# Real Insight from Code to Silicon

# SourcePoint<sup>®</sup> ScanWorks<sup>®</sup>

# Debug Automation using the SourcePoint Command Language

Alan Sguigna December 15, 202

© 202**2**, ASSET InterTech, Inc

# Agenda

- Introduction
- Reference Platform: AAEON Whiskey Lake w/DCI
- The Command Language
- **Getting Help**
- Macro buttons, User-Defined macros, Event macros
- The SourcePoint Assembler
- Built-in Macros: platform.mac, processor.mac, pch.mac, npk.mac
- Some helpful references
- Wrap-Up and Q&A





# SourcePoint

# A very powerful **JTAG-based** debugger Optimized for lowlevel firmware debug Learning curve – lots of features



**SourcePoint** 

*WhiskeylakeOpenBoardPkg* DbC3 & FSP – "Choppy Seas"



3



### **On AAEON Whiskey Lake (WHL)** UP Xtreme board:

**DCI: WHL DbC3** 

**Specialty USB cable** 



Scan\

# The Command Language

- A programming language very similar to 'C'
  - Conditional logic, branching, loops, typed data, operators, functions, etc. etc.
- Additional support for run-control, target access, and unique SourcePoint capabilities
  - go
  - halt
  - dbgbreak
  - step
  - fprintf
  - strcmp
  - asm/endasm
  - etc. etc.
- > 240 commands and control variables, ~ 400 pages of documentation

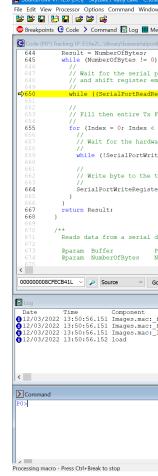


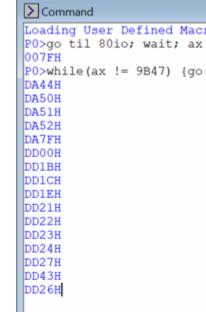
- SourcePoint Command Language
- Overview
- Introduction
- Syntax Notation
- Comments
- Constants
- Data Types
- Expressions
- Debug Variables
- Debug Variable Arrays
- Debug Procedures
- Control Variables
- Command Files
- Filenames
- Viewpoint Processor and Processor **Overrides**
- Symbolic References
- Qualified Symbol Names
- Commands and Control Variables
  - aadump
  - abort
  - abs
  - acos
  - advanced
  - asin
  - asm
  - asmmode
  - atan
  - atan2

# From the command line

### **Examples**:

go til 80io go til 80io; wait; ax while(ax != 0x9B47) {go til 80io; wait; ax}







© 2022, ASSET InterTech, Ind

5

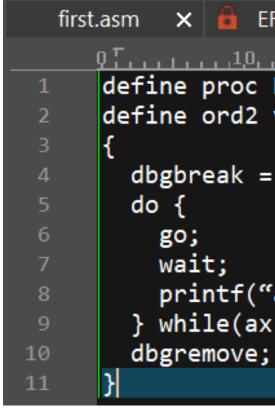
w LoadCurrent w PEIMs w DXEs pry IP Registers ⊛ Symbols ≁ Trace ⊙ Viewpoint Q Watch		dSmramSymbols 🎇 GoToShadowedPeiCore 🕯	🐮 HOBs 🦓 SysConfigTable 🖓 DumpMe	emMap 🦓 DumpCallStac	# ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
16550\baseserialportlib16550.c					
	^		Name	Description	Status
				ake / Kaby Lake ake / Kaby Lake	Stopped Not Active
t to be ready, to make sure both the transm. Y•			C P2 SkyL	ake / Kaby Lake ake / Kaby Lake	Stopped Not Active
ster (SerialRegisterBase, R_UART_LSR) & (B_1	UART_LSR_TEMT		General Registers	(P0*)	
c			⊕ IA-32	Name Value	
foSize && NumberOfBytes != 0; Index++, Numb	perOfBytes, Bu		General	RAX 00000 RBX 00000	0000A124A000 00000000003
flow control signal			-Floating Point		00000000CF8 0000A124A000
le (SerialRegisterBase));			- Segment - Control	RBP 00000	00008CF1A970
te (Seriarkegisterbase)),			Debug		00008CF1A940 00008CF1A7F0
nsmit buffer.			-MMX -YMM - SP		00008CF1A790 00000000084
(SerialRegisterBase, R_UART_TXBUF, *Buffer)	,,		-YMM - DP	R9 00000	0000000003
			⊢YMM - Int ⊕MSR		000000000000000000000000000000000000000
			User	R12 00000	00000000040
					00008CE33518 00000000000A
ice into a buffer.					00008CE37C18
nter to the data buffer to store the data re	road from the so			DS 0020	
ber of bytes to read from the serial device				SS 0020 ES 0020	
	>			FS 0020	
ursor Set Break Track IP View IP Refres	ish			GS 0020 RIP 00000	00008CFECB44
			< >>		00000010087
	ild\WhiskeylakeOpenBoard ild\WhiskeylakeOpenBoard	lPkg\UpXtreme\DEBUG_VS2019\X64\0	CoffeelakeSiliconPkg\SystemA	gent\SaInit\Dxe\Sa	InitDxe\DEBUG\SaInitDxe.
eExists FileNameString = "E:\fw2\UpX\Bui	pX\Build\WhiskeylakeOpen	3·4 · _			
eExists FileNameString = "E:\fw2\UpX\Bui eExists FileNameString = "E:\fw2\UpX\Bui dPgm Using Base	pX\Build\WhiskeylakeOpen				
eExists FileNameString = "E:\fw2\UpX\Bui eExists FileNameString = "E:\fw2\UpX\Bui dPgm Using Base	pX\Build\WhiskeylakeOpen				>
eExists FileNameString = "E:\fw2\UpX\Bui eExists FileNameString = "E:\fw2\UpX\Bui dPgm Using Base	pX\Build\WhiskeylakeOpen				>

Loading User Defined Macro #0: C:\Users\alans\Documents\Arium\SourcePoint-IA 7.12.22\Macros\EFI\EFI.mac...

P0>while(ax != 9B47) {go til 80io; wait; ax}

## In a macro

- Procedures are defined, and can be executed, within macro files
- Procedures have parameters (post code is "val") as well as global and local variables
- do/while
- dbgbreak enables a breakpoint on an I/O access, named "POST80", for port 80 and a word (16 bits)
- This routine is equivalent to:
  - while(ax != val) {go til 80io; wait; ax}





```
EFI.mac 🗙 🔍 Alan1.mac 🗙 💼 Processor.mac 🗙
define proc RunToPostCode(val)
define ord2 val
 dbgbreak = e, I/O access, n=POST80, 1=0x80, s=word;
   printf("ax = %04x n", ax);
 } while(ax != val)
```

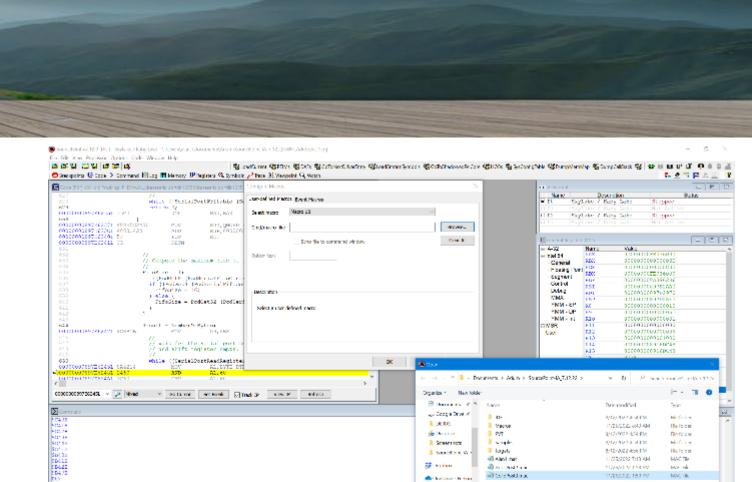


# Macros

- Can be assigned to buttons or events
- Up to 20 buttons of your own creation
- Event macros are very powerful, and can be triggered based upon:
  - Breakpoint (any)
  - **Emulator connected**
  - Go
  - **Project Load/Unload**
  - Reset (before, after)
  - Stop
  - Target configure
  - Target power on
  - etc.



7



Select event:	Breakpoint (any)	$\sim$	
Cmd/macro file:			Browse
	Echo file to command window		Clear All
	Disable macro		
Description			
Description			
	a affect and least to bit		
This event run	s after any breakpoint is hit.		
This event run	s after any breakpoint is nit.		
This event run	s after any breakpoint is nit.		

📽 LoadCurrent 🏶 PEIMs 🏶 DXEs 🖓 GoToNextDriverEntry 🖏 LoadSmramSymbols 🦓 GoToShadowedPeiCore 🎕 HOBs 🖏 SysConfigTable 🖓 DumpMemMap 🦓 DumpCallStack 🦓

ATC: N 30 Object

Desite

the number of Golford met

Com

# **The SourcePoint Assembler**

The command language provides access to the SourcePoint assembler

Can use the command line, or within macros

Single-pass assembler; cannot use labels not yet defined (but can use org directive for out-of-order assembly, and use command language for labels)

Example:

- asm 1000 =
- mov al, 01
- mov al, 02
- mov al, 03
- endasm

Edit View Processor Options Comman	d Window Help							··· · · · · · · · · · · · · · · · · ·			
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8						ToNextDriverEntry	LoadSmramSymbols 🎇 GoToShadowedP	eiCore 🍓 HOBs 🍓 SysConfigTable 🍓 DumpN	/lemMap 🦓 Dump(		UP 🔟 🗇 🖷
reakpoints 🕒 Code 💙 Command 🔝 l	og 🎹 Memory IP	Registers 🤹 Sy	mbols 🧨 Trace	OO Viewpoint C	Watch					4	🕐 👎 🛱 🗛
Code (P0*): (64-bit) Tracking IP 00000000	000000L - FFFFFFFF	FFFFFFEL				- 0 ×		00 Viewpoint			
000000000006FAL 0000	ADD		TR [RAX],AL			,		Name	Description		Status
0000000000006FCL 0000 000000000006FEL 0000	ADD ADD		TR [RAX],AL						/Lake / Kaby L		
0000000000000700L 48B80100000			000000000000000000000000000000000000000						/Lake / Kaby La /Lake / Kaby La		
00000000000070AL 48FFC0	INC	RAX							Lake / Kaby L		
00000000000070DL EBFB 00000000000070FL 0000	JMP ADD		otr 0000000 TR [RAX],AL								
0000000000000711L 0000	ADD		TR [RAX],AL				Window Spin	TT a start a	(5.0.1)		
00000000000713L 0000	ADD		TR [RAX],AL					IP General Registe			
00000000000715L 0000	ADD		TR [RAX],AL					⊞ IA-32		Value	1
000000000000717L 0000 00000000000719L 0000	ADD ADD		TR [RAX],AL					E Intel 64		000000000000000000000000000000000000000	
00000000000071BL 0000	ADD	BYTE PI	TR [RAX],AL					- General	DCV	000000000000000000000000000000000000000	
0000000000071DL 0000	ADD		TR [RAX],AL					Floating Poir Segment		00000000FE03600	
00000000000071FL 0000 00000000000721L 0000	ADD ADD		TR [RAX],AL					Control		0000000088F10B5	
000000000000723L 0000	ADD		TR [RAX],AL TR [RAX],AL					Debug		0000000897D2C4	
000000000000725L 0000	ADD		TR [RAX],AL					- MMX		0000000897D2B3	
00000000000727L 0000	ADD	BYTE PI	TR [RAX],AL					YMM - SP		0000000897D2A9	
00000000000729L 0000	ADD		TR [RAX],AL					YMM - DP		000000000000000000000000000000000000000	
00000000000072BL 0000 0000000000072DL 0000	ADD ADD		TR [RAX],AL TR [RAX],AL					YMM - Int		000000000000000000000000000000000000000	
00000000000072FL 0000	ADD		TR [RAX],AL					MSR		000000000000000000	
00000000000731L 0000	ADD	BYTE PI	TR [RAX],AL					User	R12	000000088F1138	0
00000000000733L 0000	ADD		TR [RAX],AL							000000000000004	
000000000000735L 0000 00000000000737L 0000	ADD ADD		TR [RAX],AL TR [RAX],AL							0000000897D2D7	
000000000000739L 0000	ADD		TR [RAX],AL							000000000000000	4
0000000000073BL 0000	ADD	BYTE PI	TR [RAX],AL							0038	
0000000000073DL 0000	ADD		TR [RAX],AL							0030	
0000000000073FL 0000 00000000000741L 0000	ADD ADD		TR [RAX],AL TR [RAX],AL							0030	
000000000000743L 0000	ADD		TR [RAX],AL						FS	0030	
00000000000745L 0000	ADD	BYTE PI	TR [RAX],AL							0030	
000000000000747L 0000	ADD		TR [RAX],AL							00000000000070	
	חחב		<u>דב וצבקו קי</u>							00000000000024	6
000000000070AL V	Go Cursor	Set Break	✓ Track IP	View IP	Refresh			< >>			
ommand							1				
asm 700=											
00000000000000700L>mov rax,1											
00000000000000000000000000000000000000											
00000000000070FL>endasm											
rip=7001											
step											



ScanWorks®

# **Useful macros within SourcePoint**

- TraceHub.mac
- Platform.mac
- Processor.mac
- Example:
  - define nstring strpath =
     defaultpath
  - defaultpath = strpath +
     \\Macros\\Intel
  - include platform.mac
  - pfmGetDetails()
  - show



### > Command

P0>pfmGetDetails()	
Macro Versions: Platform v1.04, Processo	or
Target Platform is Whiskey Lake	
CPU: Whiskey Lake U-Processor Line (	(Mo
IGD: GT1	
PCH: CannonLake-LP Premium U, Steppi	ng
P0>	-
1	

F1:Help, F5:Go, Shift+F5:Stop, F8:Step Into, F10:Step Over, Shift+F12:Reset

v1.03, PCH v1.03 obile) Dual Core, Stepping V0, Processor ID: 0x806EC g D0





© 2022, ASSET InterTech, Inc.



# **Good Resources**

- **SourcePoint Academy**: https://www.asset-intertech.com/resources/academy/sourcepoint-academy/ "How To" Guides and all our technical documentation online
  - Blogs: https://www.asset-intertech.com/resources/blog/category/arium-probes-sourcepoint/ Tons of articles on timely debugging and JTAG topics
  - Videos: https://www.asset-intertech.com/resources/videos/ Our recorded webinars: <u>DCI debug of UEFI and hypervisor technologies on the AAEON UP Whiskey</u> Lake and Tiger Lake boards, <u>JTAG-based debugging of AMD EPYC servers</u>, etc.
- Webinars: https://www.youtube.com/c/UEFIForum/videos Beyond Printf – Real-Time Firmware Debugging **UEFI Debug with Intel Architectural Event Trace** JTAG-based Debug & Trace







# Wrap-Up

## Questions?

# Reach me at <u>alan.sguigna@asset-intertech.com</u>, DM @AlanSguigna





# Real Insight from Code to Silicon



© 2022, ASSET InterTech, Inc.

