Phoenix Debugger 8.7

By JK Lee

Phoenix Technologies (Korea) Ltd.

Agendas

- Introduction - A must-have tool
- Concepts - Symbols, installation
- Basic Features - What at least you need to know
- Hardware Features - I/O, NVRAM, ACCESS
- Advanced Features - Saving you lots of time

Introduction

What PhDebug can do for you

- Powerful debug tool for debugging pre-boot, runtime & SMI code
- Better than ICE
  - Extremely low-cost solution
  - Symbolic debugging
  - Phoenix internal structures support
  - Advanced tools dedicated for Phoenix core
- Saving you lots of time

How/Where to start

- Installation
  - Source Code Customers: Get your copy from PM/Sales for free
- Windows NT Users
  - An additional kernel mode driver PhoenixAD needed for parallel port debugging
- Documentation
  - PHDEBUGW.DOC
- Help available by pressing F1 any time
- Tip of the day gives you the most important tips

Concepts

Symbols

- Symbols are all the labels declared as PUBLIC
- Supported symbol segments
  - All Version - BCG, EBCG, PMCG
  - 405 and above - PCG
  - 600 and above - RCG, NCG, SRCG
  - PnP 4.3 - pnpSegment
  - Additional - PnPDG, Engine, Display Manager, Custom Module
- When will we need to reload symbols
  - Platform changed
  - BIOS changed
- Add a symbol without FLASH
  - Re-declare it as PUBLIC
  - Re-Build
  - Reload symbols

Prepare Debugging

- Decide the port to be used
  - Bi-Directional vs. serial port
- Select debug engine
  - Build-in engine: STD.600/PENTIUM.600/REMUS.600
  - ADM engine
- Install engine in the build (needs about 3K additional space)

Install Debug Engine

- BCP.ASM
  - INSTALL_DEBUG Macro
  - INSTALL_DEBUG_OLD Macro
- Necessary hooks
  - hookBiosReset
  - I/O Initialization
  - I/O testing
- Debug engine works under real mode only

Break code & Break address

- Break Code - A pure software method, always available, POST codes are also associated
- BREAK_POINT / BPCALL
- Break address
  - A hardware feature, available only if the RAM is ready
  - Debug registers (386 and above) and INT1

Basic Features

Take a Quick Look

- Menu bar - Completed function list
- Tool bar
  - Icons present most often-used functions
  - Current code (last occurred debug code)
  - Break code / address (set by user, 0 means off)
- Status bar - Shows the statuses you should be aware of

Load Symbols

- Starting with OEM Tip
- Optional Symbol Classes
  - Setup Engine
  - Display Manager
  - Boot Block
  - Custom Module
- Auto-Reload Symbols on Startup

Symbol Files

- Loaded symbol files:
  - OEM - BUILD.MAP, BUILDDBG.LST (405)
  - OEM - BUILD.MAP, BUILD.LOG (600)
  - Setup Engine - MAKEROM.MAP (405)
  - Setup Engine - SETUP.MAP (600)
  - Display Manager - DISPMAN.MAP (600)
  - Boot Block - MAKEBB.MAP (405/600)

Segment Window

- Same purpose of MAPCONV, but is sorted with proper cased
- Symbols must be loaded
- How to know the space left in certain code group

Symbol Segment Window

- Active by pressing hot-key F3 or select from menu
- Lists loaded symbols and segment value assignments
- Edit by hand freely or use ‘auto-refresh’ via PDM/PMM for dynamically relocated modules (DISPMAN, Setup Engine .. Etc.)

Finding Symbol

- Find symbol window
  - Case sensitive search
  - First/Next/Go To/Set Break
- Break address history
  - Break addresses are automatically stored in history
  - Set as new break address from the history

Break Condition Window

- Break conditions can be set any time even the platform is running
- Setting break code
  - COLDSTRT.TXT the user definable test points file
- Setting hardware break (break address)
  - PENTIUM.600 engine is needed

Code Window

- What you can do when platform is running (not broken)
  - Copy/Clear the debug code region
  - The General-Purpose Timer
  - Start a new line
  - Set the Line-Wrap Code
  - Break by Ctrl-P/N/Ctrl-C
- Go (F5) / Go Next (F6)
- View debug code history (Ctrl-V)
- First line of the code window is always symbolized
- Memory prompter / Jump prompter
- Current CS:IP is in bold font with a >
- Browse the code and return to CS:IP (F2)
- Switch between code/data/stack region
- Sizeable Unassembler Lines (15-50)
- Change registers/flags
- View source / duplicated symbols
- Pop up menu

Trace

- Trace into (F7) vs. Step over (F8)
- Execute to cursor (F4)
- If the RAM is not ready ...
- What happens after trace
- Trace into ISR
- Step over JBX, JDI ...
- Undo last trace

Setting Break

- Break condition window
- Click/Right-Click on break labels
- Instant commands
- Hot keys
- Break history feature
- Find symbol window

Hardware vs. Software Break

- Hardware Break Condition (DR0)
  - Break on execution/memory write/IO access/memory access
- Software Break Point (INT 3)
  - Up to 128 break points
  - Ctrl-F9 for toggling
- Limitation - Break location must be Read-Writable (either in RAM or R/W enabled shadow memory)

Source Window

- How to apply ‘View Source’ function
- Freely open a source file
- Load another source file by finding a new symbol
- Select a source file from history
- Browser Style ‘Back’ & ‘Forward’

Hardware

I/O & NVRAM

- I/O Window
  - Byte/Word/Dword access
  - When will you need it?
- NVRAM Window
  - Access CMOS via token name
  - Search/Search Next
  - View/Change values

ACCESS Window

- Bit breakdown & switches
- Additional Method - Double index
- Dynamically Changeable Register Width (BYTE/WORD/DWORD)
- PCI functions
  - Direct Bus/Device/Function access
  - Scan devices, maximum bus is selectable
  - Examines Configuration Space
  - Append to ACCESS.TXT for you
- Run DOS version

Memory Functions

- Data region in code window
- Extra data windows (up to 5)
  - Add new / switch to next
  - Read-only in big real mode
- Save/Load/Compare memory
  - Up to 1MB for real mode & 8MB for BRM
  - The cache and shadow should turn on

Instant Commands

- Compatible with DOS version
- Sometimes the instant command is most useful
- Available only the platform is broken
- "Command" menu lists all commands
- Math (+, -, *, /) is available
- Multiple steps trace (P nn / T nn)
- Multiple entries allowed (i.e. O 64 20, I 60)

Options Window

- ‘Debug’ tab sets basic debug connection
- ‘Environment’ tab sets additional debug options
- ‘Appearance’ tab sets debugger behaviors
- Live Demo

Switch Between Windows

- Close a window by pressing Ctrl-F4
- Ctrl-C switch to Code Window
- Ctrl-A switch to Access Window
- Ctrl-Z switch to NVRAM Window
- Ctrl-S switch to Source Window
- And so on ...

Advanced

Additional Break Conditions

- A serial of register condition limits platform to be broken only all conditions match
- Available conditions: < = >
- Example:
  - AX=5F,CH<2F,BL>8
- All conditions are ANDed

Sequence Break

- Unlimited numbers of break sequence
- Any combination (code, execution, I/O ... etc with addition condition for each).
- Platform will be broken only if all the criterions are matched.
- Purpose:
  - Platform hangs because careless keystrokes. You don't need to start it all over again.
  - Debugging tricky bugs.
  - Handling complicated issues.

CPU Window

- View Pentium Class CPU Model Specified Registers (MSR), 64 bits format
- Especially useful for PII L2 cache status
- Needs PENTIUM.600 debug engine

BCP Analyst

- Display item, size, default/current value and comment/EQUs
- Symbols are necessary
- Check the BCP Path in 'Options' & specify the directory of BCPS.INC
- ATAGS data path is optional, if not specified, it only analysis from BCPS.INC

PnP Analyst

- Analyze the Phoenix PnP table structure during POST
- Ways to active it
- Only available after PciInit (49h)
- If no symbolic available, input the start address of the table manually
- PCI/ISA/MCD/MB device selectable for PnP 4.3 or above

PMM Analyst (600)

- Analyze all memory blocks allocated by 600 POST Memory Manager
- Hotkey Shift-F3
- Available only after PMM is initialized (POST code 29h PMM_INIT)
- Be careful not to use after boot, it works during POST only
- Platform must be in big real mode

PDM Analyst (600)

- Analyze all modules and services in ROM, RAM and Service Directory managed by 6.00 'Phoenix Dispatch Manager’
- Hotkey Shift-F4
- Symbolic debugging must be enabled
- Available only after PDM is initialized (POST code 33h TP_PDM_INIT)

POST Analyst

- Analyze the actual POST table in exact order, also shows all the hook & post routines
- Analyze current Interrupt Table
- Analyze ACPI Registered Functions Both for POST & SMM
- Hotkey Shift-F5
- Symbolic debugging needed
- Cold/Warm table for 600, cold table only for 405
- Double click on any routine name to view the code
- Usage of ‘Refresh’ Button

Register Table Analyst

- Analyze all register tables in your build
- Hotkey Shift-F6
- Especially useful for verifying chipset port
- Identify chipset type, model and read/write routines
- Available for both 405/600

Miser Analyst

- Analyzes Miser's State Transition Table
- Hotkey Shift-F7
- Lists all PM states and parses all related states with 'To' & 'From', easy to check both states by pair
- Double-click on the routine name will take you there at once
- Available only if the platform is broken in SMM (CS = A040)

Break After Boot

- The INT1 vector is destroyed after boot
- Set new break address without BREAK_POINT
- Debug device drivers
- Any function is available (IO / access)
- Trace run time functions (ISR / PnP / APM)
- ADM module supported
- Disable PM for this feature

Other Features

- Load & Save Settings, very useful when working on more than one projects
- Command line option -p
- Automatic save size & position
- Log Window with save / load functions
- Colored Break Code/Address Indicators
- Option ROM Scan
- ASCII Code Table
- Get Information
  - Engine version
  - DR0 status
  - L1 Cache status (CR0)
- PMAP.EXE Utility
- DBSWITCH.EXE Utility

Amber Debug Module

- An add-on debug card with engine on it
- Very few ROM space needed (128bytes)
- No memory needed for trace
- Plug to enable debugging, unplug to disable debugging. Even useful for shipped BIOS
- On-board UART, I/O mode or memory mapped registers, no any resource required, no any hook routines needed
- Debug MCD/PnP easily
- Use INSTALL_DEBUG_CARD instead of INSTALL_DEBUG
- INSTALL_DEBUG_CARD debugCardSegment, UartEnable (optional)
- Card jumper settings
- ROM/RAM Engine, ISA/PCI slots
- On-board port 80h (4 digits for PCI)

Tips

- Trace during very early POST (even 02):
  - Power up platform and break at 2A or above
  - Change break point to 02 or what you want
  - Restart the debugger (press Ctrl-R or click at restart button)
- Skip a POST routine (SG/SGN)
- Break limitation
- Two continuous F7 force a break without setting break code

That's All For Today

- Questions?
- Suggestions?