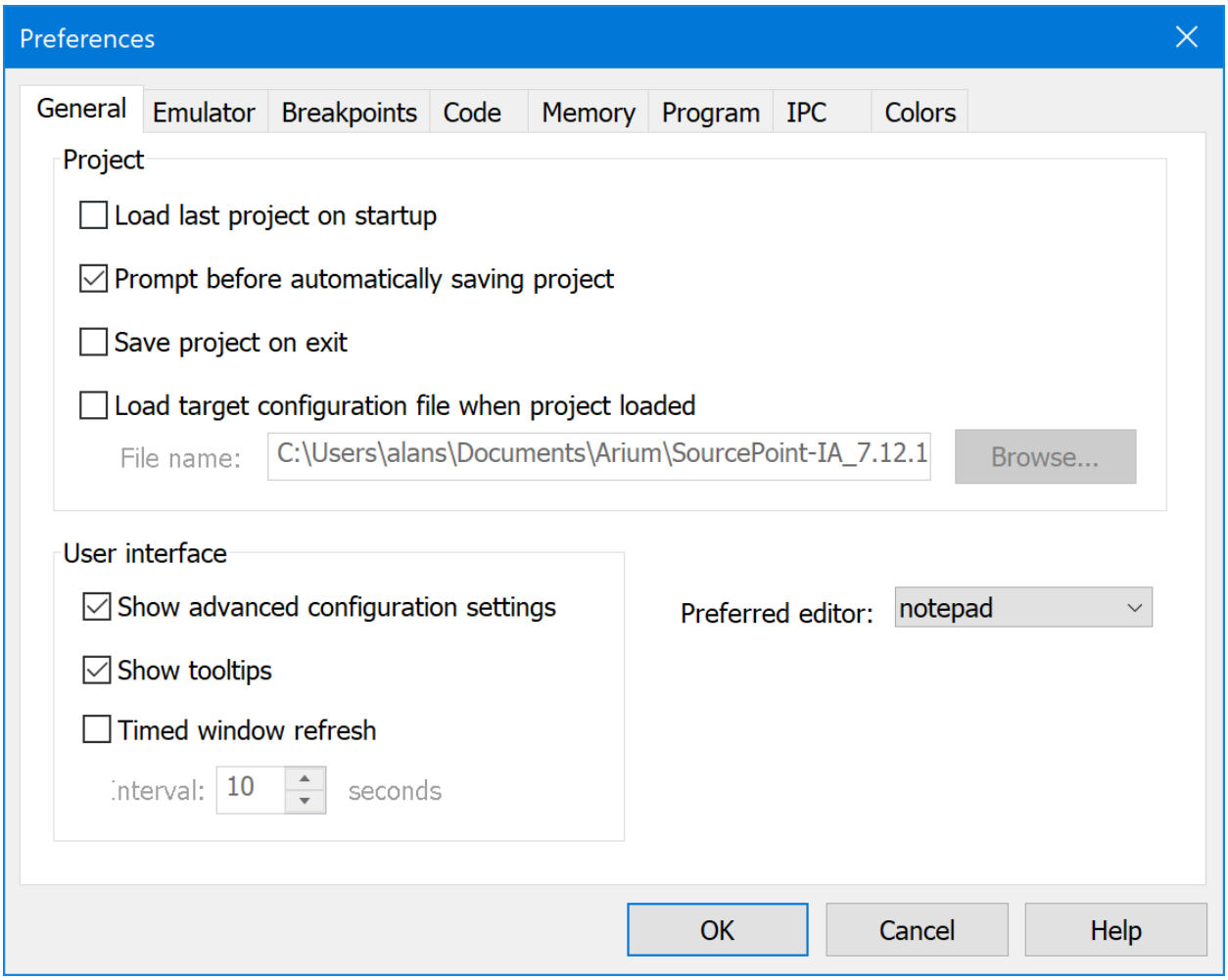


A sample layout is below:



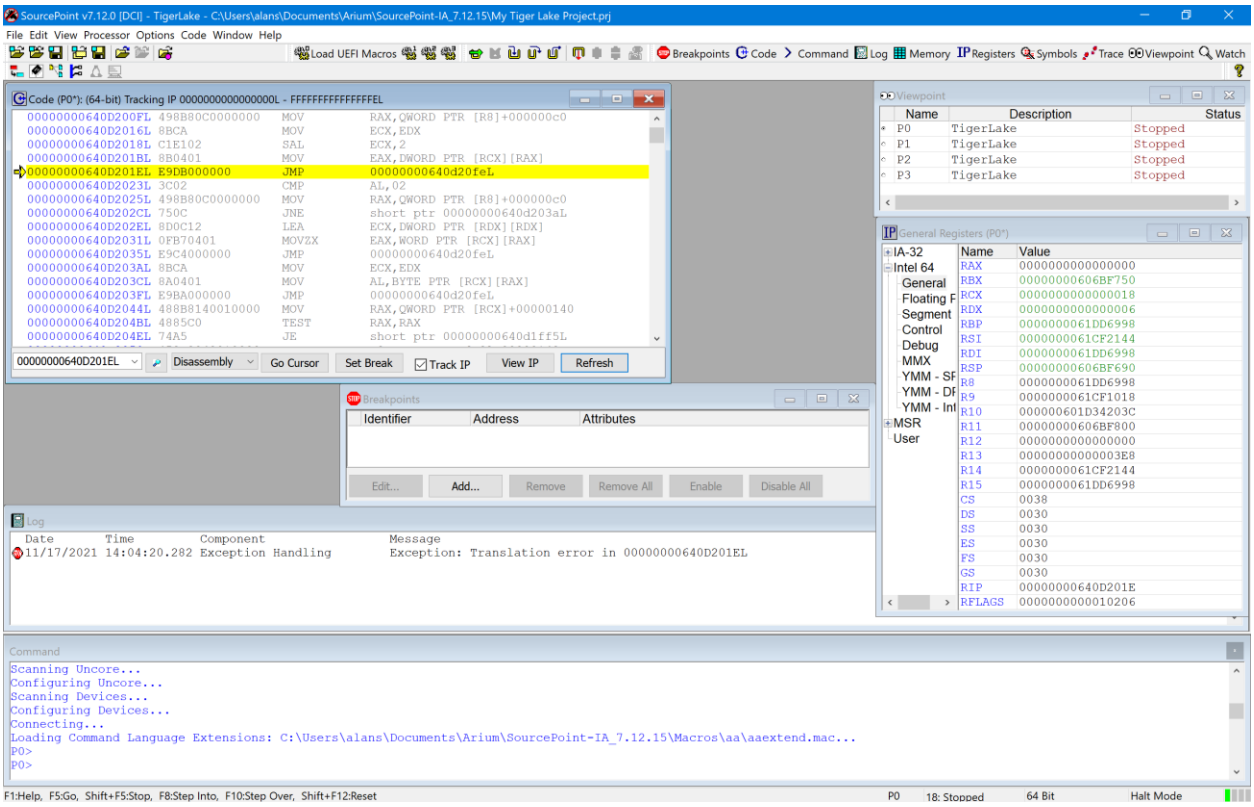
Power Tip: the window layout can be saved as a separate file to your Project. That way, when you create a new Project, you can just Load the layout file separately, without having to create and move the windows around again. Choose `File > Layout > Save Layout…` to create a .lyt file, and then do a `Load Layout…` to save yourself time every time you create a new Project.

Power Tip: Exploring `Options > Preferences…` and some of the other menu items may give you some more labor-saving ideas. For instance, I like to ensure that both `Load last project on startup` and `Save project on exit` are disabled. This gives me more control on entry and exit from the application:



This is a good point to do a `Project > Save Project…` That way, you don't have to start all over, if for some reason your project gets messed up.

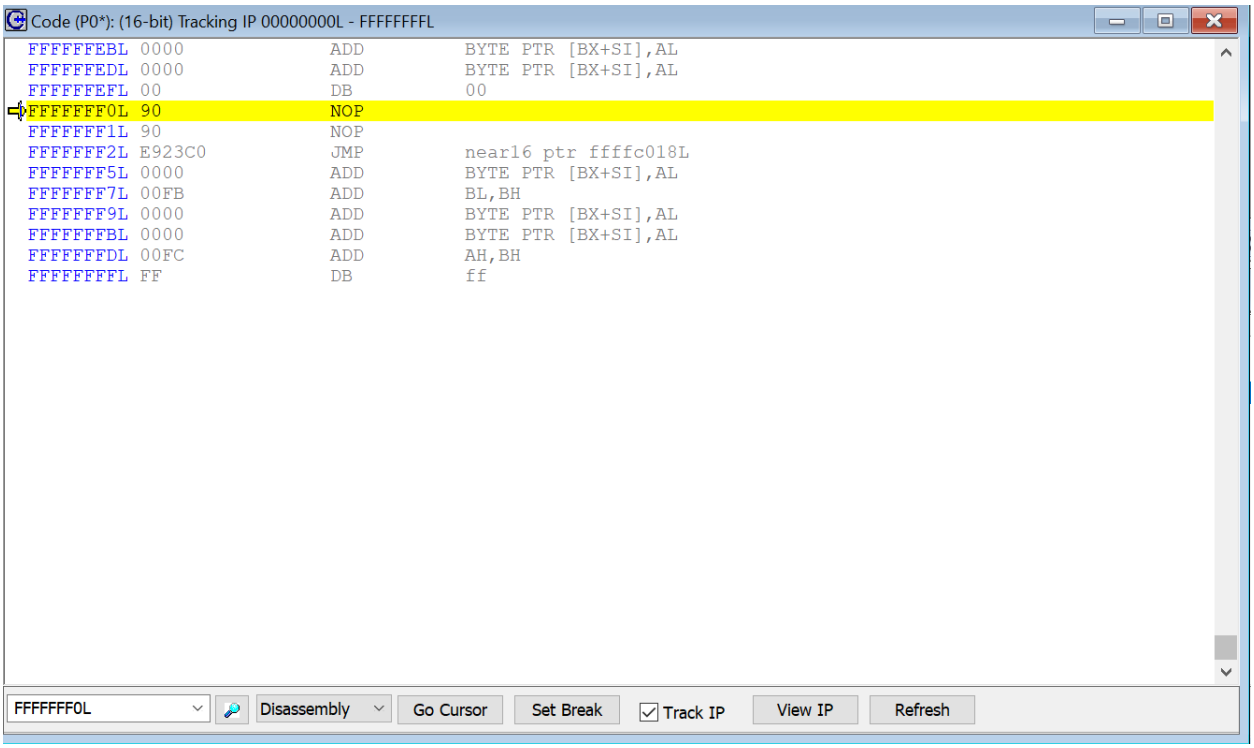
At this point, you are ready to fully begin your debug session. Many of the operations can be accessed via the toolbar at the top of the screen. You can issue a `Stop` on the target, `Step Into`, `Reset` it and halt at the reset vector, set some breakpoints, `Go` to the breakpoint, and so on.



Hit the `Refresh` button in the Code window after the first Stop. This is only necessary once; the Code window will automatically refresh with all run-control operations (stop, go, single-step, etc.) afterwards.

Refer to the SourcePoint User Guide in your install directory for detailed instructions on using all of the tool's features.

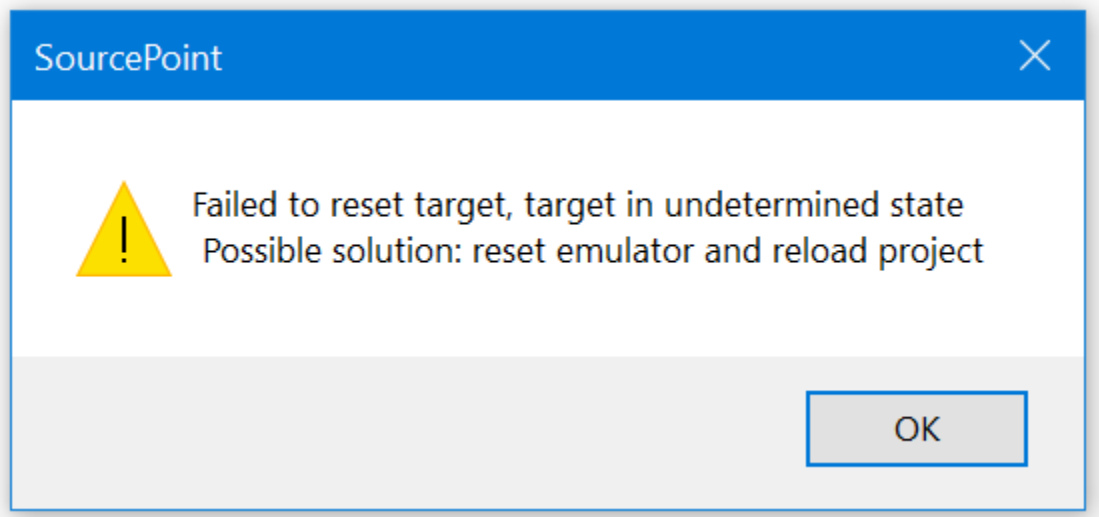
The v0000 board will halt automatically at the reset vector when you hit the SourcePoint Reset button:



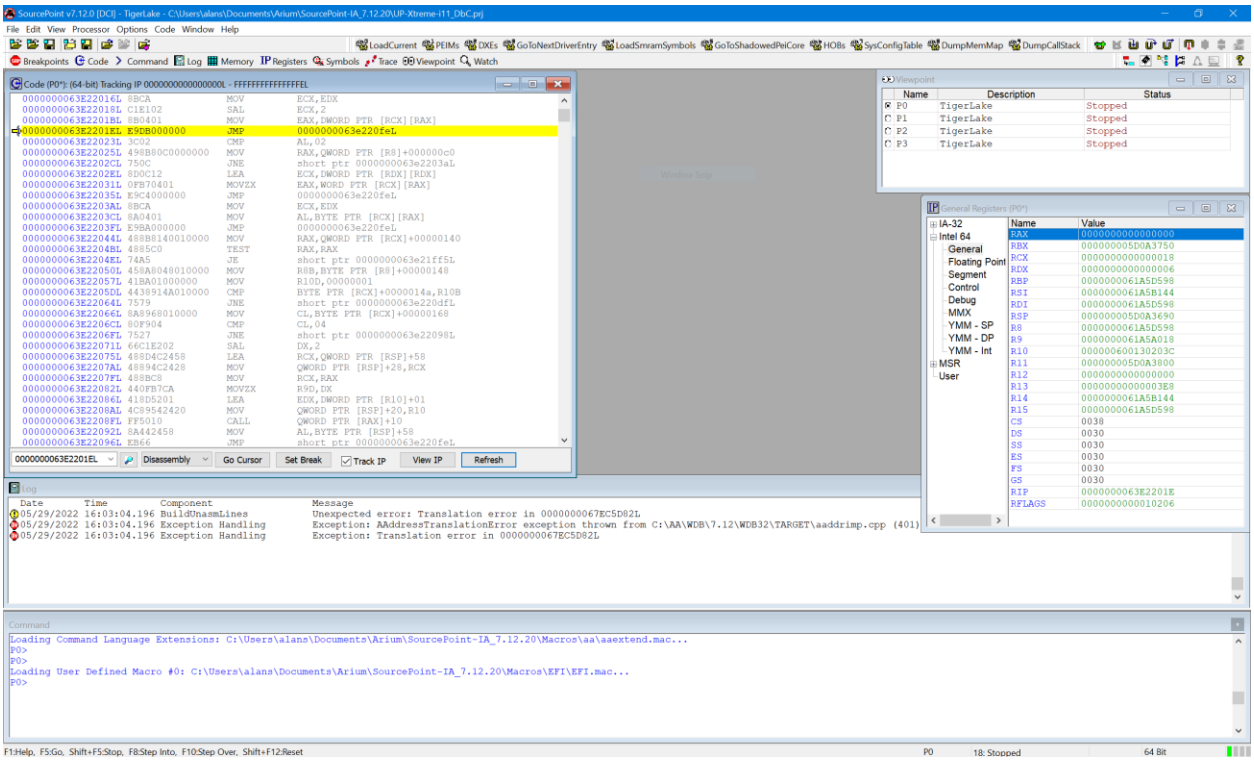
This is desirable, because often you wish to start at early SEC or PEI to debug the portion of UEFI of interest. You'll also need to halt at the reset vector in order to set up Architectural Event Trace (covered later in this document).

But, the v0001 board has an incomplete implementation of DbC2, as you can see from the yellow ball reference earlier, so a couple of extra steps are needed to get there:

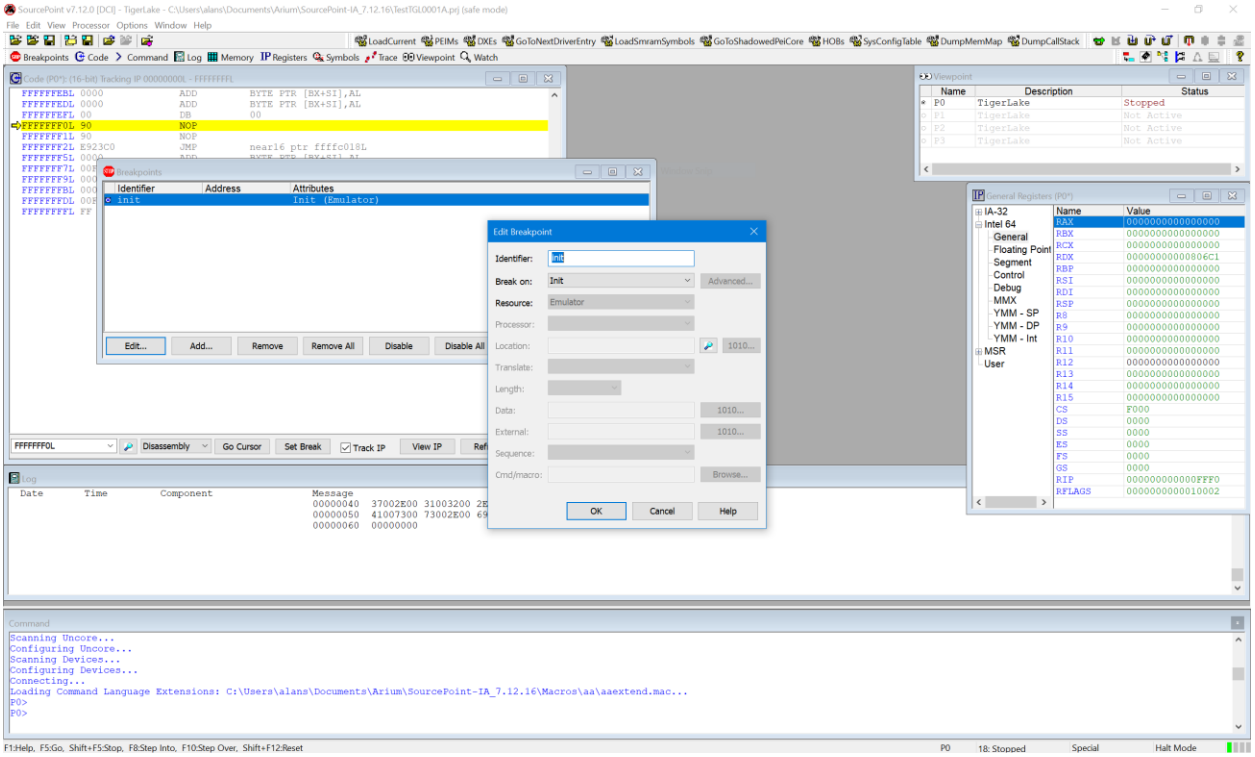
Firstly, after hitting the Reset button, an error message is thrown:



Hit `OK`, and then hit `Go`, to boot again to the UEFI shell (or as far as it will go before stopping autonomously). Then `Stop` and `Refresh` in the Code window if needed:

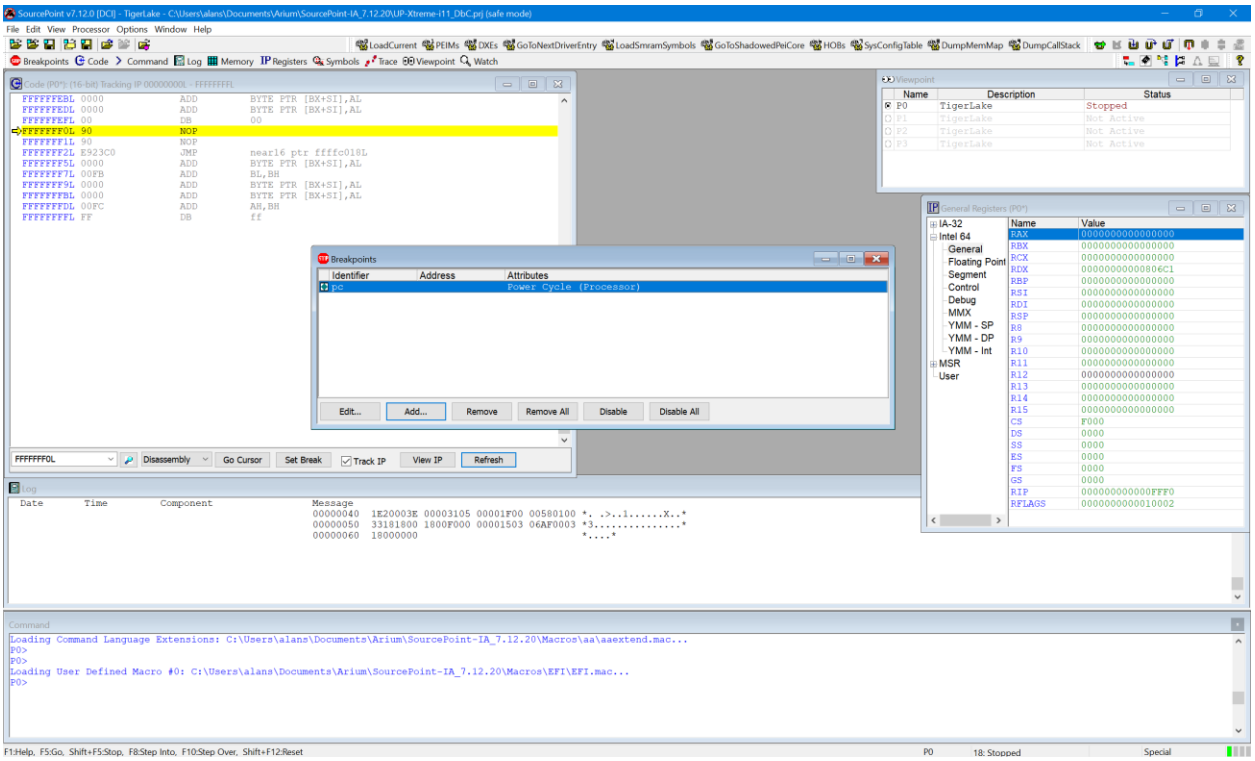


Click on the `Breakpoints` button, and then set an Init breakpoint:

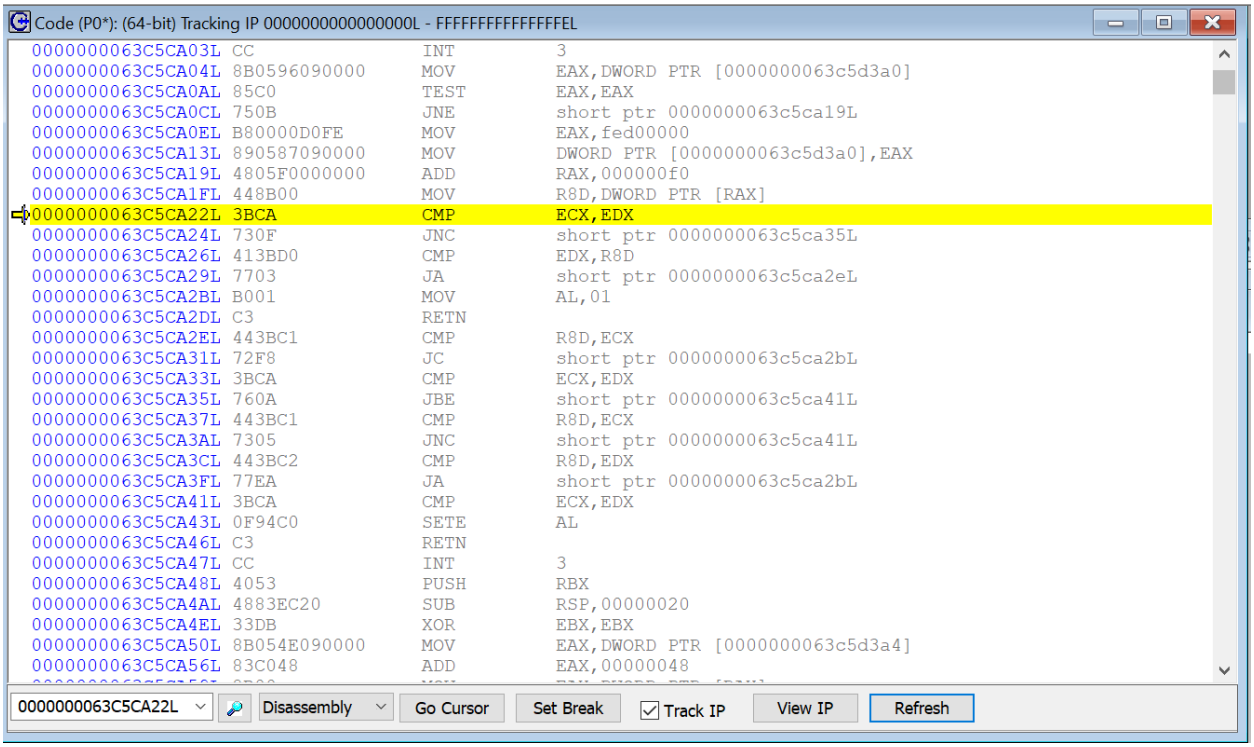


Hit `Go`.

Hit the `Reset` button again, and wait 20 seconds. Voila! You're at the reset vector:



Note that for the v0001 board, hitting Go will not necessarily take you all the way up to the UEFI shell; rather, it will break (Stop) for an Unknown Reason autonomously somewhere in DXE, often right after the splash screen display.



This is a “feature” of either the different DbC2 implementation, or the fact that this BIOS is resetting the XHCI controller – you’ll hear the beep from the DbCStatus.exe application. But regardless, SourcePoint will retain control of the target, and you can continue debugging.

Congratulations, you have mastered SourcePoint’s basic capabilities, and are using run-control. Many users are content to just use these basic operations, because run-control by itself is very powerful. However, if you wish to master the product and use some of its more advanced features, read on.

Advanced Topics: Using Trace

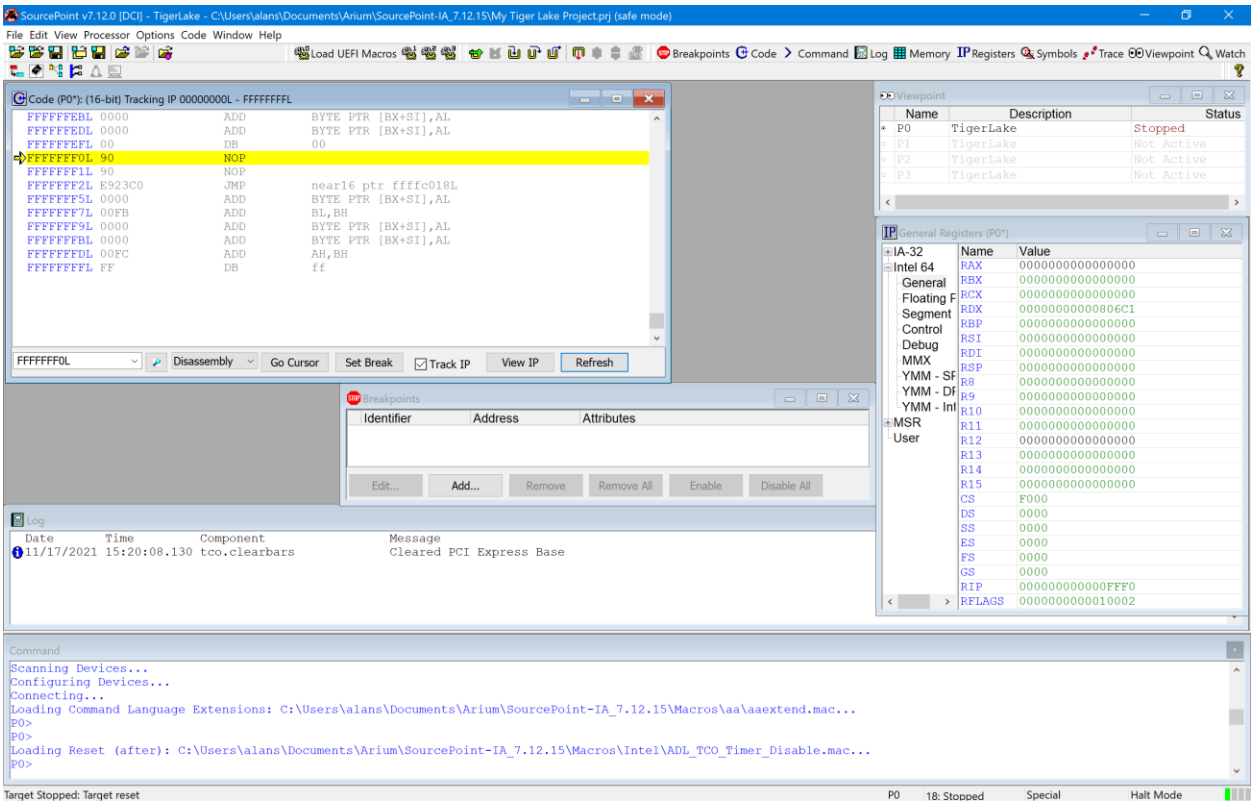
Trace is by far one of the most useful debugging utilities for triaging the most difficult, hard-to-reproduce bugs. Fortunately, the Tiger Lake CPU is equipped with all the latest-and-greatest trace logic, and SourcePoint supports them all.

Let's look at a few of them, and how to configure their use in SourcePoint.

First Step: Configuring the Intel Trace Hub

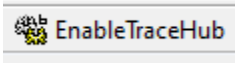
Event tracing on the TGL platform is accomplished by the Intel Trace Hub (ITH). Fortunately, using DCI, events supported by the ITH can be streamed directly out of system reset. The one limitation that exists is that some events (like Port IN/OUT tracing) happen so frequently at some points of the boot process that they overwhelm the capacity of the USB 2.0 (DbC2) connection and event processing, and thus cause buffer overflows – but these should be rare as long as the events collected are relatively close to the debug point of interest.

The first thing to do is to configure the ITH. Reset the target and halt at the reset vector, address FFFFFFF0:



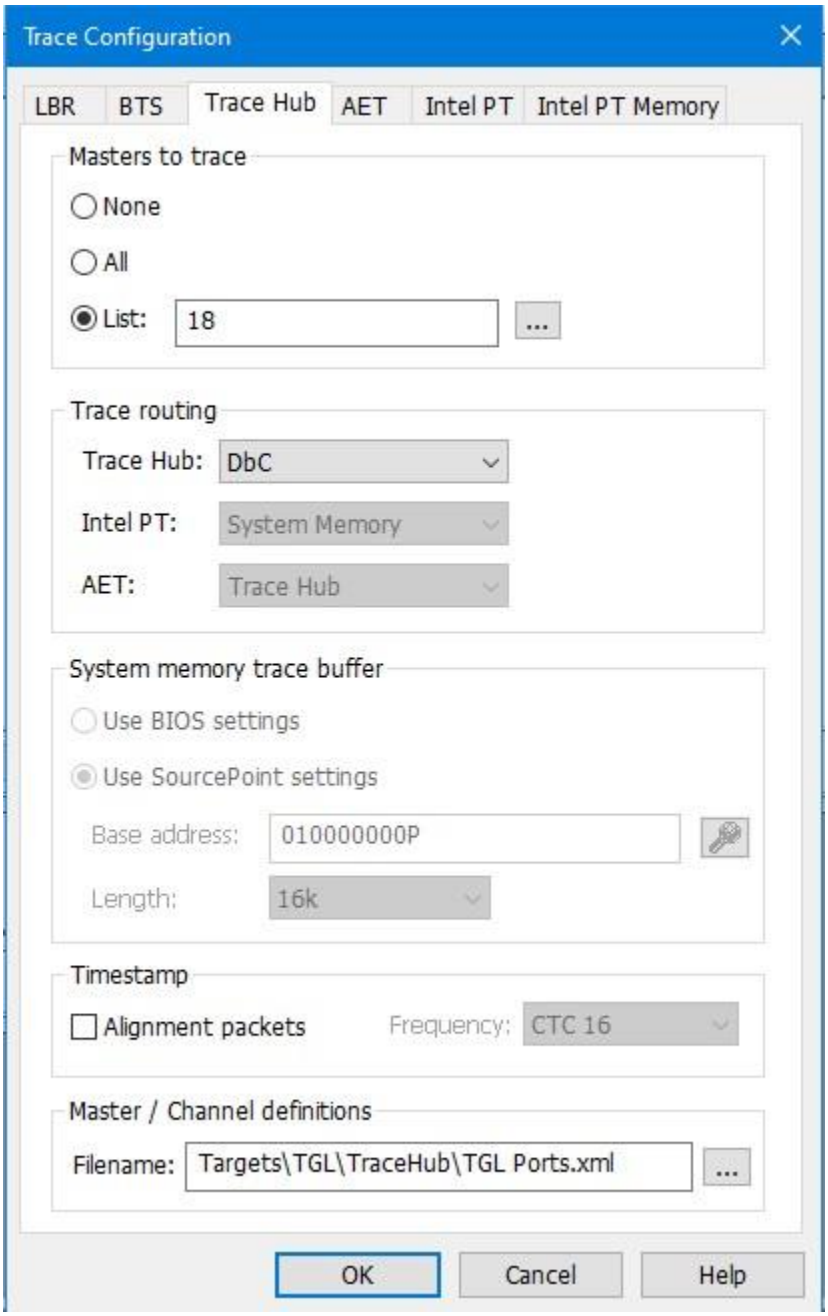
The next step is to load in the ITH macro that enable the Trace Hub and hide it from the OS. Fortunately, SourcePoint comes equipped with a macro button built in to make this

process easier, At the top of the screen, you'll see the macro button labeled `EnableTraceHub:`



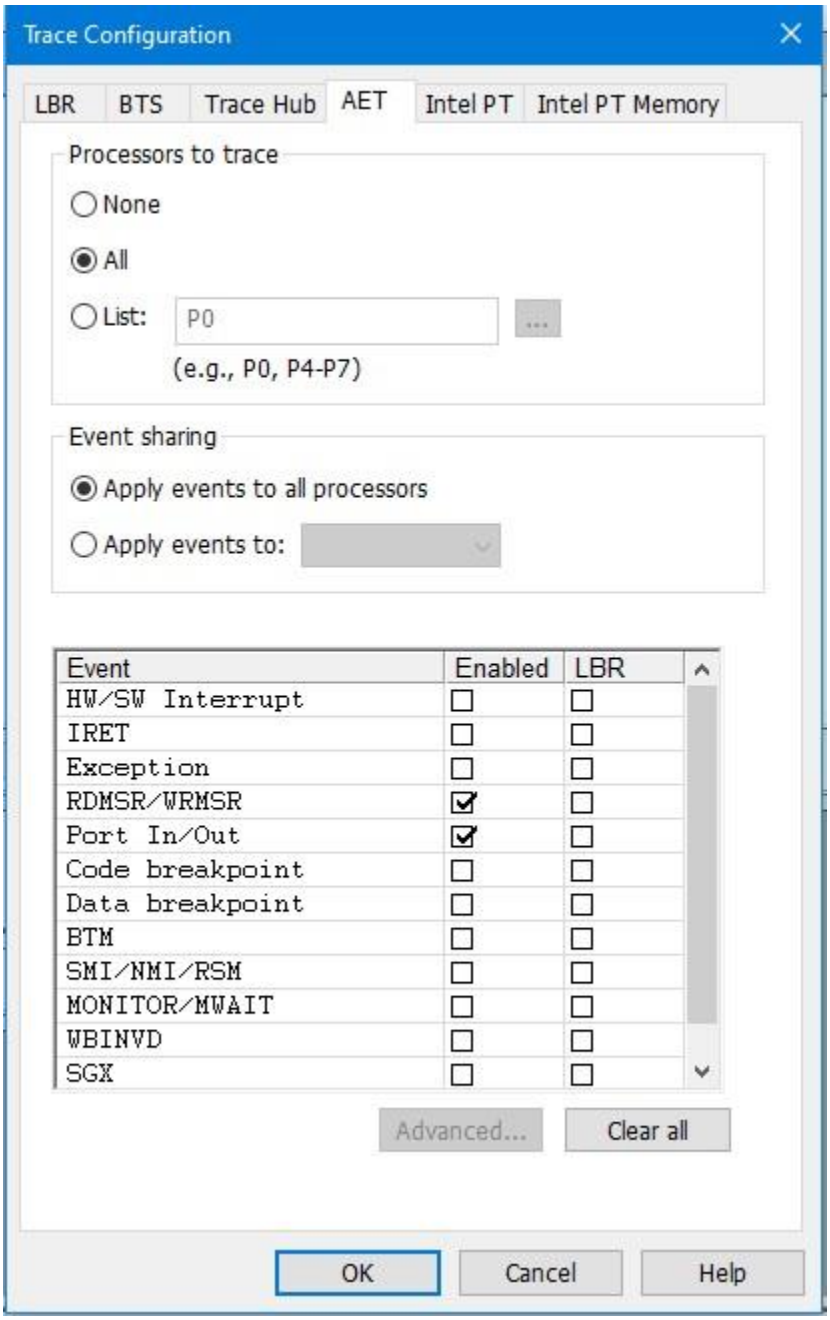
Click on it. Wait about five seconds. You've enabled the Trace Hub.

Now, it's time to set up the ITH, and let it know what you want to trace. Click on the `Trace` button in the toolbar at the top, to open the Trace window; then click on the `Configure…` button; then click on the `Trace Hub` tab. Set the settings as below:



Architectural Event Trace

Once the Trace Hub has been enabled for the features you need, click on the `AET` tab, select `All` as `Processors to trace`, and select `RDMSR/WRMSR` and `Port In/Out` as events to trace:

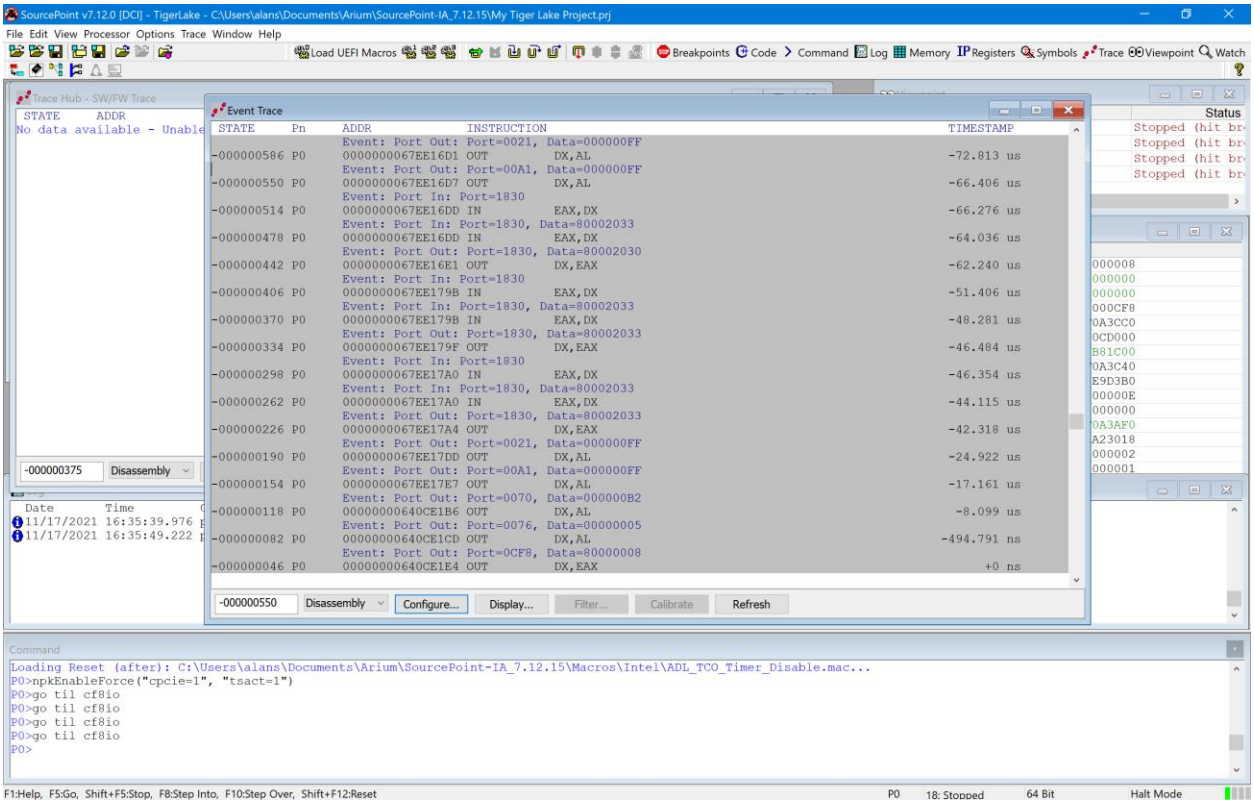


Now, you can simply do a Go/Stop to capture the event trace data. Below shows the use the `Command` window to simulate a break on any read/write of, say, port x'CF8', the PCI CONFIG_ADDRESS. This is conveniently done by issuing at the `Command` window `P0>` prompt:

```
go til cf8io
```

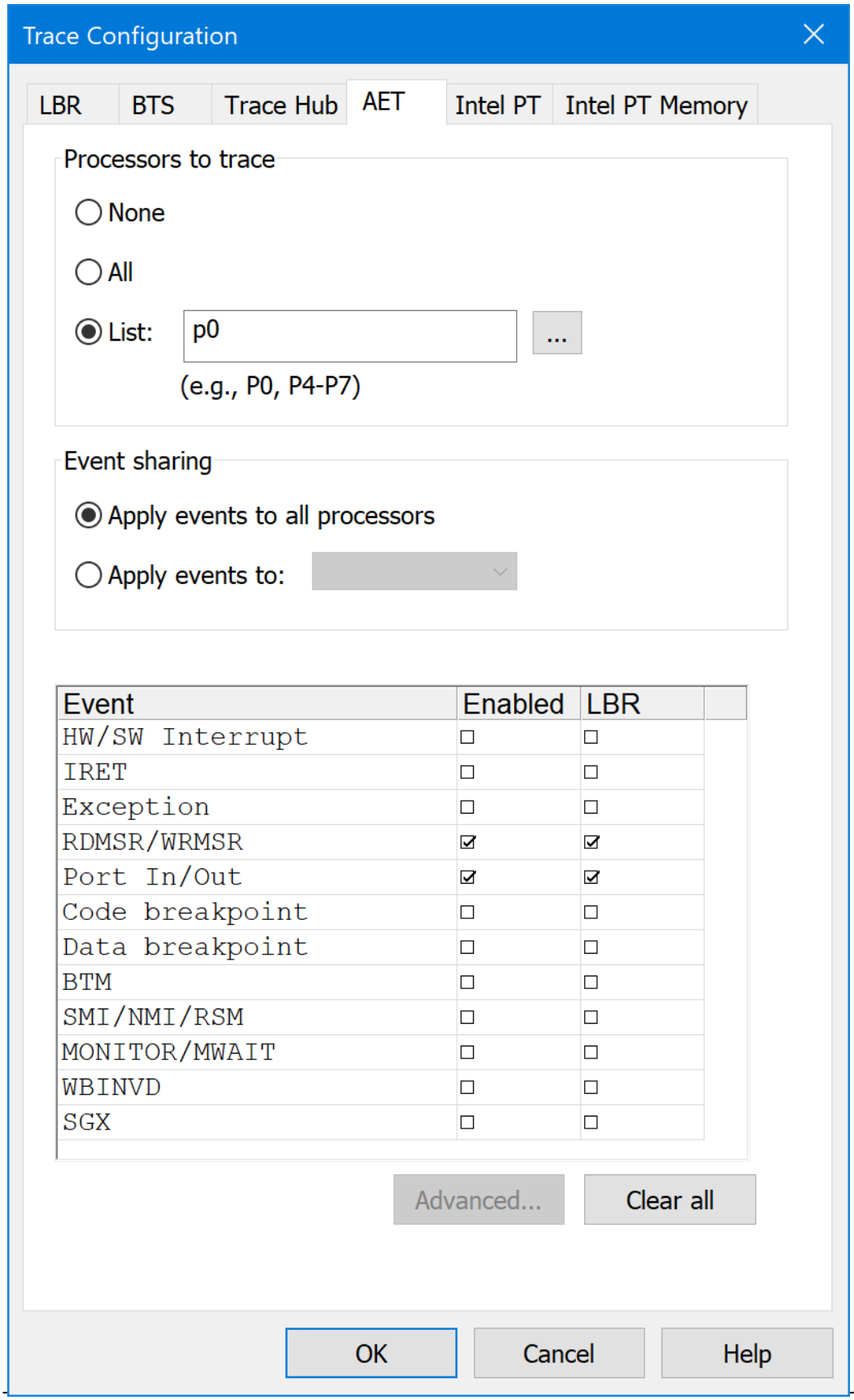
This will run the target until the next IN or OUT to CF8.

After issuing the command, you'll see something like this:



Scrolling up a little, you'll see a mix of `Port In/Out` and `RDMSR/WRMSR`, all timestamped.

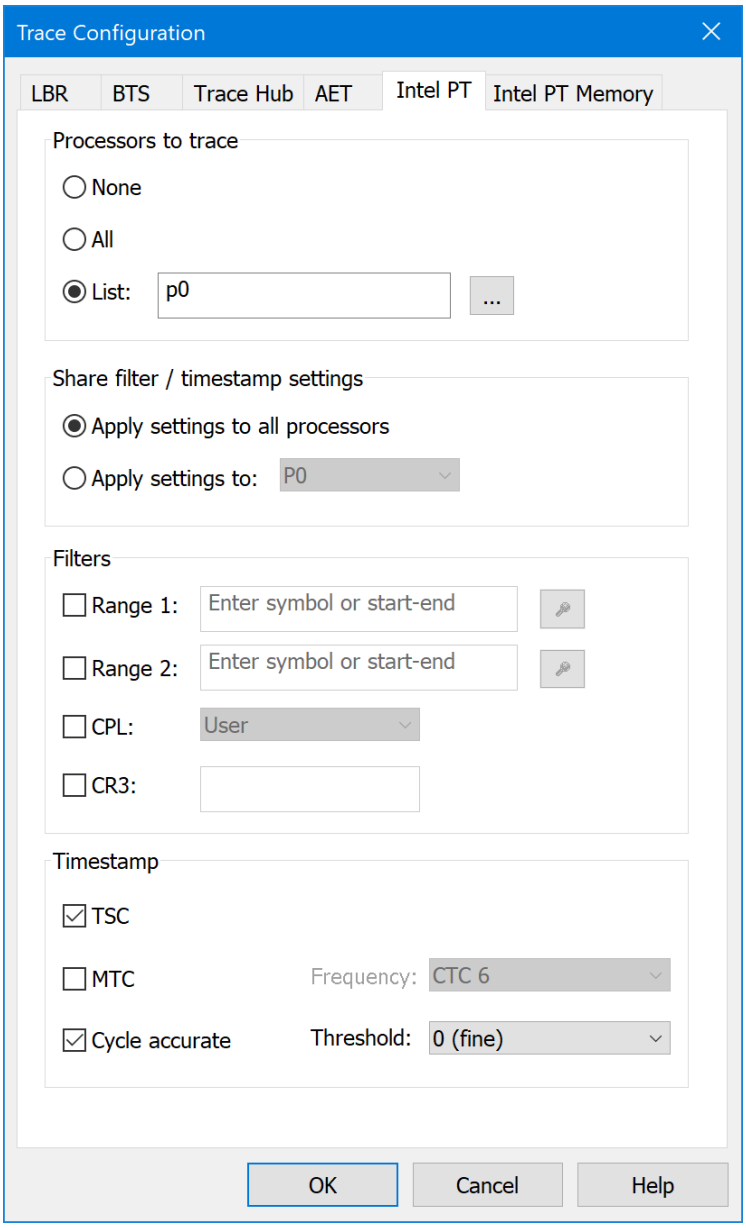
Power tip: The Last Branch Record (LBR) stack associated with each event can be captured as well. This is a very powerful debugging utility, especially when troubleshooting code execution leading up to events before system memory is initialized and Intel Processor Trace is available.


Trace Configuration
LBR
BTS
Trace Hub
AET
Intel PT
Intel PT Memory
Processors to trace
None
All
List:
p0
...
(e.g., P0, P4-P7)
Event sharing
Apply events to all processors
Apply events to:
Event
Enabled
LBR
HW/SW Interrupt
IRET
Exception
RDMSR/WRMSR
Port In/Out
Code breakpoint
Data breakpoint
BTM
SMI/NMI/RSM
MONITOR/MWAIT
WBINVD
SGX
Advanced...
Clear all
OK
Cancel
Help

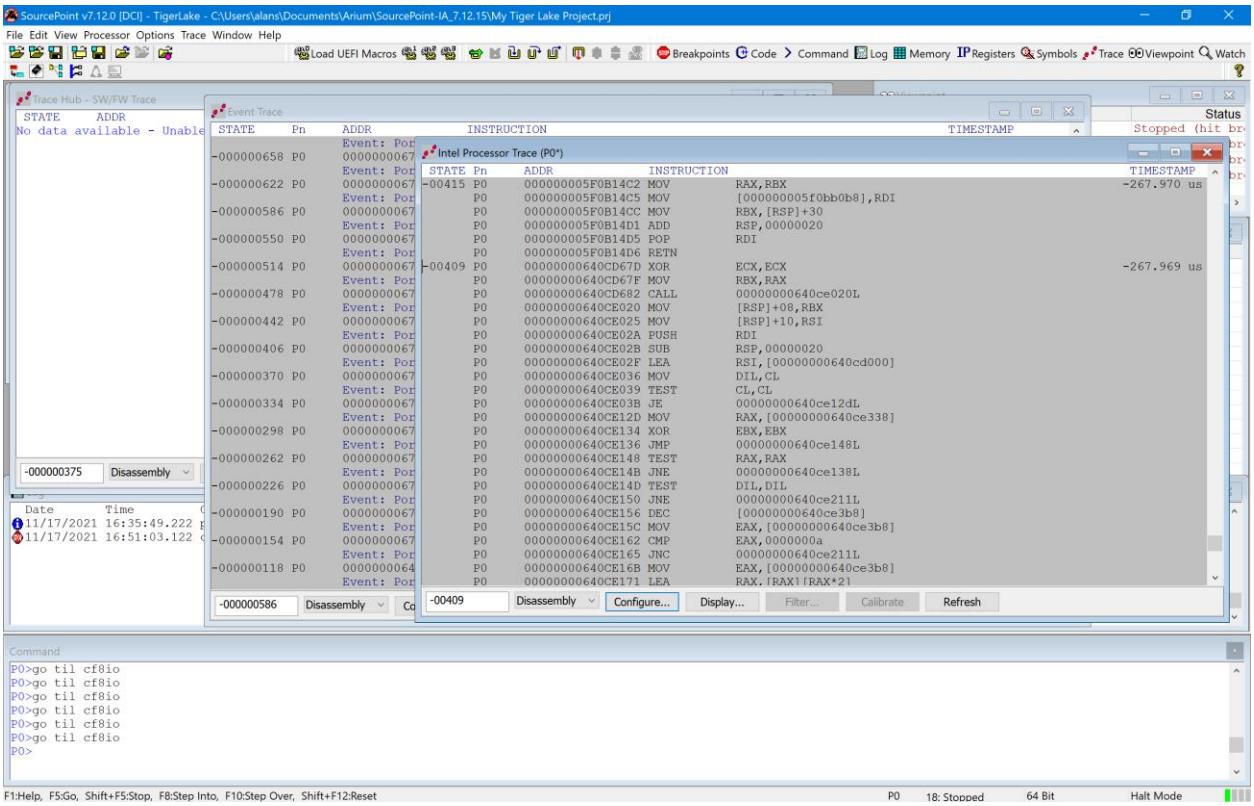
Intel Processor Trace

Intel PT is available only after system memory is initialized.

It's easy to set up. Click on the `Trace` button in the top toolbar, click the `Configure…` button, click on the `Intel PT` tab, put `p0` in `Processors to trace`, and be sure that `TSC` and `Cycle accurate` under the `Timestamp` heading are enabled:



That's all. Then use the `go til cf8io` trick to capture some instruction trace data:



There's a lot more that you can see and do with these Trace utilities. SourcePoint can use Intel PT to display a Call Chart and Call Tree. You can open up Code Tracking windows that update dynamically as you walk through the code, showing you exactly where you are and the interaction between code and events. When you have source and symbols available, the firmware flow becomes much more intuitive and visual. Indulge your curiosity and imagination.

The video at https://www.asset-intertech.com/wp-content/uploads/2021/11/UP-Xtreme-i11-Getting-Started.mp4 shows some of these capabilities. The SourcePoint User Guide also provides a very thorough, comprehensive review of the tool. And visit our SourcePoint Academy for helpful "How To" content.

Troubleshooting Tips

At some point, you'll run into something strange. We're the first to admit that JTAG-based run-control and trace are not always deterministic. JTAG is a 30-year hardware protocol, and when something goes astray at a very low level, SourcePoint tries to (but sometimes doesn't) recover gracefully. There will be times that the board will power cycle on its own. Or the firmware thinks that a thread is running but gets out of sync with the SourcePoint software, which thinks it's halted. Or the DbCStatus.exe ball stays red instead of turning green, while you swear you have a good DbC connection. Sometimes you have no choice but to quit SourcePoint and power cycle the target. That usually clears up the one-of's. But if the issue is repeatable, we ask that you collect as much information as you can, and open a ticket with us at https://www.asset-intertech.com/support/. We'll respond as soon as possible.

In the meantime, here are a few errata that we've noticed on the UP Xtreme i11, and the steps needed to mitigate.

Firmware gets out of sync with software

On the host PC using DCI, functionality is roughly partitioned between software (SourcePoint application with its GUI) and firmware (lower-level run-control primitives). Broad-brushing it, SourcePoint software on the host communicates with the firmware, that encapsulates JTAG traffic into packets which are sent to the PCH, which in turn performs the JTAG mastering function.

Somewhere along the line, the firmware may get out of sync with the software. You may see symptoms like:

In the Viewpoint window, the threads are shown as Running, whereas the Status Bar at the bottom right shows Stopped.

P0 in the Viewpoint window is Running, with one or more threads below it are in the Stopped state.

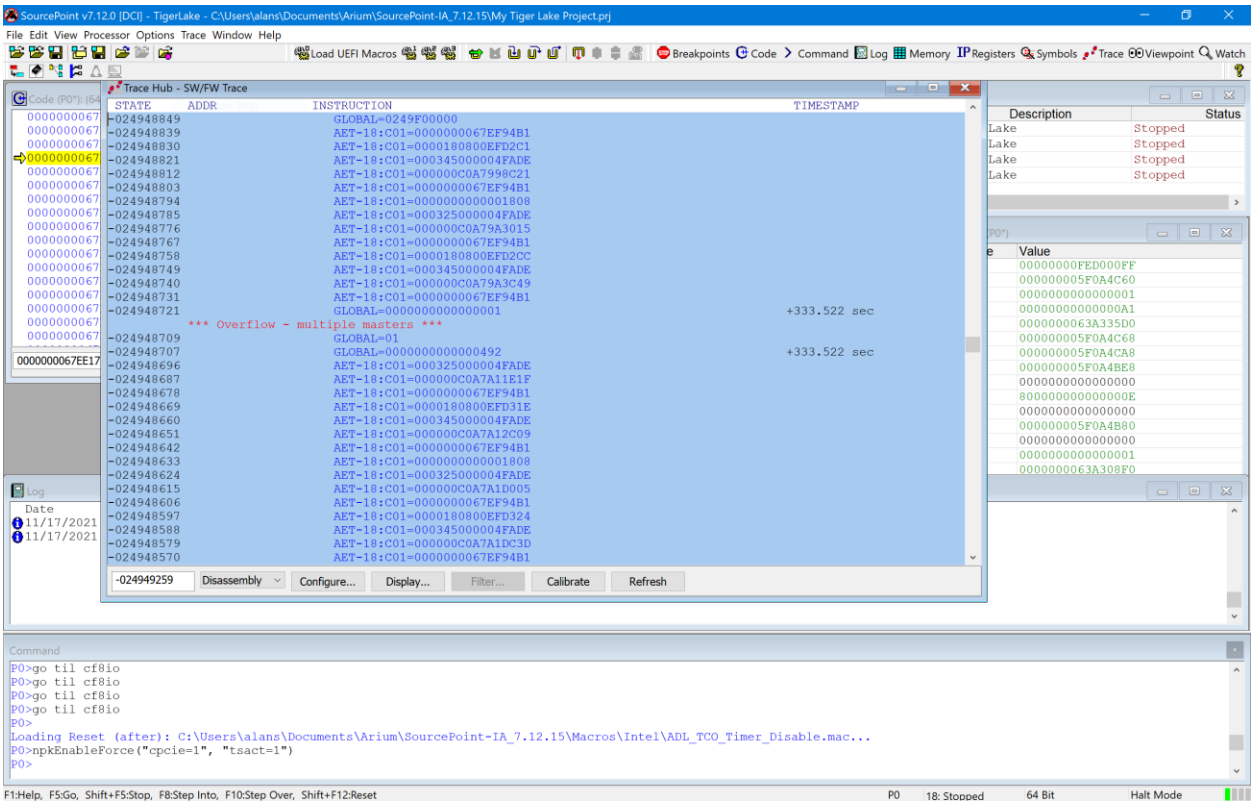
If this happens, you'll likely have to quit SourcePoint, kill the AssetDCI process (see below), power-cycle the target, then start over. Sorry. Then please enter `aalog = 20987` in the `Command` window, and try to reproduce. We're extremely interested in these cases, so capture the verbose logs in the `Log` window and send to us.

Trace buffer overflows

DCI traffic processing has its limitations. When you try to collect too much trace data, the trace buffer overflows, causing aberrant behavior.

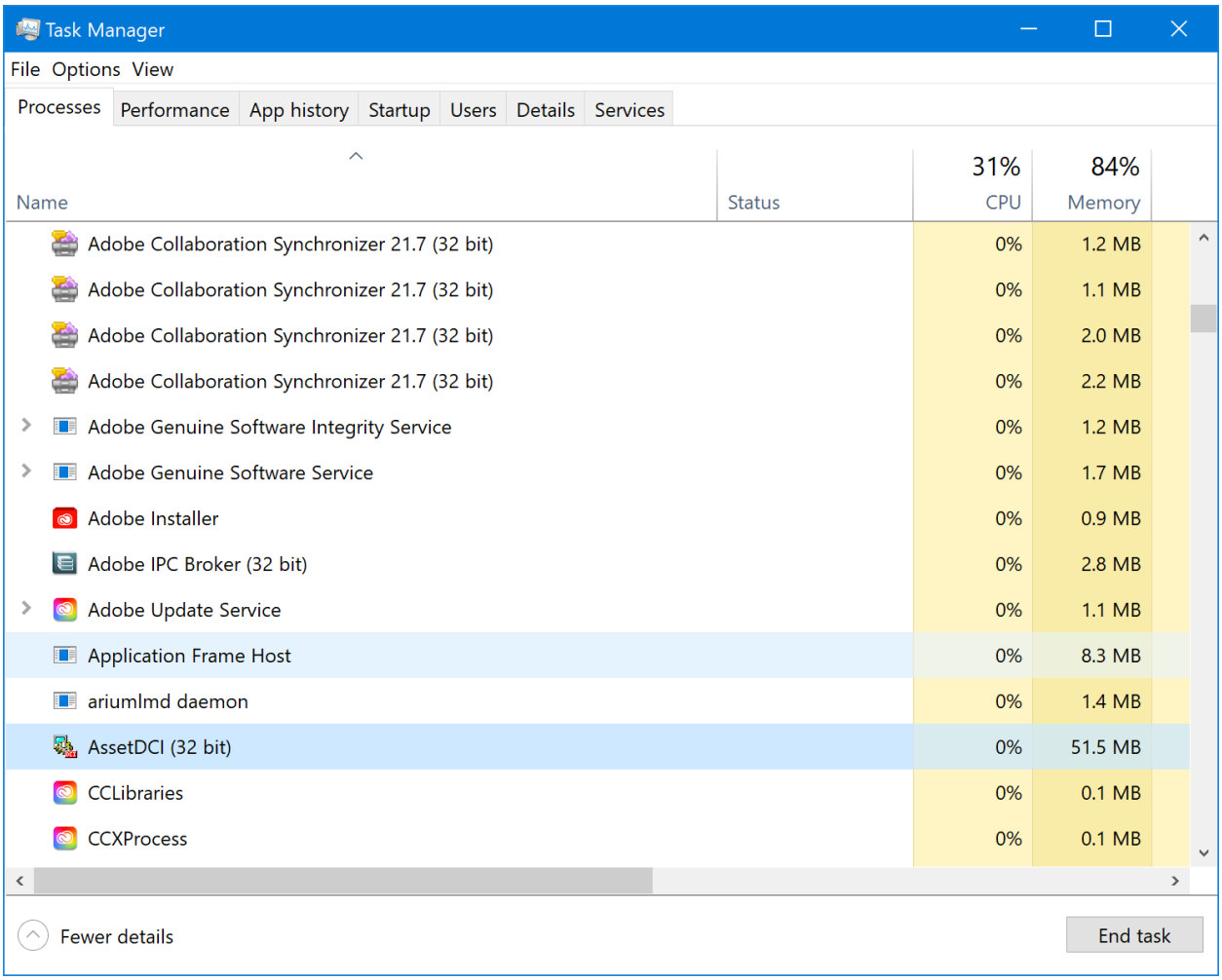
Although the trace data is highly compressed, some trace sources, in particular with AET, running through the Trace Hub can exceed the capacity of the USB 2.0 connection. In theory running at 480Mbps, in practicality SourcePoint can only process trace data at approximately 100Mbps. Beyond that, we collect ~ 20kB of trace data before a buffer in SourcePoint overflows, and we don't recovery gracefully.

You'll see these symptoms of this occurrence in the SW/FW Trace window:



A few of these overflows are no big deal. But, if you're tracing a huge amount of data, SourcePoint may spin, as it tries to process all that data, and deal with the mess. Sometimes, after maybe a few minutes, it recovers. Sometimes, you end up in limbo.

The only solution at this point is to quit SourcePoint, do an `End task` on the AssetDCI Background process, power cycle the target, and start over:



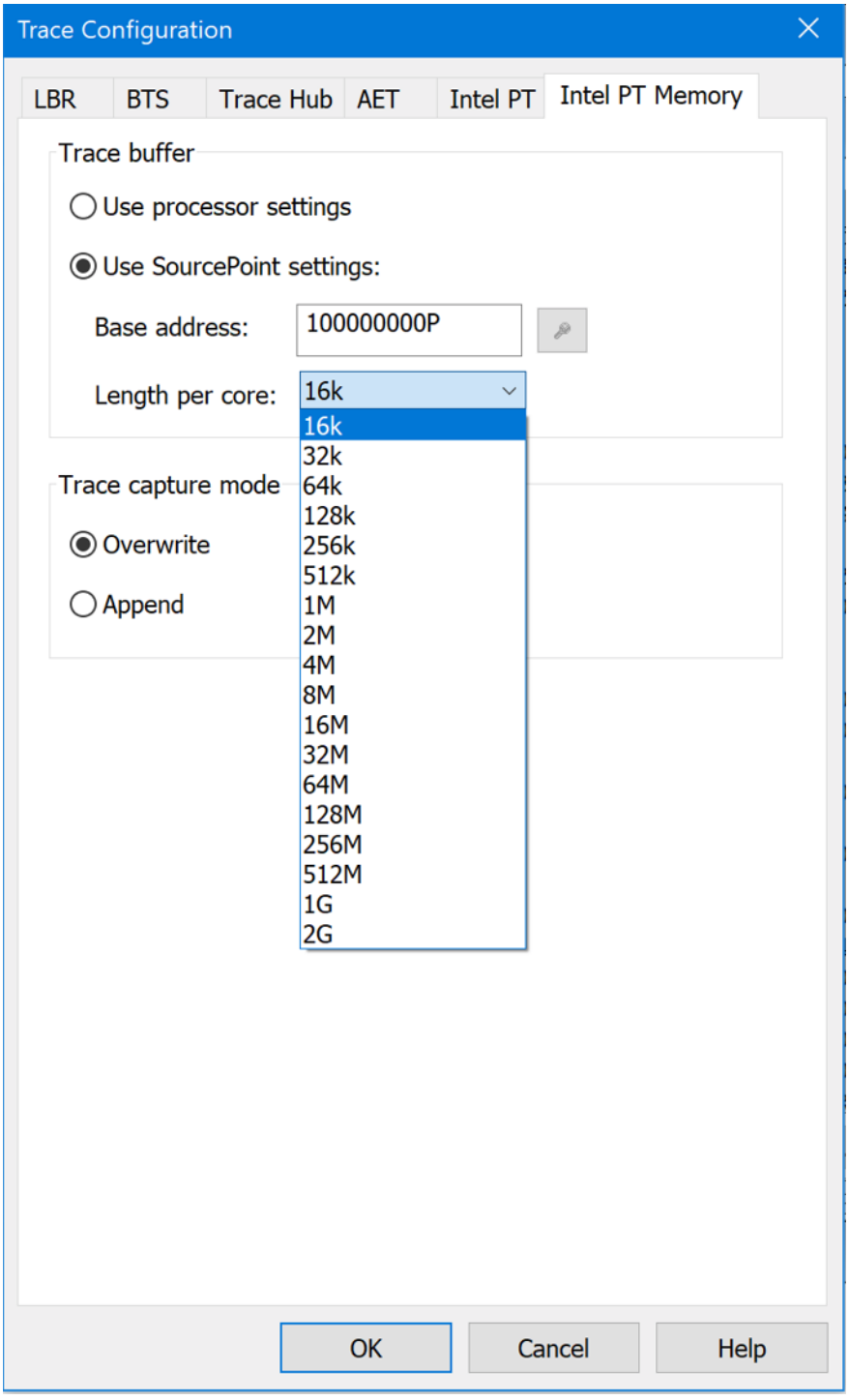
Ultimately, we're working on improving the performing of the AssetDCI driver, and behaving more graciously when overflow is encountered. But, in the interim, it is key to ensure that the trace data collected is relatively "sparse". Focus your debugging in the specific area of interest. Don't try to collect all Port IN/OUT from reset through to the UEFI shell. At 750K I/Os per second, you'll swamp the host debugger processing, and you can't deal with all that data anyway.

Note that this only happens with AET – that is, you can only overflow by collecting too much AET data. It does not apply to LBR, SW/FW Trace, SVEN, or Intel Processor Trace. It may take some trial and error to limit the scope of your event data collection.

NOTE: Release 7.12.18 and later enhanced the performance of the ASSET DCI driver. Except in very rare circumstances, trace buffer overflows should not be seen. We've preserved this section of the Getting Started Guide in case they are still encountered out in the wild – please contact our Support line directly if you see one.

Intel Processor Trace – Slow!

When you configure Intel PT, you can specify the size of the buffer in system memory to collect trace data:



Note that Intel PT directs instruction trace data to system memory. Although highly compact and efficient, we are not streaming over DCI to the host in this case; we buffer the code execution data until the platform is halted, at which point SourcePoint uses JTAG over DCI to collect the trace data out of system memory, reconstruct and display it. JTAG operates at fairly low speeds, and for large buffer sizes the transfer of all that data can be slow. It starts to become noticeable beyond a buffer size of 64kB. If you try to collect 1GB of data from system memory over JTAG, you can enjoy a cup of coffee and cake (or perhaps a full meal) while this is in process. SourcePoint will display as Not Responding, and you'll get the "grey screen" if you're too persistent. SourcePoint will eventually recover, but in the interest of time and frustration, it's probably best not to try to save huge collections of instruction trace. Focus your debug efforts in the vicinity of the bug.

My board is not booting – what now?

Once in a while, especially during an intense debug session, we have found that the target goes into la-la land. You get to the UP splash screen, and then it just stops. Or the screen stays black. SourcePoint run-control continues to work, but it won't boot all the way up to the UEFI shell. Quitting SourcePoint, unplugging the DCI cable, killing the AssetDCI process – all are good steps in this instance, when you need to recover.

But then, once in a while, you wait the needed 20 seconds; and it doesn't boot. The screen stays blank or frozen.

Don't panic! For reasons we're not sure of just yet, after about 60 seconds, the board "wakes up" and should boot all the way to the UEFI shell.

Clearing the CMOS on the target has also been known to help when it still won't boot up.

There's only one thing: it has gone back to the factory settings, so you need to reset the WDT timer, as per BIOS Settings section in this manual.

Then, you'll be back in business.

Conclusion

Thank you for getting this far! We hope that you have enjoyed the ride, and are using the power of SourcePoint successfully in your debugging and learning journeys.

Feel free to browse the SourcePoint Academy at https://www.asset-intertech.com/sourcepoint-academy/ for helpful reference guides, help material and "how to" videos.

If you ever have any questions, please call, email or open a Support Case here: https://www.asset-intertech.com/support/. We'll be glad to help!