

Flash Technology

- ◆ Tools For Intel IA32 , IA64
And Core 2 Solo/Duo
- ◆ Agenda
 - About Flash Technology
 - American Arium Backgroud and Products
 - SourcePoint Key Features
 - EFI Framework Support
 - Conclusions
- ◆ Demonstration of Legacy BIOS Debugging
Q & A



FLASH TECHNOLOGY

One Stop Tools Solution

2009年6月12日

Who we are

- ◆ We are **distributors** providing the most advanced **tools and dedicated software** used in the field of electronics products **Development / Production / Testing** stage
- ◆ We are founded in **Singapore** at 1994 and have branch offices at **Hong Kong** and **Shanghai**
Totally about 25 professional employee
- ◆ Our business area: South East Asia (Singapore, Malaysia ,Thailand, Indonesia ...), HK, China, TW
- ◆ Systematic understanding of local market with global view



FLASH TECHNOLOGY

One Stop Tools Solution

Our Products – Development tools

- ◆ C/C++ **Compiler** and **IDE** (Integrated Development Environment)
JTAG Debugger and **In Circuit Emulator**
Dedicated **Software** solutions such as **Embedded GUI**, **x86 Firmware/BIOS**, **RTOS** etc
- ◆ Supported **MCUs**:
Almost all MCUs from 8 bits to 64 bits
Such as 8051,ARM,PowerPC,MIPS,x86 and many others from various semiconductors
- ◆ Targeted and Dedicated **Industries**:
Consumer, Telecom, Auto, x86 and many others...



FLASH TECHNOLOGY

One Stop Tools Solution

Our Products – Development tools (2)

- ◆ Protocol Analyzer / Traffic Generator
- ◆ Bit Error Rate Tester (BERT)
- ◆ Supported and Dedicated Industry Protocols:
 - USB(1.1 2.0 3.0)
 - Wireless USB/ UWB
 - Bluetooth
 - Serial ATA / SAS
 - Fiber Channel
 - Ethernet / TCP/ IP (up to 10G bps)
 - I2C / SMBus / SPI / IPMI
 - CAN, CAN Open



FLASH TECHNOLOGY

One Stop Tools Solution

Our Products – Development tools (3)

- ◆ Embedded BIOS
- ◆ Embedded GUI
- ◆ Embedded Linux
- ◆ RTOS
- ◆ Targeted Platforms and Applications
 - x86 (Intel AMD VIA)
 - ARM
 - Power PC
 - MIPS

And many more ...



FLASH TECHNOLOGY

One Stop Tools Solution

Our Products – Production/Testing

- ◆ Universal Programmer
- ◆ Gang Programmer
- ◆ In System Programmer
- ◆ Boundary Scan In Circuit Tester
- ◆ Protocol Tester for
USB, PCMCIA, 1394, CAN, CAN open ...



FLASH TECHNOLOGY

One Stop Tools Solution

Our Products – Smart Card Industry

- ◆ Dedicated Smart Card MCU Emulator
- ◆ Smart Card Analyzer and Simulator
- ◆ Smart Card Protocol Analyzer and Validation
- ◆ Targeted Applications and Protocol
 - Contact and Contactless Smart Card
 - VISA
 - Master Card
 - EMV
 - China UnionPay
 - GSM
 - CDMA



FLASH TECHNOLOGY

One Stop Tools Solution

Our Service

- ◆ Customer Training
- ◆ Boundary Scan Test Partition Generation
- ◆ Software NRE
Including BIOS NRE, Embedded GUI and more
- ◆ Consultant (R&D->MP->Test total tools solution)



FLASH TECHNOLOGY

One Stop Tools Solution

Our Products for x86 Industry

- ◆ BIOS/Firmware
- ◆ In-Circuit Emulator for Intel and AMD
- ◆ Protocol Analyzer and BERT
- ◆ Boundary Scan ICT and ISP
- ◆ Single/Gang SPI Flash Programmer
- ◆ SPI Flash Emulator
- ◆ I2C/SPI/ IPMI Analyzer/Adapter
- ◆ LPC POST Card with POST code logging functionality
- ◆ Cardbus/1394/USB Production Tester



FLASH TECHNOLOGY

One Stop Tools Solution

BIOS/Firmware



- ◆ General Software Embedded BIOS with StrongFrame Technology (Now acquired by Phoenix Technology)
- ◆ Deep Support for Intel, AMD, VIA, Transmeta, Broadcom/Serverworks, STPC, Acer Labs ...
- ◆ Both Legacy and UEFI solutions

*Embedded***BIOS**[®]
with StrongFrame[™] Technology

- ◆ <http://www.gensw.com>
<http://www.embeddedbios.com>



FLASH TECHNOLOGY

One Stop Tools Solution

In-Circuit Emulators for Intel and AMD

- ♦ American Arium ITP/XDP for Intel
- ♦ FS2 System Navigator for AMD Geode



<http://www.arium.com>

<http://www.fs2.com>

<http://www.tianocore.org> -> useful links



FLASH TECHNOLOGY

One Stop Tools Solution

Protocol Analyzer and BERT

- ◆ Protocol Analyzer / Bit Error Rate Tester/ Traffic Generator FOR
USB, USB 3.0, Wireless USB, PCI, PCI Express, SAS, SATA, Ethernet/IP, Fiber Channel ...



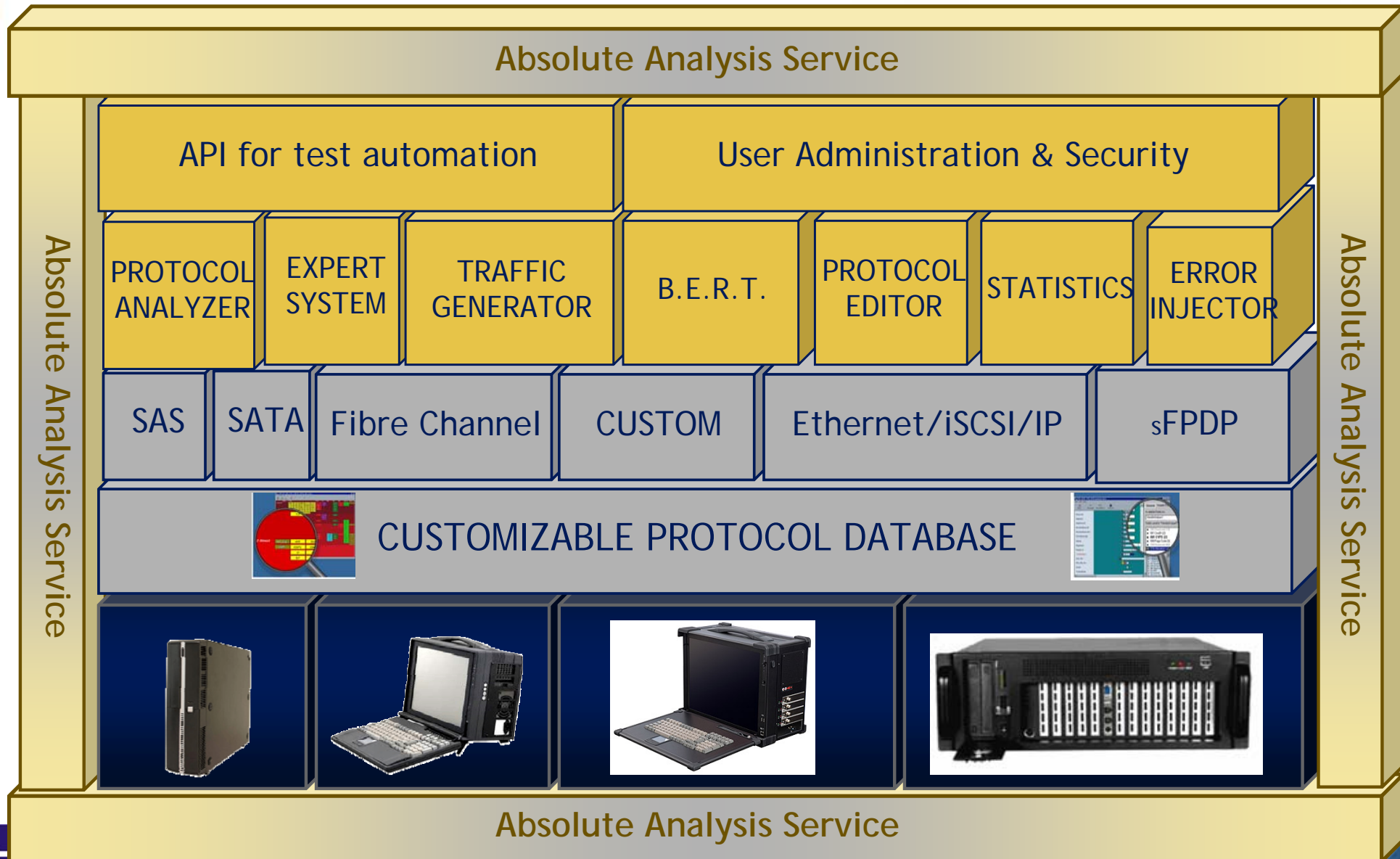
<http://www.absoluteanalysis.com> <http://www.ellisys.com>



FLASH TECHNOLOGY

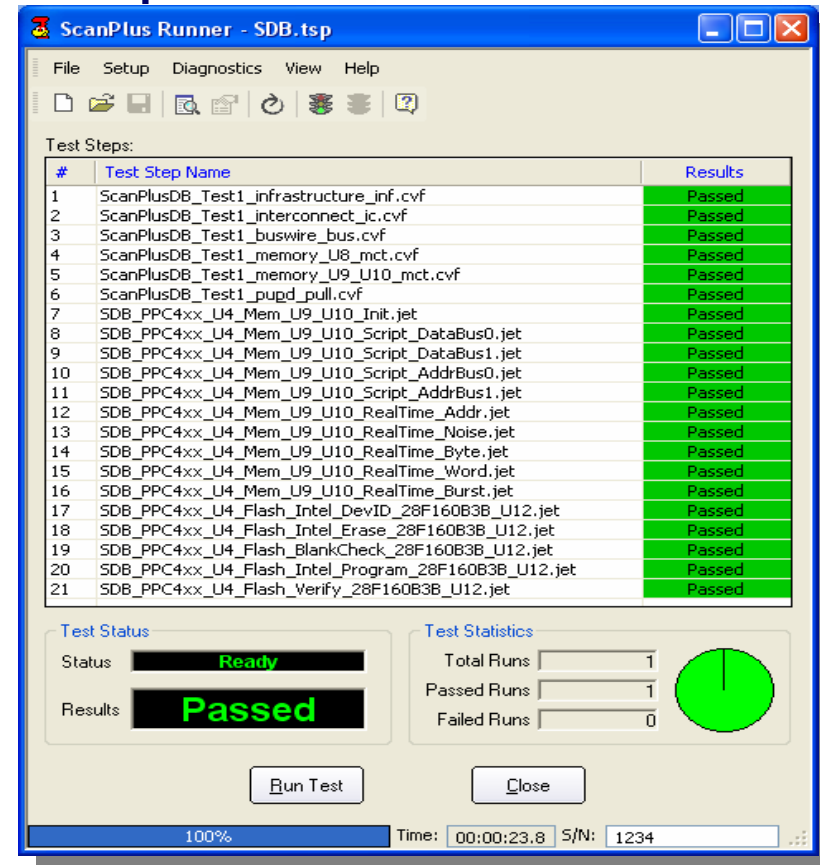
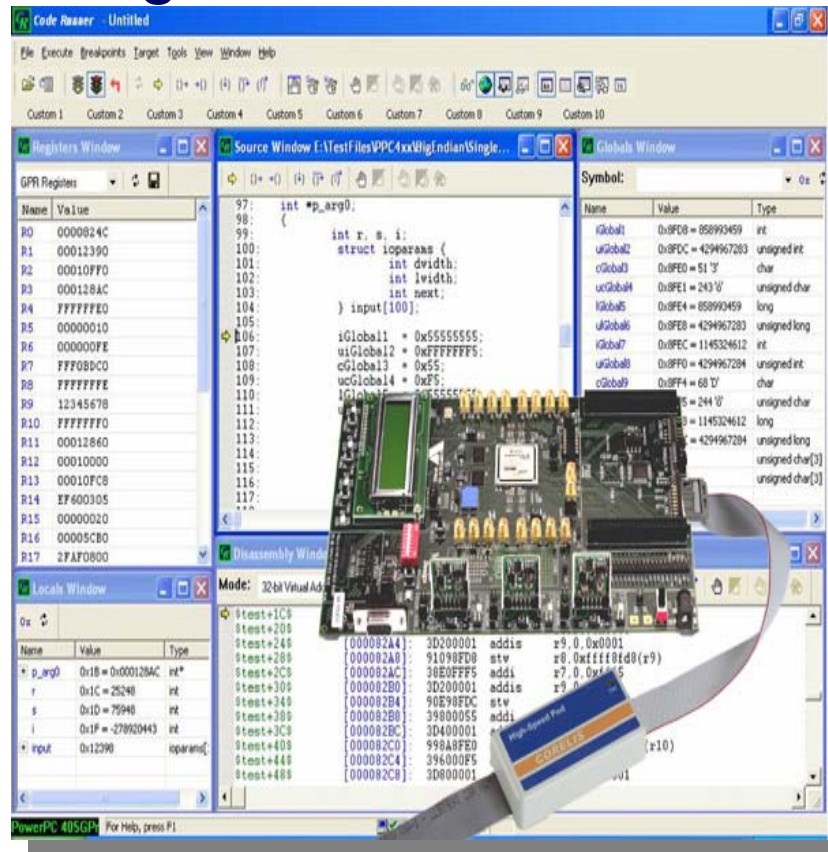
One Stop Tools Solution

Protocol Analyzer and BERT- more



Boundary Scan ICT/ISP

- ◆ Boundary Scan In-Circuit Tester and In-System Programmer from Corelis <http://www.corelis.com>

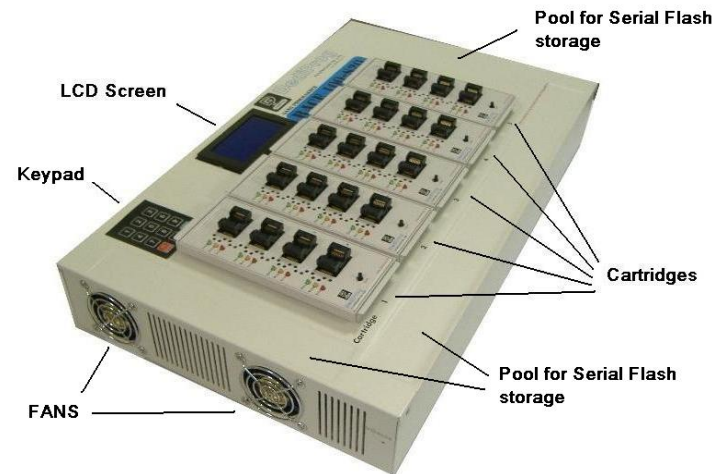
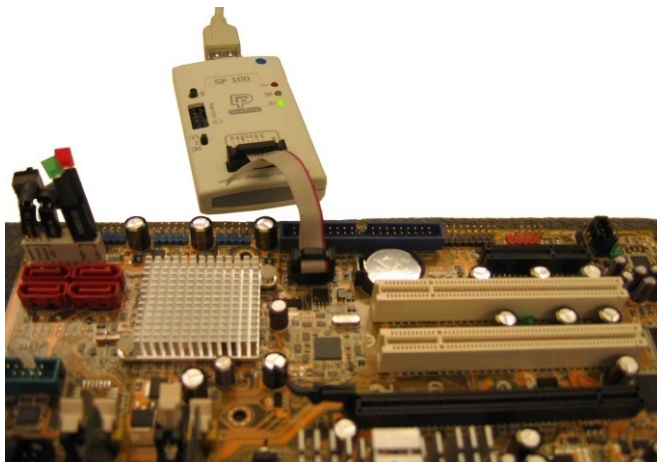


FLASH TECHNOLOGY

One Stop Tools Solution

SPI Programmer

- ◆ Stand Alone SPI Flash Programmer
- ◆ In System SPI Flash Programmer
- ◆ Backup BIOS Flash Module
- ◆ Gang Programmer



- ◆ <http://www.dediprog.com>



FLASH TECHNOLOGY

One Stop Tools Solution

SPI Flash Emulator

- ◆ High speed SPI Flash Emulator
- ◆ Display, Edit code in real time
- ◆ Less than 2 seconds for code download whatever densities
- ◆ Emulate up to two SPI Flash



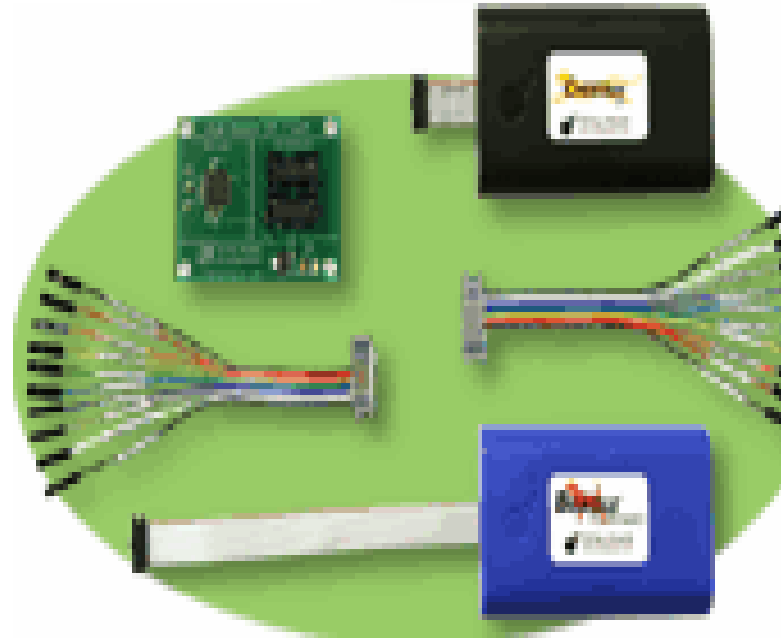
- ◆ [Http://www.dediprog.com](http://www.dediprog.com)



FLASH TECHNOLOGY

One Stop Tools Solution

I2C/SPI/ IPMI Analyzer / Adapter



FLASH TECHNOLOGY

One Stop Tools Solution

LPC POST Card with POST code logging functionality

- ◆ POST code (Port 80h) capture through LPC interface
- ◆ Real time 7-Segment LED display
- ◆ POST Code Logger
(send all post codes to host PC through RS-232 as hexadecimal bytes in ASCII so that you can understand the full and detailed POST flow)
- ◆ POST code playback, manually replay the POST flow if there isn't a RS232 connection to host PC
- ◆ Can be used in BIOS/Firmware development stage and product burn in process, and also can work in stand alone mode without a HOST PC



FLASH TECHNOLOGY

One Stop Tools Solution

Cardbus/1394/USB Production Tester



FLASH TECHNOLOGY

One Stop Tools Solution

Our Reference Customers

- ◆ Consumer Industry
Motorola Samsung Haier TCL .etc
- ◆ Automotive Industry
BOSCH Delphi Infineon Freescale SAIC .etc
- ◆ IT Industry
Intel Dell IBM HP Lenovo Seagate .etc
- ◆ Telecom
Alcatel-Lucent Cisco Huawei ZTE .etc
- ◆ OEM/EMS
Foxconn Flextronics Inventec Jabil Compal .etc
- ◆ Others
China Telecom, China Unionpay, China Mobile and
lots of universities and research institutes



FLASH TECHNOLOGY

One Stop Tools Solution

Our Offices

- ♦ **Flash Technology Pte Ltd (HQ)**
No 61 Kaki Bukit Avenue 1 #04-15
Shun Li Industrial Park **SINGAPORE** 417943
Tel: (65) 6749 6168 Fax: (65) 6749 6138
- ♦ **Flash Technology (HK) Ltd**
Room 02, 13/F, Times Tower, 928-930 Cheung Sha Wan Road, Cheung Sha Wan, Kowloon, **Hong Kong**
Tel: (852)23109662 Fax: (852)28157209
- ♦ **Flash Technology Trading (Shanghai) Ltd**
Room 405 No 1 Lane 600 TianShan Road Tong Da Chuang Ye Building Shanghai **CHINA** 200051
Tel: (86)21 61457130 Fax: (86)21 61457131



FLASH TECHNOLOGY

One Stop Tools Solution

American Arium Background



- ◆ Privately held, founded in 1978
- ◆ Supplies emulators and debuggers to embedded software and hardware developers
- ◆ Started Intel® Pentium® processor ITP run-control and FSB trace offering in 1992
- ◆ Worldwide market leader for Intel Architecture microprocessor emulation
- ◆ Started ARM® offering in 2001, XScale™ in 2002
- ◆ Key industry partner relationships
 - ARM, Atmel, Altera, Cogent, Intel, Microcross, MontaVista, Motorola, NetSilicon, SlickEdit

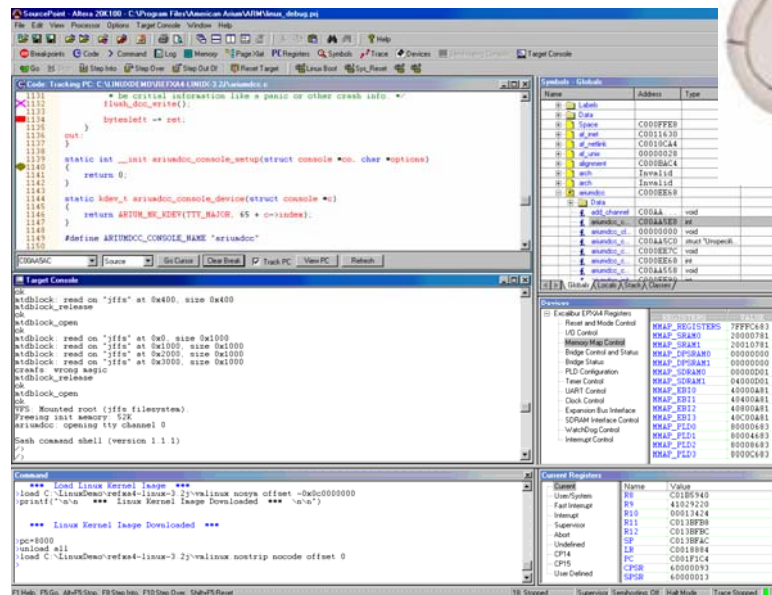


FLASH TECHNOLOGY

One Stop Tools Solution

Arium Product Lines

- ♦ ARM Debug Tools
 - SourcePoint IDE
 - SourcePoint Debugger
 - GNU-X Compiler
 - SourcePoint Co-Design
 - JTAG Emulators
 - Development Boards
 - Simulator
- ♦ Intel IA-32/IA-64 Debug Tools
 - SourcePoint Debugger
 - In-Target Probes (ITP)
 - Interposers



FLASH TECHNOLOGY

One Stop Tools Solution

ECM-700 (Intel ITP700)

- ◆ Supported Processors:
 - Intel IA32/IA64 processors with ITP700 debug interface, those include:
Pentium II/III/
Celeron / Celeron D / Celeron M
Pentium 4/ Pentium M
Xeon™/
Itanium/ Itanium® 2
- ◆ All processor speeds
- ◆ Hyper-thread support
- ◆ Multi-processor debug, up to four
- ◆ Multi-cluster debug
- ◆ Instruction trace for Pentium 4,Xeon
- ◆ Framework C debug
- ◆ USB and Ethernet interfaces
- ◆ SourcePoint™ debugger and IDE



FLASH TECHNOLOGY

One Stop Tools Solution

ECM-XDP3 (Intel eXtend Debug Port)

- ♦ **Supported Processors**
Intel Core Solo/Core Duo/Core 2 Duo processors with XDP debug interface, those include:
Conroe, Yonah, Morem
Sossaman, Woodcrest
Dempsey, Clovertown
Penryn, Gainestown and more
- ♦ **All processor speeds**
- ♦ **Hyper-thread support**
- ♦ **Multi-processor debug, up to four**
- ♦ **Multi-cluster debug**
- ♦ **Instruction trace**
- ♦ **Framework C debug**
- ♦ **USB and Ethernet interfaces**
- ♦ **SourcePoint™ debugger and IDE**



FLASH TECHNOLOGY

One Stop Tools Solution

Interposers

- ◆ TAP-PSC (used with ECM-700)
Interposer for Intel Pentium 4
with 478pin PGA
- ◆ TAP-BN (used with ECM-700)
Interposer for Intel Pentium M
with 478pin PGA
- ◆ TAP-YN (used with ECM-XDP)
Interposer for Intel Socket M
Core Solo/ Core 2 Duo processor
- ◆ TAP-MB (used with ECM-XDP)
Interposer for Intel Socket P
Core Duo/ Core 2 Duo processor

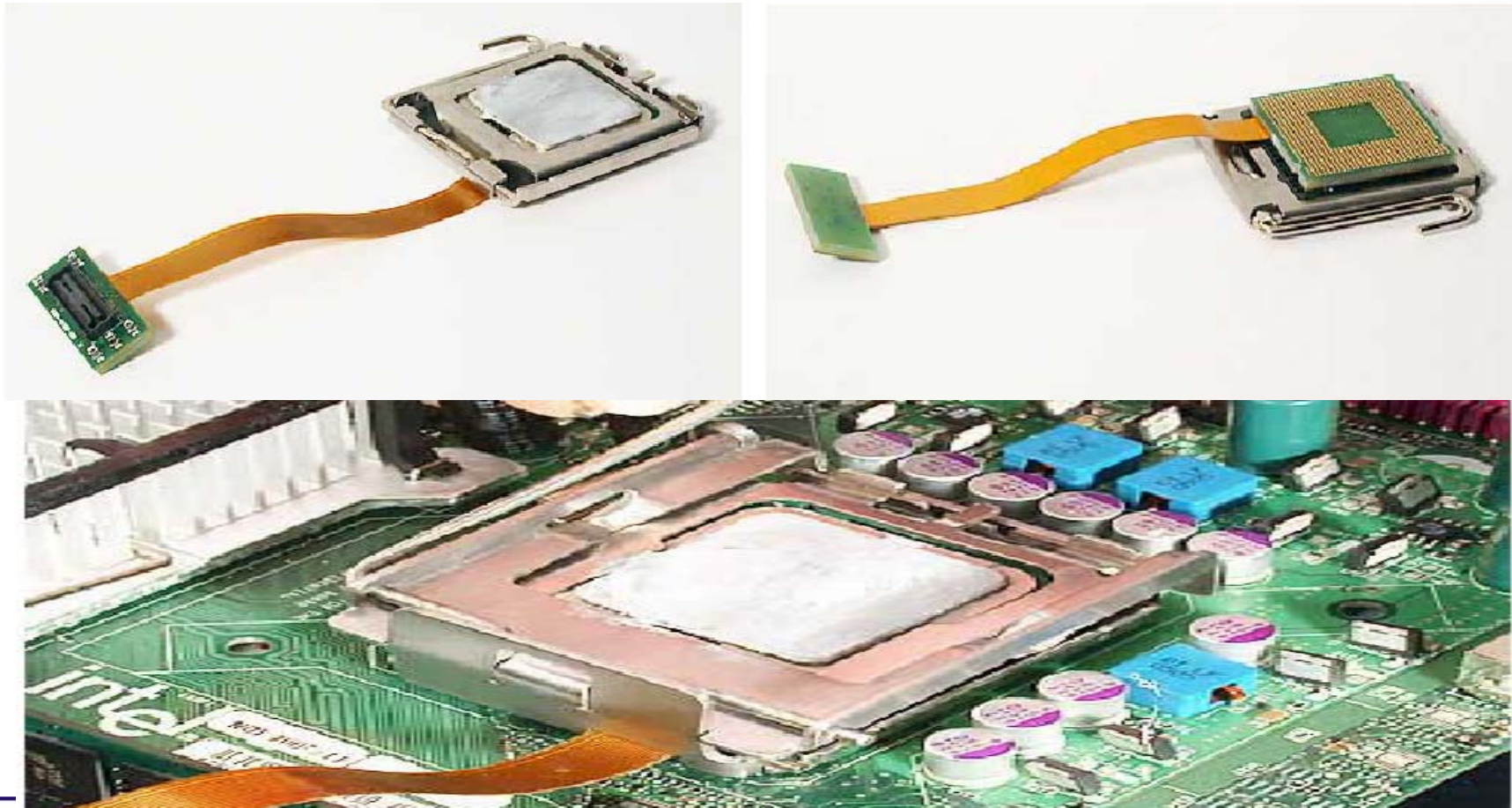


FLASH TECHNOLOGY

One Stop Tools Solution

Interposers -- Continued

- ♦ LGA775/771 interposer (used with ECM-XDP)
Not manufactured by American Arium
But you also can purchase from us

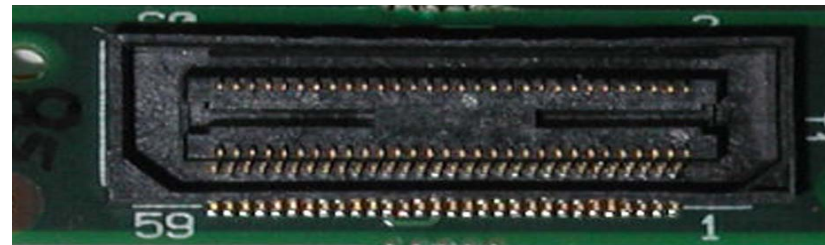
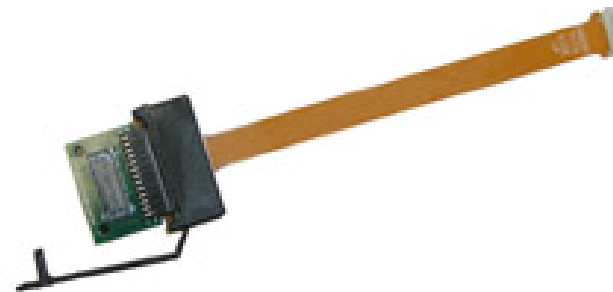


FLASH TECHNOLOGY

One Stop Tools Solution

Converter and Adapter

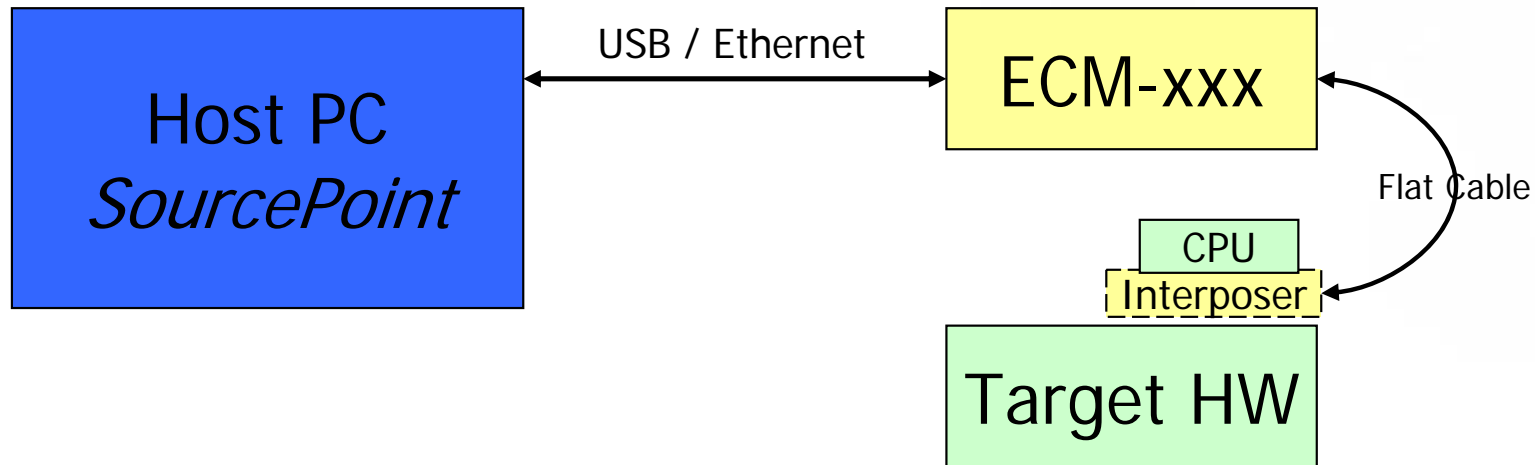
- ♦ ITP-FLX
25->28 Pins converter to
small ITP (ITP700 Flex)
used with ECM-700
- ♦ ITP-FLXXDP
28->60pin converter
used with ECM-XDP
- ♦ XDP-SSA
60->31 pin SSA



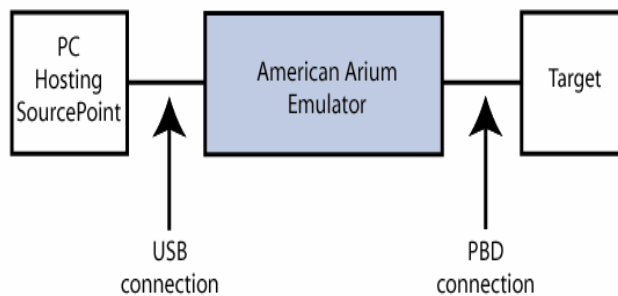
FLASH TECHNOLOGY

One Stop Tools Solution

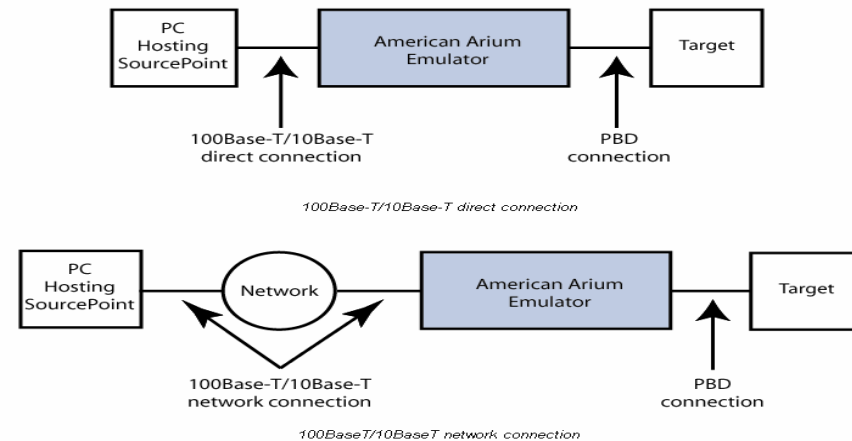
System Setup



USB Connection



Ethernet Connections



FLASH TECHNOLOGY

One Stop Tools Solution

SourcePoint™ Debugger Interface

The screenshot displays the SourcePoint debugger interface for an Altera 29K100 device. The main window shows the source code for a file named `flash.c`, which is part of a project located at `C:\Program Files\American Arrium\ARM\Linux\src`. The code is currently at line 547, which is a comment: `/* This is where the file system is built and initialized. */`. The code defines a function `jffs_build_fs` that builds and initializes the file system. The function takes a `struct super_block *sb` as an argument and returns an integer value. The code includes comments about cleaning up old allocated memory and trying again if an error occurs.

The right-hand side of the interface contains several panels:

- Control Registers:** A table showing the current state of various control registers. The registers are listed in the first column, their names in the second, and their values in the third. The values are mostly 0, except for `R0` which is `C7EE1800` and `R15` which is `C7EC2040`.
- Symbols:** A table showing the current state of various symbols. The symbols are listed in the first column, their addresses in the second, and their values in the third. The values are mostly 0, except for `R0` which is `C7EE1800` and `R15` which is `C7EC2040`.
- Trace:** A table showing the current state of the trace. The trace is currently stopped at instruction `00000014`. The trace shows the state of the processor at each instruction, including the address, instruction, and timestamp.

The bottom of the interface shows the **Devices** panel, which lists the various hardware components of the target device, such as the `BOOT_CR`, `RESET_SR`, `IDCODE`, `SRAM0_SR`, `SRAM1_SR`, `DPSRAM0_SR`, `DPSRAM0_DCR`, `DPSRAM1_SR`, `DPSRAM1_DCR`, `UART`, `UART0`, `UART1`, `UART2`, `UART3`, `UART4`, `UART5`, `UART6`, `UART7`, `UART8`, `UART9`, `UART10`, `UART11`, `UART12`, `UART13`, `UART14`, `UART15`, `UART16`, `UART17`, `UART18`, `UART19`, `UART20`, `UART21`, `UART22`, `UART23`, `UART24`, `UART25`, `UART26`, `UART27`, `UART28`, `UART29`, `UART30`, `UART31`, `UART32`, `UART33`, `UART34`, `UART35`, `UART36`, `UART37`, `UART38`, `UART39`, `UART40`, `UART41`, `UART42`, `UART43`, `UART44`, `UART45`, `UART46`, `UART47`, `UART48`, `UART49`, `UART50`, `UART51`, `UART52`, `UART53`, `UART54`, `UART55`, `UART56`, `UART57`, `UART58`, `UART59`, `UART60`, `UART61`, `UART62`, `UART63`, `UART64`, `UART65`, `UART66`, `UART67`, `UART68`, `UART69`, `UART70`, `UART71`, `UART72`, `UART73`, `UART74`, `UART75`, `UART76`, `UART77`, `UART78`, `UART79`, `UART80`, `UART81`, `UART82`, `UART83`, `UART84`, `UART85`, `UART86`, `UART87`, `UART88`, `UART89`, `UART90`, `UART91`, `UART92`, `UART93`, `UART94`, `UART95`, `UART96`, `UART97`, `UART98`, `UART99`, `UART100`, `UART101`, `UART102`, `UART103`, `UART104`, `UART105`, `UART106`, `UART107`, `UART108`, `UART109`, `UART110`, `UART111`, `UART112`, `UART113`, `UART114`, `UART115`, `UART116`, `UART117`, `UART118`, `UART119`, `UART120`, `UART121`, `UART122`, `UART123`, `UART124`, `UART125`, `UART126`, `UART127`, `UART128`, `UART129`, `UART130`, `UART131`, `UART132`, `UART133`, `UART134`, `UART135`, `UART136`, `UART137`, `UART138`, `UART139`, `UART140`, `UART141`, `UART142`, `UART143`, `UART144`, `UART145`, `UART146`, `UART147`, `UART148`, `UART149`, `UART150`, `UART151`, `UART152`, `UART153`, `UART154`, `UART155`, `UART156`, `UART157`, `UART158`, `UART159`, `UART160`, `UART161`, `UART162`, `UART163`, `UART164`, `UART165`, `UART166`, `UART167`, `UART168`, `UART169`, `UART170`, `UART171`, `UART172`, `UART173`, `UART174`, `UART175`, `UART176`, `UART177`, `UART178`, `UART179`, `UART180`, `UART181`, `UART182`, `UART183`, `UART184`, `UART185`, `UART186`, `UART187`, `UART188`, `UART189`, `UART190`, `UART191`, `UART192`, `UART193`, `UART194`, `UART195`, `UART196`, `UART197`, `UART198`, `UART199`, `UART200`, `UART201`, `UART202`, `UART203`, `UART204`, `UART205`, `UART206`, `UART207`, `UART208`, `UART209`, `UART210`, `UART211`, `UART212`, `UART213`, `UART214`, `UART215`, `UART216`, `UART217`, `UART218`, `UART219`, `UART220`, `UART221`, `UART222`, `UART223`, `UART224`, `UART225`, `UART226`, `UART227`, `UART228`, `UART229`, `UART230`, `UART231`, `UART232`, `UART233`, `UART234`, `UART235`, `UART236`, `UART237`, `UART238`, `UART239`, `UART240`, `UART241`, `UART242`, `UART243`, `UART244`, `UART245`, `UART246`, `UART247`, `UART248`, `UART249`, `UART250`, `UART251`, `UART252`, `UART253`, `UART254`, `UART255`, `UART256`, `UART257`, `UART258`, `UART259`, `UART260`, `UART261`, `UART262`, `UART263`, `UART264`, `UART265`, `UART266`, `UART267`, `UART268`, `UART269`, `UART270`, `UART271`, `UART272`, `UART273`, `UART274`, `UART275`, `UART276`, `UART277`, `UART278`, `UART279`, `UART280`, `UART281`, `UART282`, `UART283`, `UART284`, `UART285`, `UART286`, `UART287`, `UART288`, `UART289`, `UART290`, `UART291`, `UART292`, `UART293`, `UART294`, `UART295`, `UART296`, `UART297`, `UART298`, `UART299`, `UART300`, `UART301`, `UART302`, `UART303`, `UART304`, `UART305`, `UART306`, `UART307`, `UART308`, `UART309`, `UART310`, `UART311`, `UART312`, `UART313`, `UART314`, `UART315`, `UART316`, `UART317`, `UART318`, `UART319`, `UART320`, `UART321`, `UART322`, `UART323`, `UART324`, `UART325`, `UART326`, `UART327`, `UART328`, `UART329`, `UART330`, `UART331`, `UART332`, `UART333`, `UART334`, `UART335`, `UART336`, `UART337`, `UART338`, `UART339`, `UART340`, `UART341`, `UART342`, `UART343`, `UART344`, `UART345`, `UART346`, `UART347`, `UART348`, `UART349`, `UART350`, `UART351`, `UART352`, `UART353`, `UART354`, `UART355`, `UART356`, `UART357`, `UART358`, `UART359`, `UART360`, `UART361`, `UART362`, `UART363`, `UART364`, `UART365`, `UART366`, `UART367`, `UART368`, `UART369`, `UART370`, `UART371`, `UART372`, `UART373`, `UART374`, `UART375`, `UART376`, `UART377`, `UART378`, `UART379`, `UART380`, `UART381`, `UART382`, `UART383`, `UART384`, `UART385`, `UART386`, `UART387`, `UART388`, `UART389`, `UART390`, `UART391`, `UART392`, `UART393`, `UART394`, `UART395`, `UART396`, `UART397`, `UART398`, `UART399`, `UART400`, `UART401`, `UART402`, `UART403`, `UART404`, `UART405`, `UART406`, `UART407`, `UART408`, `UART409`, `UART410`, `UART411`, `UART412`, `UART413`, `UART414`, `UART415`, `UART416`, `UART417`, `UART418`, `UART419`, `UART420`, `UART421`, `UART422`, `UART423`, `UART424`, `UART425`, `UART426`, `UART427`, `UART428`, `UART429`, `UART430`, `UART431`, `UART432`, `UART433`, `UART434`, `UART435`, `UART436`, `UART437`, `UART438`, `UART439`, `UART440`, `UART441`, `UART442`, `UART443`, `UART444`, `UART445`, `UART446`, `UART447`, `UART448`, `UART449`, `UART450`, `UART451`, `UART452`, `UART453`, `UART454`, `UART455`, `UART456`, `UART457`, `UART458`, `UART459`, `UART460`, `UART461`, `UART462`, `UART463`, `UART464`, `UART465`, `UART466`, `UART467`, `UART468`, `UART469`, `UART470`, `UART471`, `UART472`, `UART473`, `UART474`, `UART475`, `UART476`, `UART477`, `UART478`, `UART479`, `UART480`, `UART481`, `UART482`, `UART483`, `UART484`, `UART485`, `UART486`, `UART487`, `UART488`, `UART489`, `UART490`, `UART491`, `UART492`, `UART493`, `UART494`, `UART495`, `UART496`, `UART497`, `UART498`, `UART499`, `UART500`, `UART501`, `UART502`, `UART503`, `UART504`, `UART505`, `UART506`, `UART507`, `UART508`, `UART509`, `UART510`, `UART511`, `UART512`, `UART513`, `UART514`, `UART515`, `UART516`, `UART517`, `UART518`, `UART519`, `UART520`, `UART521`, `UART522`, `UART523`, `UART524`, `UART525`, `UART526`, `UART527`, `UART528`, `UART529`, `UART530`, `UART531`, `UART532`, `UART533`, `UART534`, `UART535`, `UART536`, `UART537`, `UART538`, `UART539`, `UART540`, `UART541`, `UART542`, `UART543`, `UART544`, `UART545`, `UART546`, `UART547`, `UART548`, `UART549`, `UART550`, `UART551`, `UART552`, `UART553`, `UART554`, `UART555`, `UART556`, `UART557`, `UART558`, `UART559`, `UART560`, `UART561`, `UART562`, `UART563`, `UART564`, `UART565`, `UART566`, `UART567`, `UART568`, `UART569`, `UART570`, `UART571`, `UART572`, `UART573`, `UART574`, `UART575`, `UART576`, `UART577`, `UART578`, `UART579`, `UART580`, `UART581`, `UART582`, `UART583`, `UART584`, `UART585`, `UART586`, `UART587`, `UART588`, `UART589`, `UART590`, `UART591`, `UART592`, `UART593`, `UART594`, `UART595`, `UART596`, `UART597`, `UART598`, `UART599`, `UART600`, `UART601`, `UART602`, `UART603`, `UART604`, `UART605`, `UART606`, `UART607`, `UART608`, `UART609`, `UART610`, `UART611`, `UART612`, `UART613`, `UART614`, `UART615`, `UART616`, `UART617`, `UART618`, `UART619`, `UART620`, `UART621`, `UART622`, `UART623`, `UART624`, `UART625`, `UART626`, `UART627`, `UART628`, `UART629`, `UART630`, `UART631`, `UART632`, `UART633`, `UART634`, `UART635`, `UART636`, `UART637`, `UART638`, `UART639`, `UART640`, `UART641`, `UART642`, `UART643`, `UART644`, `UART645`, `UART646`, `UART647`, `UART648`, `UART649`, `UART650`, `UART651`, `UART652`, `UART653`, `UART654`, `UART655`, `UART656`, `UART657`, `UART658`, `UART659`, `UART660`, `UART661`, `UART662`, `UART663`, `UART664`, `UART665`, `UART666`, `UART667`, `UART668`, `UART669`, `UART670`, `UART671`, `UART672`, `UART673`, `UART674`, `UART675`, `UART676`, `UART677`, `UART678`, `UART679`, `UART680`, `UART681`, `UART682`, `UART683`, `UART684`, `UART685`, `UART686`, `UART687`, `UART688`, `UART689`, `UART690`, `UART691`, `UART692`, `UART693`, `UART694`, `UART695`, `UART696`, `UART697`, `UART698`, `UART699`, `UART700`, `UART701`, `UART702`, `UART703`, `UART704`, `UART705`, `UART706`, `UART707`, `UART708`, `UART709`, `UART710`, `UART711`, `UART712`, `UART713`, `UART714`, `UART715`, `UART716`, `UART717`, `UART718`, `UART719`, `UART720`, `UART721`, `UART722`, `UART723`, `UART724`, `UART725`, `UART726`, `UART727`, `UART728`, `UART729`, `UART730`, `UART731`, `UART732`, `UART733`, `UART734`, `UART735`, `UART736`, `UART737`, `UART738`, `UART739`, `UART740`, `UART741`, `UART742`, `UART743`, `UART744`, `UART745`, `UART746`, `UART747`, `UART748`, `UART749`, `UART750`, `UART751`, `UART752`, `UART753`, `UART754`, `UART755`, `UART756`, `UART757`, `UART758`, `UART759`, `UART760`, `UART761`, `UART762`, `UART763`, `UART764`, `UART765`, `UART766`, `UART767`, `UART768`, `UART769`, `UART770`, `UART771`, `UART772`, `UART773`, `UART774`, `UART775`, `UART776`, `UART777`, `UART778`, `UART779`, `UART780`, `UART781`, `UART782`, `UART783`, `UART784`, `UART785`, `UART786`, `UART787`, `UART788`, `UART789`, `UART790`, `UART791`, `UART792`, `UART793`, `UART794`, `UART795`, `UART796`, `UART797`, `UART798`, `UART799`, `UART800`, `UART801`, `UART802`, `UART803`, `UART804`, `UART805`, `UART806`, `UART807`, `UART808`, `UART809`, `UART810`, `UART811`, `UART812`, `UART813`, `UART814`, `UART815`, `UART816`, `UART817`, `UART818`, `UART819`, `UART820`, `UART821`, `UART822`, `UART823`, `UART824`, `UART825`, `UART826`, `UART827`, `UART828`, `UART829`, `UART830`, `UART831`, `UART832`, `UART833`, `UART834`, `UART835`, `UART836`, `UART837`, `UART838`, `UART839`, `UART840`, `UART841`, `UART842`, `UART843`, `UART844`, `UART845`, `UART846`, `UART847`, `UART848`, `UART849`, `UART850`, `UART851`, `UART852`, `UART853`, `UART854`, `UART855`, `UART856`, `UART857`, `UART858`, `UART859`, `UART860`, `UART861`, `UART862`, `UART863`, `UART864`, `UART865`, `UART866`, `UART867`, `UART868`, `UART869`, `UART870`, `UART871`, `UART872`, `UART873`, `UART874`, `UART875`, `UART876`, `UART877`, `UART878`, `UART879`, `UART880`, `UART881`, `UART882`, `UART883`, `UART884`, `UART885`, `UART886`, `UART887`, `UART888`, `UART889`, `UART890`, `UART891`, `UART892`, `UART893`, `UART894`, `UART895`, `UART896`, `UART897`, `UART898`, `UART899`, `UART900`, `UART901`, `UART902`, `UART903`, `UART904`, `UART905`, `UART906`, `UART907`, `UART908`, `UART909`, `UART910`, `UART911`, `UART912`, `UART913`, `UART914`, `UART915`, `UART916`, `UART917`, `UART918`, `UART919`, `UART920`, `UART921`, `UART922`, `UART923`, `UART924`, `UART925`, `UART926`, `UART927`, `UART928`, `UART929`, `UART930`, `UART931`, `UART932`, `UART933`, `UART934`, `UART935`, `UART936`, `UART937`, `UART938`, `UART939`, `UART940`, `UART941`, `UART942`, `UART943`, `UART944`, `UART945`, `UART946`, `UART947`, `UART948`, `UART949`, `UART950`, `UART951`, `UART952`, `UART953`, `UART954`, `UART955`, `UART956`, `UART957`, `UART958`, `UART959`, `UART960`, `UART961`, `UART962`, `UART963`, `UART964`, `UART965`, `UART966`, `UART967`, `UART968`, `UART969`, `UART970`, `UART971`, `UART972`, `UART973`, `UART974`, `UART975`, `UART976`, `UART977`, `UART978`, `UART979`, `UART980`, `UART981`, `UART982`, `UART983`, `UART984`, `UART985`, `UART986`, `UART987`, `UART988`, `UART989`, `UART990`, `UART991`, `UART992`, `UART993`, `UART994`, `UART995`, `UART996`, `UART997`, `UART998`, `UART999`, `UART1000`, `UART1001`, `UART1002`, `UART1003`, `UART1004`, `UART1005`, `UART1006`, `UART1007`, `UART1008`, `UART1009`, `UART1010`, `UART1011`, `UART1012`, `UART1013`, `UART1014`, `UART1015`, `UART1016`, `UART1017`, `UART1018`, `UART1019`, `UART1020`, `UART1021`, `UART1022`, `UART1023`, `UART1024`, `UART1025`, `UART1026`, `UART1027`, `UART1028`, `UART1029`, `UART1030`, `UART1031`, `UART1032`, `UART1033`, `UART1034`, `UART1035`, `UART1036`, `UART1037`, `UART1038`, `UART1039`, `UART1040`, `UART1041`, `UART1042`, `UART1043`, `UART1044`, `UART1045`, `UART1046`, `UART1047`, `UART1048`, `UART1049`, `UART1050`, `UART1051`, `UART1052`, `UART1053`, `UART1054`, `UART1055`, `UART1056`, `UART1057`, `UART1058`, `UART1059`, `UART1060`, `UART1061`, `UART1062`, `UART1063`, `UART1064`, `UART1065`, `UART1066`, `UART1067`, `UART1068`, `UART1069`, `UART1070`, `UART1071`, `UART1072`, `UART1073`, `UART1074`, `UART1075`, `UART1076`, `UART1077`, `UART1078`, `UART1079`, `UART1080`, `UART1081`, `UART1082`, `UART1083`, `UART1084`, `UART1085`, `UART1086`, `UART1087`, `UART1088`, `UART1089`, `UART1090`, `UART1091`, `UART1092`, `UART1093`, `UART1094`, `UART1095`, `UART1096`, `UART1097`, `UART1098`, `UART1099`, `UART1100`, `UART1101`, `UART1102`, `UART1103`, `UART1104`, `UART1105`, `UART1106`, `UART1107`, `UART1108`, `UART1109`, `UART1110`, `UART1111`, `UART1112`, `UART1113`, `UART1114`, `UART1115`, `UART1116`, `UART1117`, `UART1118`, `UART1119`, `UART1120`, `UART1121`, `UART1122`, `UART1123`, `UART1124`, `UART1125`, `UART1126`, `UART1127`, `UART1128`, `UART1129`, `UART1130`, `UART1131`, `UART1132`, `UART1133`, `UART1134`, `UART1135`, `UART1136`, `UART1137`, `UART1138`, `UART1139`, `UART1140`, `UART1141`, `UART1142`, `UART1143`, `UART1144`, `UART1145`, `UART1146`, `UART1147`, `UART1148`, `UART1149`, `UART1150`, `UART1151`, `UART1152`, `UART1153`, `UART1154`, `UART1155`, `UART1156`, `UART1157`, `UART1158`, `UART1159`, `UART1160`, `UART1161`, `UART1162`, `UART1163`, `UART1164`, `UART1165`

SourcePoint Key Features

- ◆ General
 - Run Control from target reset (FFFFFFFF0)
 - Register access
 - PCI Device View
 - TCP/IP setup wizard
 - Context menus
 - Docking windows
 - Comprehensive C-like command language
- ◆ Source-Level Debug
 - Advanced breakpoints
 - Symbol Search
 - Mixed mode/ Source Code
- ◆ Target Specific
 - Reliably debug from reset
 - Multi-processor/core/cluster
- ◆ Operating System
 - Intel EFI Framework debug



FLASH TECHNOLOGY

One Stop Tools Solution

Source-Level Debug

```
Code P0*: (32 bit) Tracking IP: 0010:00000000 - 0010:FFFFFFFF
128
129     static struct _FooStruct fooStruct = { 1, &fooStruct._int, "CSam
130
131     const char* pfoo = "fooString";
132     const char fooStr[] = "fooString";
133
134     void* pVoid = (void*) 0xdeadbeef;
135
136     int main()
137     {
138 main:
139     0010:00001000 55          PUSH     EBP
140     0010:00001001 89E5        MOV      EBP,ESP
141     0010:00001003 83EC18      SUB      ESP,00000018
142
143     138         int f = factorial( 5 );
144     0010:00001006 6A05        PUSH     00000005
145     0010:00001008 E8F7000000  CALL     near32 ptr factorial
146     0010:0000100D 83C404      ADD      ESP,00000004
147     0010:00001010 89C0        MOV      EAX,EAX
148     0010:00001012 8945FC      MOV      dword ptr [EBP]-04,EAX
149
150     139         int index = 0;
151     0010:00001015 C745F8000000 MOV      dword ptr [EBP]-08,00000000
152
153     140
154     141         while( ++index )
155     0010:0000101C 8D742600    LEA      ESI,dword ptr [ESI]
156     0010:00001020 FF45F8      INC      dword ptr [EBP]-08
157     0010:00001023 837DF800    CMP      dword ptr [EBP]-08,00000000
```

```
Code P0*: Tracking IP: C:\Program Files\American Arrium\Intel\SAMPLES\ELFC_SAMPLE\csample.c
125
126     struct _FooStruct fooStructArray[4];
127
128
129     static struct _FooStruct fooStruct = { 1, &fooStruct._int, "CSample" };
130
131     const char* pfoo = "fooString";
132     const char fooStr[] = "fooString";
133
134     void* pVoid = (void*) 0xdeadbeef;
135
136     int main()
137     {
138         int f = factorial( 5 );
139         int index = 0;
140
141         while( ++index )
142         {
143             struct _FooStruct foo;
```

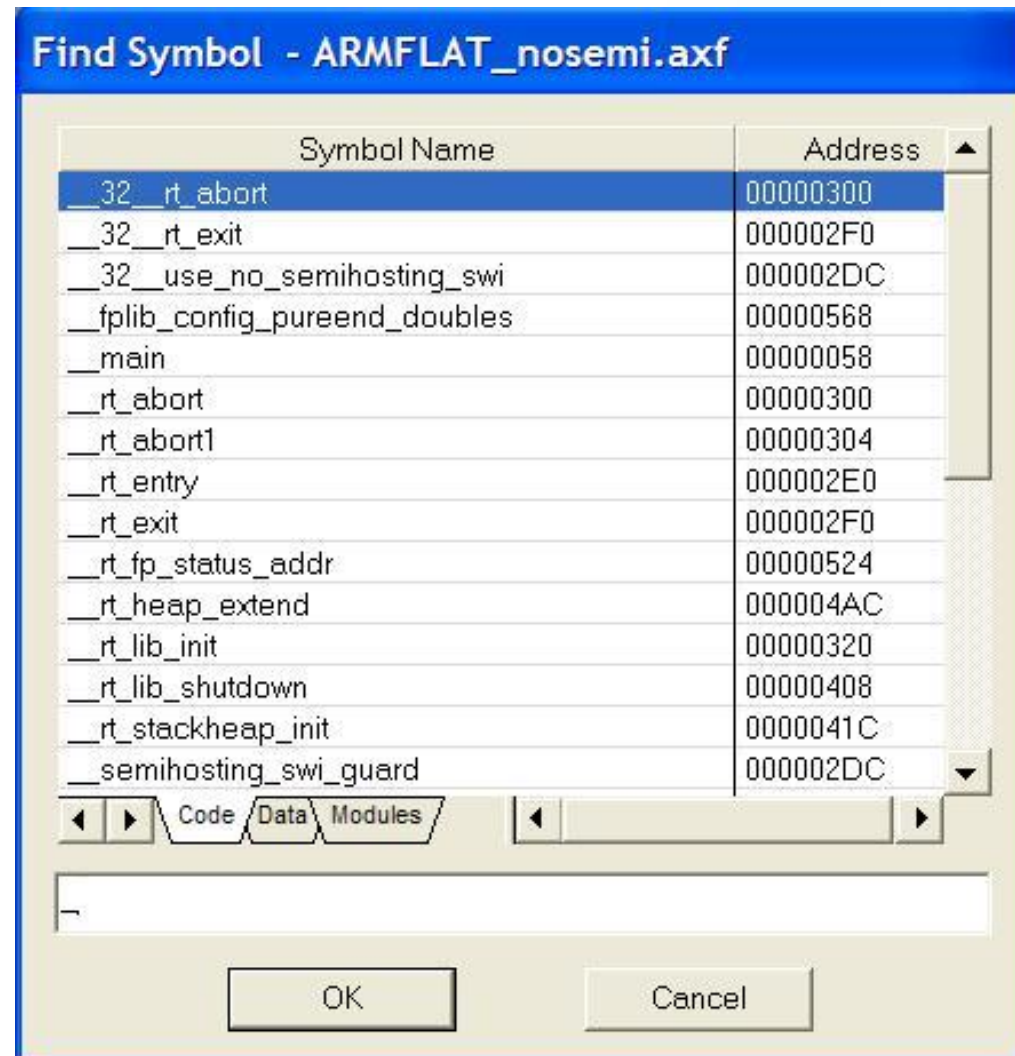
```
FFFFFFFFFEL
{ .nib add r25=0010.sp
  add r22=0020.sp
  nop 0 ; ; }
{ .mii alloc r33=pfis.0.4.1.0
  mov r32=rp
  add sp=-0040.sp ; ;
  r35=gp.r0 ; ;
  nop 0 ; ;
  mov r34=pr ; ;
  mov r2=[sp]
  r36=000005
  br.call.sptk.many rp=000000003eed64f0L ; ; }
{ .mii add r31=0038.sp
  add r30=0038.sp
  add r28=0010.sp ; ;
  gp=r35.r0 ; ;
  [r31]=ret0
  nop 0 ; ; }
{ .nmi ld4 r29=[r30] ; ;
  st4 [r28]=r29
  nop 0 ; ; }
{ .nmi add r27=0014.sp ; ;
  st4 [r27]=r0
  nop 0 ; ; }
{ .mii add r26=0014.sp
  add r23=0014.sp
  add r22=0014.sp ; ; }
{ .nmi ld4 r25=[r26] ; ;
  add r24=0001.r25
  nop 0 ; ; }
{ .nmb st4 [r23]=r24
  ld4 r21=[r22]
```



FLASH TECHNOLOGY

One Stop Tools Solution

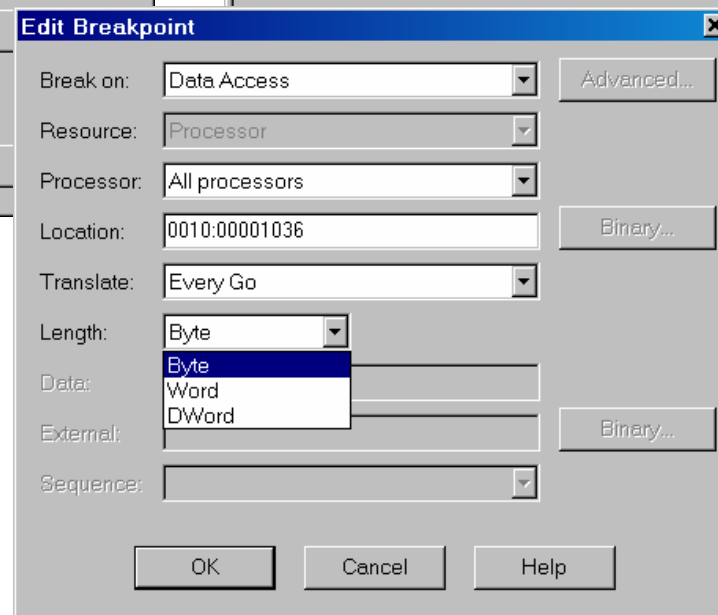
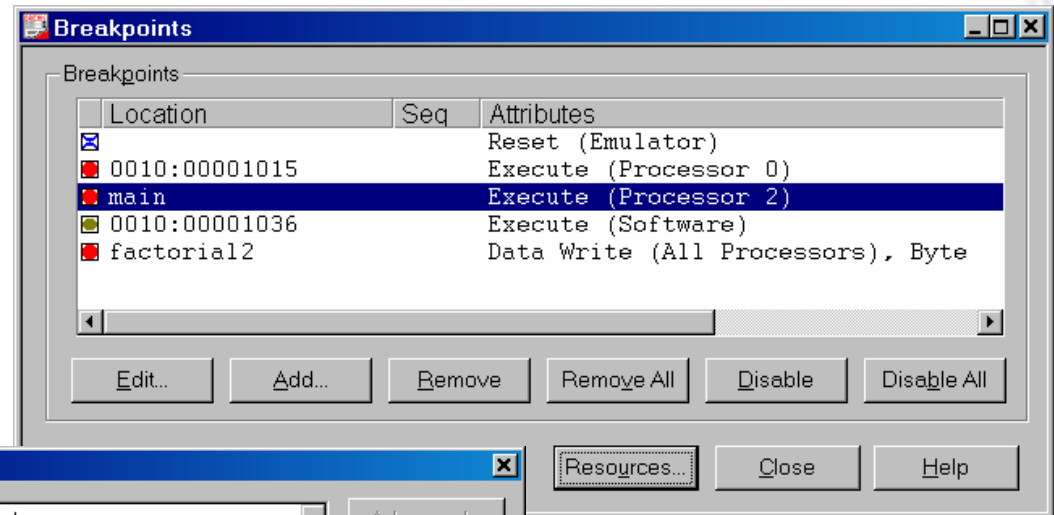
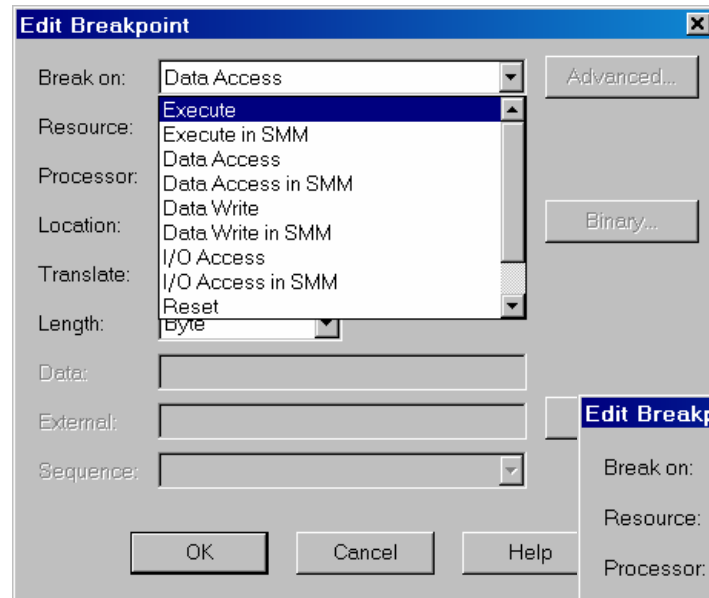
Fast Symbol Search



FLASH TECHNOLOGY

One Stop Tools Solution

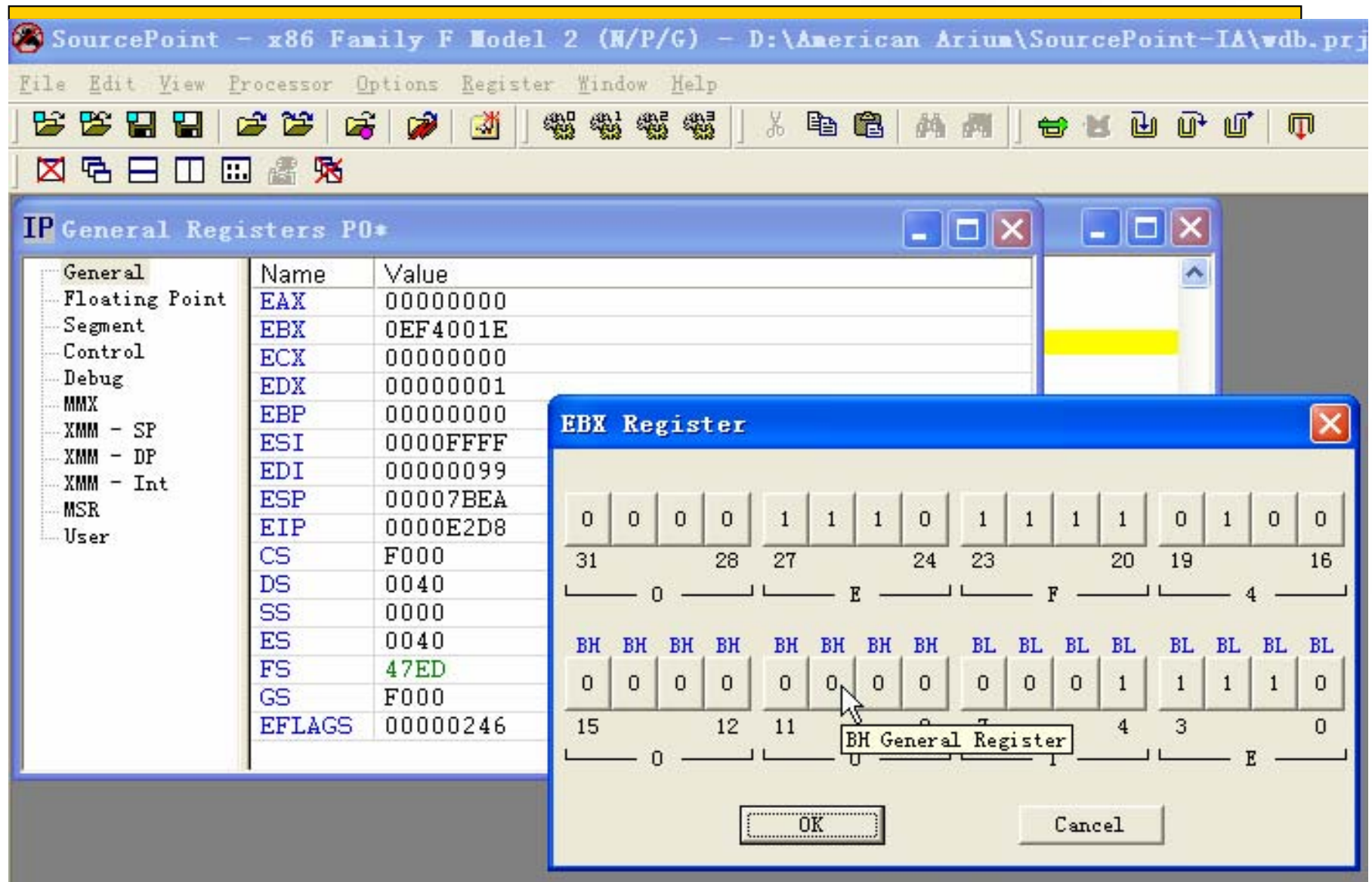
Advanced Breakpoints



FLASH TECHNOLOGY

One Stop Tools Solution

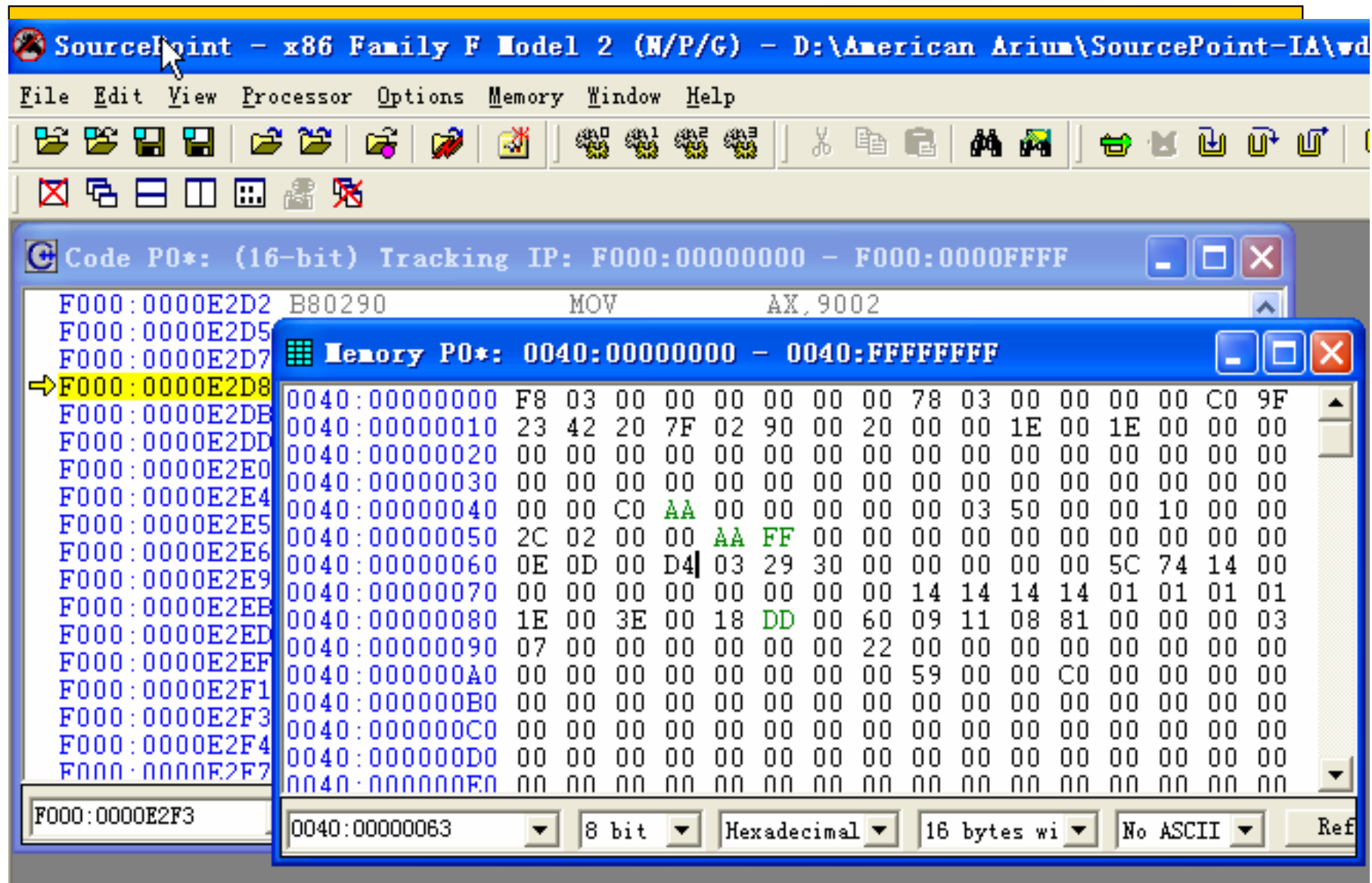
Complete Registers Control



FLASH TECHNOLOGY

One Stop Tools Solution

Quick Memory Access/Modification



FLASH TECHNOLOGY

One Stop Tools Solution

PCI Device View

PCI Devices						
Vendor	Vendor ID	Device ID	Bus	Dev	Func	Description
Intel Corporation	8086	2570	0	0	0	DRAM Controller / Host-Hub Interface
Intel Corporation	8086	2572	0	2	0	Integrated Graphics Device
Intel Corporation	8086	2573	0	3	0	PCI-to-CSA Bridge
Intel Corporation	8086	24d2	0	29	0	USB UHCI Controller #1
Intel Corporation	8086	24d4	0	29	1	USB UHCI Controller #2
Intel Corporation	8086	24d7	0	29	2	USB UHCI Controller #3
Intel Corporation	8086	24de	0	29	3	USB UHCI Controller #4
Intel Corporation	8086	24dd	0	29	7	USB EHCI Controller
Intel Corporation	8086	244e	0	30	0	Hub Interface to PCI Bridge
Intel Corporation	8086	1019	1	1	0	Gigabit Ethernet Controller (LOM)

Refresh PCI Devices

PCI devices setup

First bus:

Last bus:

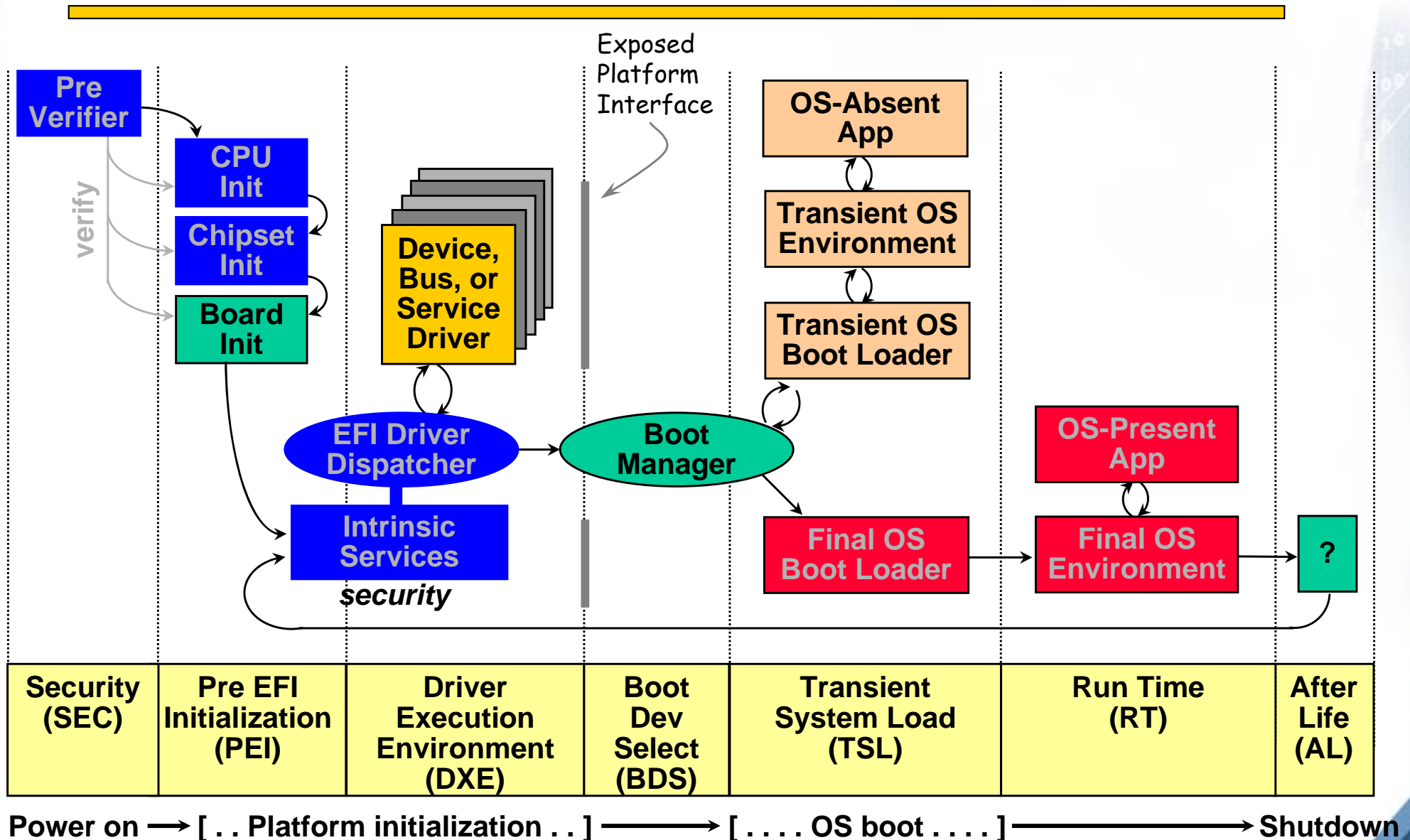
PCI Registers[00:1d:00]: Intel Corporation:USB UHCI Controller #1		
Register	Offset	Value
Vendor ID	00	8086
Device ID	02	24D2
Command Register	04	0000
Status Register	06	0000
Revision ID	08	00
Class Code	09	000000
Cache Line Size	0C	00
Latency Timer	0D	00
Header Type	0E	0
BIST	0F	00
Base Address Register 0	10	00000000
Base Address Register 1	14	00000000
Base Address Register 2	18	00000000
Base Address Register 3	1C	00000000
Base Address Register 4	20	00000000
Base Address Register 5	24	00000000
CardBus CIS Pointer	28	00000000
Subsystem Vendor ID	2C	00
Subsystem ID	2E	00
Expansion ROM Base Address	30	00000000
I/O Line	3C	00



FLASH TECHNOLOGY

One Stop Tools Solution

Framework Background

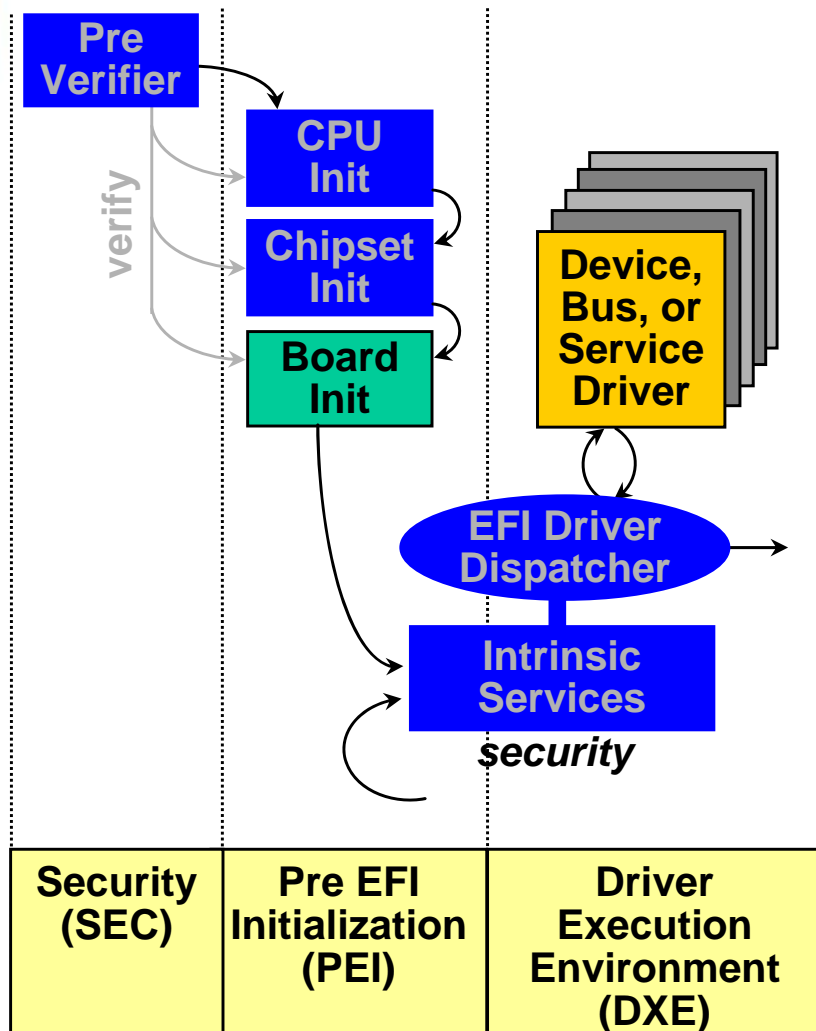


FLASH TECHNOLOGY

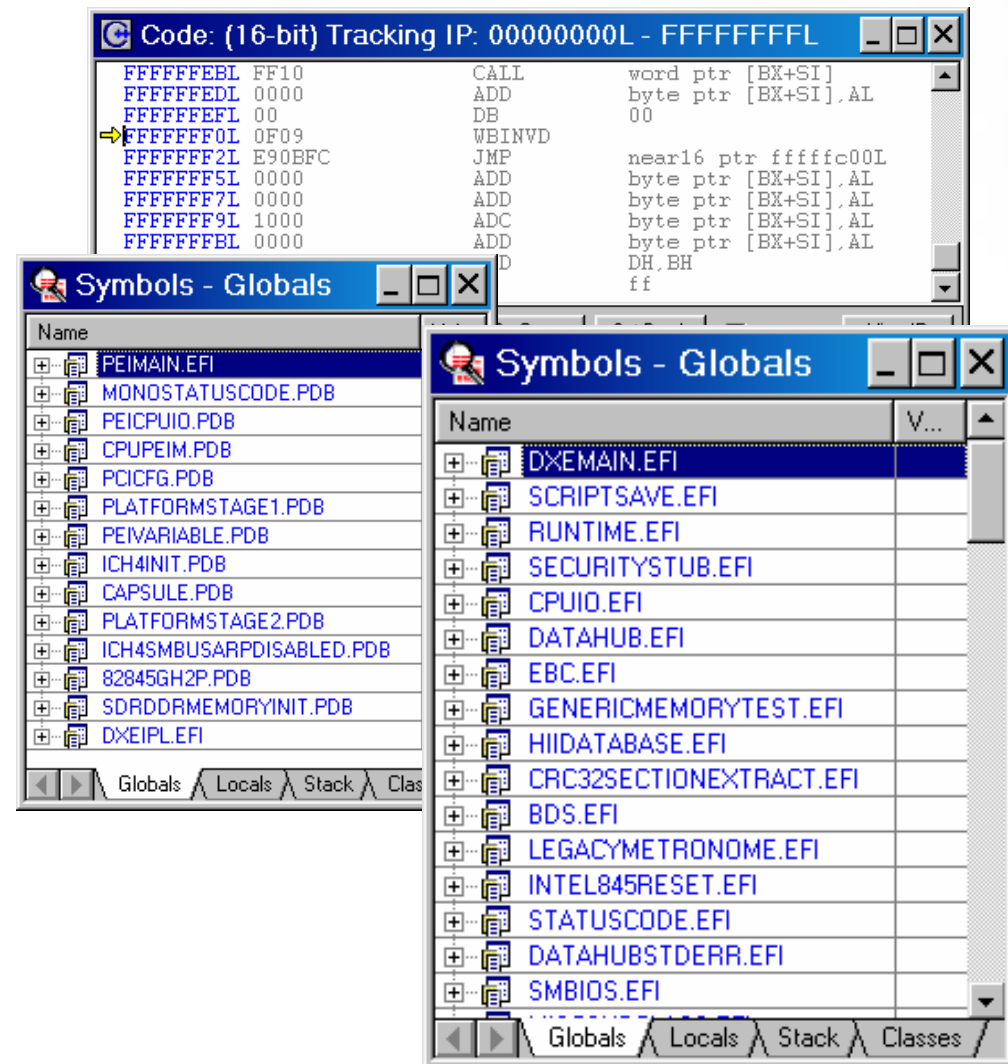
One Stop Tools Solution

Framework Debug

- ◆ Debug from reset, into PEI then DXE, up to OS boot



Power on [.. Platform initialization ..]

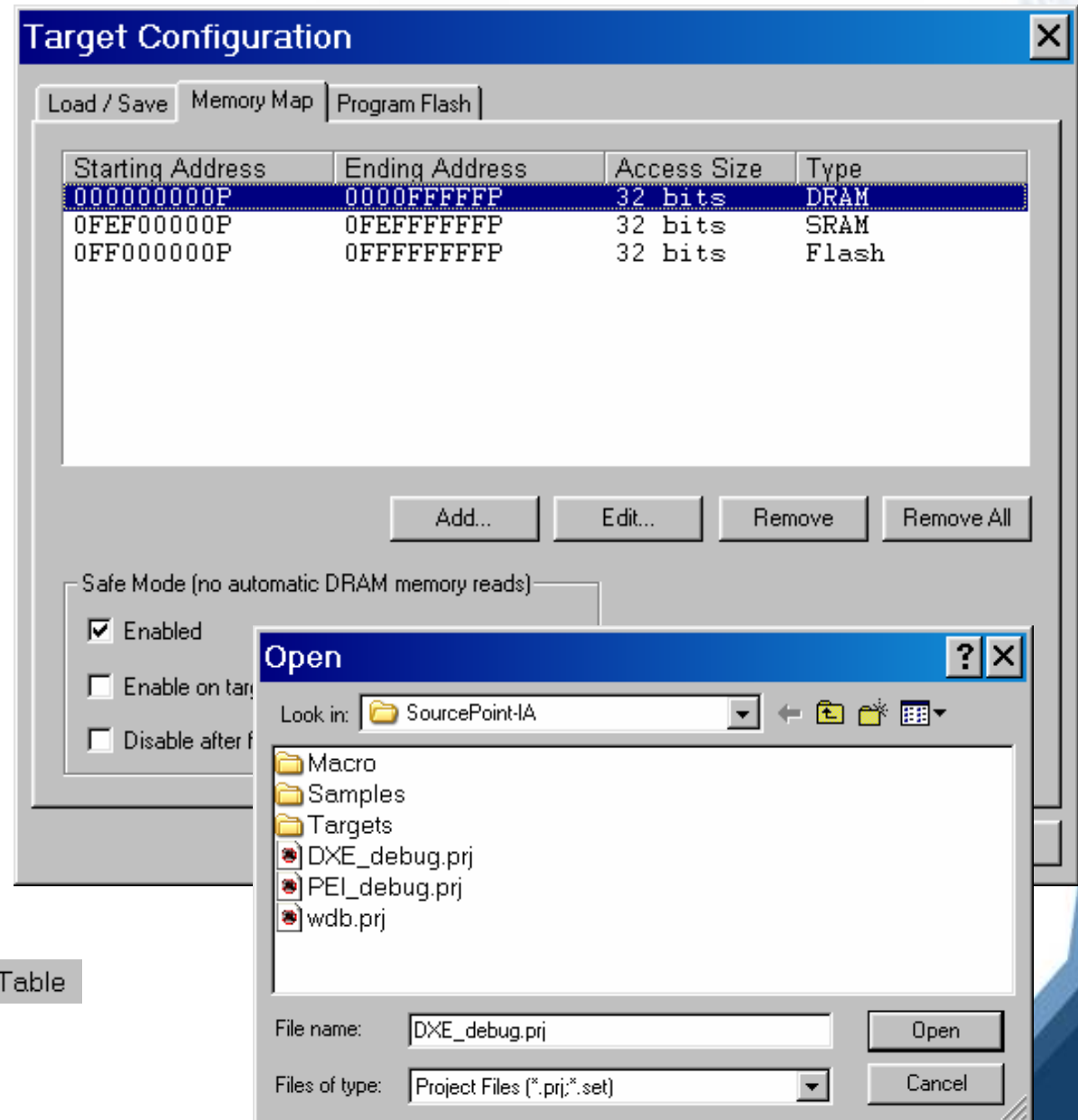


FLASH TECHNOLOGY

One Stop Tools Solution

Framework Debug Environment

- ◆ SourcePoint uses different project files to set up debug environment for...
 - PEI
 - DXE
- ◆ Framework debug uses macros to...
 - Disable watchdog timer
 - Setup PEI and DXE debug modes
 - Display HOBs
 - Display System Configuration Table



PEIMs DXEs HOBs SysConfigTable



FLASH TECHNOLOGY

One Stop Tools Solution

Framework Debug: PEI

- ◆ Break in early PEI
 - Load PEI project
 - Reset processor
 - Step to or run to CPUID instruction
 - Load PEI modules
 - Set breakpoint on function PeiMain
 - Open Trace window
 - Open tracking Code window
 - Step back through source code
- ◆ Break on beginning transition to DXE
 - Set breakpoint in peimain.c at peicore+0x4b2 and go

The screenshot displays three debugger windows. The 'Breakpoints' window shows two breakpoints: one at 0010:FFFFFC75 and another at PeiCore+0x4b2, both with 'Execute (Processor)' attributes. The 'Trace' window shows a list of instructions starting with 'CALL DS:[EDI]' and 'PUSH EBP'. The 'Code' window shows the source code for 'peimain.c' with a cursor at line 98, which is 'PeiMain (&PeiStartup);'. The code includes conditional compilation for the EFI_NT_EMULATOR and sets up the PeiStartup structure.

Breakpoints

Location	Seq	Attributes
0010:FFFFFC75		Execute (Processor)
PeiCore+0x4b2		Execute (Processor)

Trace

STATE	ADDR	INSTRUCTION
	FFFFFFE11	CALL DS:[EDI]
-00001	FFFE033A	PUSH EBP
	FFFE033B	MOV EBP, ESP
	FFFE033D	SUB ESP, 00000010
	FFFE0340	MOV EAX, [EBP]+08
	FFFE0343	MOV [EBP]-08, EAX
	FFFE0346	MOV EAX, [EBP]+0C
	FFFE0349	MOV [EBP]-0C, EAX
	FFFE034C	LEA EAX, [EBP]-10
	FFFE034F	PUSH EAX
	FFFE0350	MOV [EBP]-10, FFFE17CA
	FFFE0357	MOV [EBP]-04, FFFE2108
	FFFE035E	CALL FFFE0BC7L

Code: c:\t\apeia32\nem\secstartup.c (Tr...

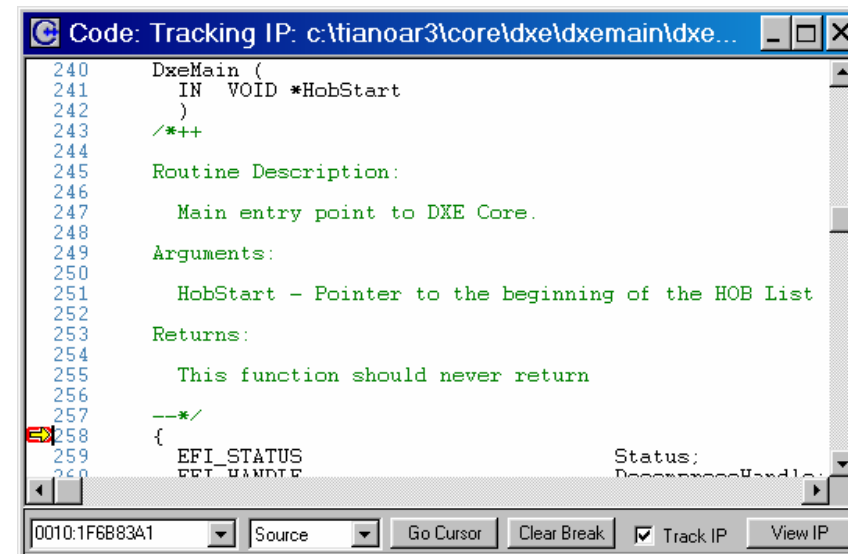
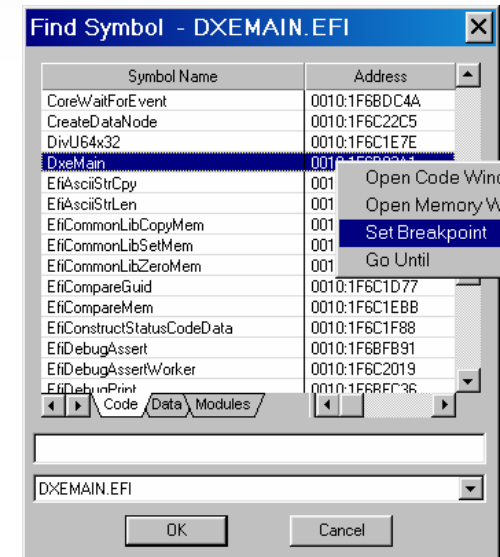
```
88  #ifdef EFI_NT_EMULATOR
89      PeiMain ((EFI_PEI_STARTUP_DESCRIPTOR *) SizeOfRam);
90  #else
91      EFI_PEI_STARTUP_DESCRIPTOR PeiStartup;
92
93      PeiStartup.SizeOfCacheAsRam      = SizeOfRam;
94      PeiStartup.BootFirmwareVolume   = BootFirmwareVolume;
95      PeiStartup.VerificationService   = StubVerify;
96      PeiStartup.PrivateDispatchTable = &mPpiList;
97
98      PeiMain (&PeiStartup);
99  #endif
100
101      return;
102  }
```

0010:FFFE034C Source Go Cursor Set Break Track IP View IP



Framework Debug: DXE

- ◆ Break on early DXE
 - Reset processor
 - Load DXE project file
 - Go until Framework prompt then stop
 - Load DXE modules
 - Ctrl-S to Find Symbol DxeMain
 - Set breakpoint at function DxeMain
 - Reset processor and go to breakpoint

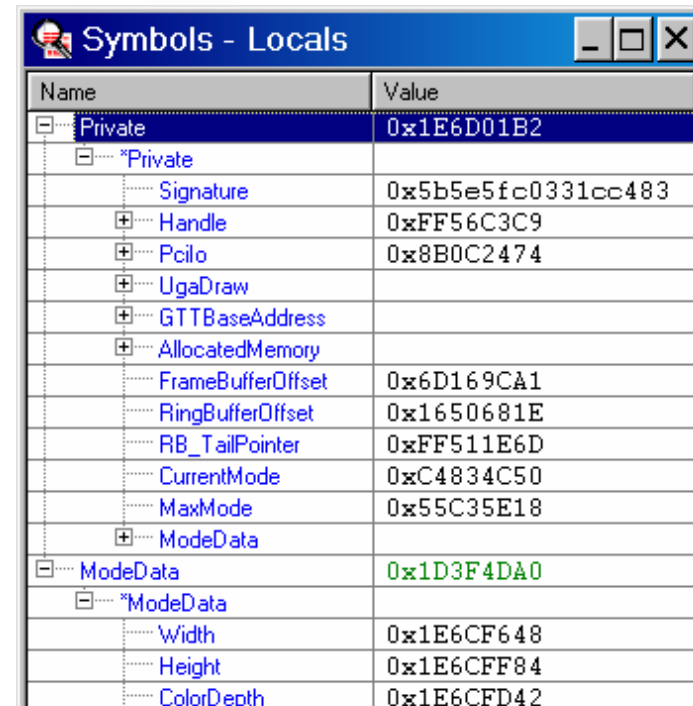


FLASH TECHNOLOGY

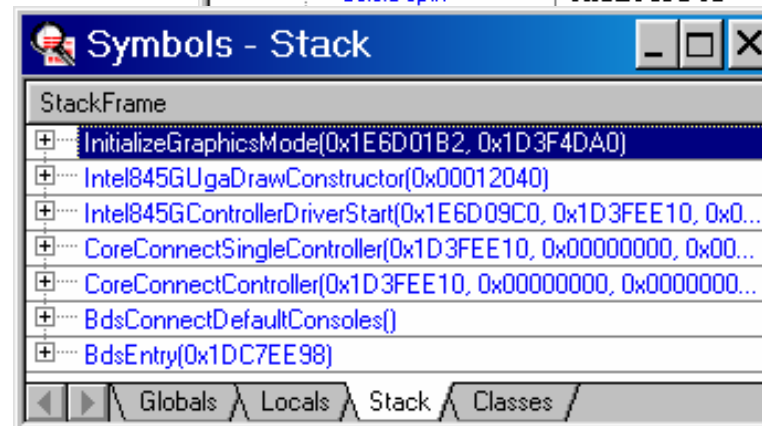
One Stop Tools Solution

Framework Debug: DXE

- ◆ Use Symbol window to set breakpoint on InitializeGraphicsMode
- ◆ Step into routine and view parameters passed using the Locals tab
- ◆ Select Stack tab to view stack trace
- ◆ Examine stack trace to find execution path



Name	Value
Private	0x1E6D01B2
*Private	
Signature	0x5b5e5fc0331cc483
Handle	0xFF56C3C9
Pcilo	0x8B0C2474
UgaDraw	
GTTBaseAddress	
AllocatedMemory	
FrameBufferOffset	0x6D169CA1
RingBufferOffset	0x1650681E
RB_TailPointer	0xFF511E6D
CurrentMode	0xC4834C50
MaxMode	0x55C35E18
ModeData	
*ModeData	
Width	0x1E6CF648
Height	0x1E6CFF84
ColorDepth	0x1E6CFD42



StackFrame
InitializeGraphicsMode(0x1E6D01B2, 0x1D3F4DA0)
Intel845GUgaDrawConstructor(0x00012040)
Intel845GControllerDriverStart(0x1E6D09C0, 0x1D3FEE10, 0x0...
CoreConnectSingleController(0x1D3FEE10, 0x00000000, 0x00...
CoreConnectController(0x1D3FEE10, 0x00000000, 0x00000000...
BdsConnectDefaultConsoles()
BdsEntry(0x1DC7EE98)



FLASH TECHNOLOGY

One Stop Tools Solution

Conclusions

- ◆ Hardware-assisted debug can be done from reset through OS boot
- ◆ Target will be full controllable and accessible
- ◆ Debugging EFI Framework code is straightforward
- ◆ One debugger interface for IA-32 (including 64-bit extensions), IA-64, Core 2 Solo/Duo and AMD64 for full system debug
- ◆ Speed your time to market with proven tools
- ◆ Get more reliable products to market faster



FLASH TECHNOLOGY

One Stop Tools Solution

Contact Us

- ◆ Flash Technology Trading (Shanghai) Limited
上海市长宁区天山路600弄1号同达创业大厦405
Tel: 8621-61457130 Fax: 8621-61457131
Email : sales@flashtech.com.cn (sales)
support@flashtech.com.cn (support)
- ◆ Flash Technology (HK) Limited
香港九龙长沙湾长沙湾道928-930时代中心13楼02室
Tel : 852-23109662 Fax: 852-28157209
Email: sales@flashtech.com.cn (sales)
support@flashtech.com.cn (support)



FLASH TECHNOLOGY

One Stop Tools Solution

Thank you

◆ Demonstration



FLASH TECHNOLOGY

One Stop Tools Solution