

TRACE32 Directory

TRACE32 Directory

TRACE32 Online Help	
TRACE32 Index	
TRACE32 Directory	1

Directory and Index

About the TRACE32 Online Help	(main.pdf)	1

TRACE32 Terminology

TRACE32 Terminology	(trace32_terms.pdf)	1
History		3
TRACE32 Online Help Terminology		4
Common Terms and Definitions		5
Abbreviations		7
TRACE32 Product Story		9
Power Debug Modules		9
Trace Modules		10

T32Start

T32Start	(app_t32start.pdf)	1
History		4
Introduction		5
Features		5
Quick Start		7
T32Start User Interface		8
Buttons		8
Context Sensitive Menu		9
Mouse Actions		11
Configuration Tree: Settings		12
Global Settings		12
Default Advanced Settings		13
Configuration Container and Configuration		20
Podbus Device Chain		20
MicroTrace		26
Software-only Debugging (Host MCI)		28
References to Tree Items		35
Configuration Examples		36
Hardware-based TRACE32 Tools		36
TRACE32 Software-only Tools		53
Command Line Arguments		56
Error Messages		57

Establish Your Debug Session

Establish Your Debug Session	(tutor_setup.pdf)	1
History		4
Establish your Debug Session		5
Key TRACE32 Setup Commands		5
Debug Scenarios		9
Establish the Debug Communication		11
Debug Scenario 1		17

Onchip/NOR Flash Programming	18
Configure the TRACE32 OS Awareness	35
Debug Scenario 2	36
Typical Boot Sequence	36
Flash Programming (NAND/Serial/eMMC)	41
Establish the Communication	55
Load the Debug Symbols	55
Debug Scenario 3	56
Run the Boot Loader	57
Load Application (and/or OS) Code and Debug Symbols	58
Load Debug Symbols only	58
Configure the TRACE32 OS Awareness	58
Complete Setup Example	58
Debug Scenario 4	59
Write a Script to Configure the Target	60
Load Application (and/or OS) Code and Debug Symbols	60
Configure the TRACE32 OS Awareness	60
Start-Up Scripts	61
Write a Start-Up Script	61
Run a Start-up Script	62
Automated Start-up Scripts	62

Debugger Tutorial

Debugger Tutorial (debugger_tutorial.pdf)	1
History	4
About the Tutorial	4
Connect the TRACE32 Hardware	5
Start TRACE32 PowerView	5
Set Up the Debug Environment	8
A Typical Setup Procedure	9
Start-Up Scripts	14
Write a Start-Up Script	14
Run a Start-up Script	15
Automated Start-up Scripts	15
User Interface - TRACE32 PowerView	16
TRACE32 Command Line and Softkeys	18
Window Captions - What Makes Them Special in TRACE32?	19

Debugging the Program	20
Basic Debug Commands	20
Debug Modes	21
Displaying the Stack Frame	23
Breakpoints	24
Setting Breakpoints	24
Setting Read/Write Breakpoints	25
Listing all Breakpoints	26
Variables	27
Displaying Variables	27
Displaying Variables of the Current Program Context	28
Using the Symbol Browser	28
Formatting Variables	29
Modifying Variables	30
Memory	31
Displaying Memory	31
Modifying Memory	32
Peripheral View	33
Store Window Configuration	34
Getting Online Help	35
Contacting the Lauterbach Support	37

Trace Tutorial

Trace Tutorial (trace_tutorial.pdf)	1
History	3
About the Tutorial	3
What is Trace?	3
Trace Use Cases	4
Trace Methods	5
Simulator Demo	6
Trace Configuration	7
Trace Recording	8
Displaying the Trace Results	10
Trace List	10
Displaying Function Run-Times	13
Variable Display	17
Track Option	18

Searching Trace Results	19
Trace Save and Load	20

TRACE32 Concepts

TRACE32 Concepts (trace32_concepts.pdf) 1

History 5

Access Classes 6

Access Class Expansion 12

Address Spaces 13

 Zones 13

 Zone Spaces 14

 MMU Space 15

 Machine Spaces 16

Address Types 17

 Absolute Physical Address 18

 Guest Logical Address 18

 Host Logical Address 19

 Intermediate Address (synonym: guest physical address) 19

 Logical Address (synonyms: virtual address, effective address) 19

 Physical Address (synonym: real address) 20

Awareness 21

 Hypervisor Awareness 21

 OS Awareness 21

Build Path 22

Chip Timestamp 23

Common Address Range 23

Cycle-accurate Tracing 23

CombiProbe 24

Extension 24

Hypervisor 24

Machine ID 25

Machine 27

 Guest Machine (synonym: virtual machine, VM) 27

 Host Machine 27

Magic Number 28

 Machine Magic Number 28

Space Magic Number	28
Task Magic Number	29
MCDS in Infineon TriCore AURIX MCUs	29
Memory Management Unit (MMU)	30
Multicore Debugging	30
Multiprocessor Debugging	30
Observability Gap	30
Order of Source Code Lines	31
OS-aware Debugging	33
OS (No Dynamic Memory Management)	33
AUTOSAR/OSEK Operating Systems	34
OS+MMU (Dynamic Memory Management)	34
OS-aware Tracing	35
Task Switch by Tracing Special Write Accesses	36
Task Switch by Tracing Task Switch Packets	39
Process	43
RTOS	43
Run-time Memory Access	44
Sample-based Profiling	52
Space ID	53
Spot Breakpoint	54
StopAndGo Mode	56
Symmetrical Multi-Processing (SMP)	57
Task	57
Thread	57
TRACE32 Virtual Memory	58
Trace Errors	61
TARGET FIFO OVERFLOW	61
FLOWERROR	62
Trace Sources	65
Tool Timestamp	66
VCPU	66

PowerView User's Guide

PowerView User's Guide	(ide_user.pdf)	1
History		6
Structure and Contents of the Documentation		7
Online Documentation		7
In-Circuit Debugger TRACE32-ICD		8
Program Start		10
In-Circuit Debugger TRACE32-ICD		10
Shut Down PowerView		11
Interactive Connection Mode		12
Interactive Connection on PowerView Startup		13
Connection Wizard		13
Connection Select Dialog		14
Automatic Connection upon Startup		16
Connection Script Generation for Test Automation		16
Connection Configuration Window		17
Reboot into Interactive Connection Mode		17
PowerView - Screen Display		18
Concept		18
Graphical User Interface - Window Modes		18
Main Menu Bar		21
Accelerators		21
Main Toolbar		22
Work Area		22
Message Line		23
Softkeys		24
State Line		25
Window Pages		30
Colors		31
How the TRACE32 PowerView GUI Assists You in Scripting		32
Commands		34
Command Structure		34
Long Form and Short Form of Commands and Functions		35
Entering Commands		36
Command History		38
Command and Function Parameters		39
General Command Parameter Parser		52

A. Object of Description	52
B. Support of C Language Expressions	54
C. Radix Mode Support	55
Operands	56
Operators	58
Window System	60
Windows	60
Window Operations	64
Text-based Functions	73
Selection Service	73
Message Windows	74
Window Tracking	75
File and Folder Operations	77
File Contents	78
Encrypt/Execute Encrypted Files	79
Host Commands	80
Printer Operations	81
System Setup and Configuration	83
Logging Commands	84
Dialog Programming	85
Dialog Syntax and File Types	85
Comments in Dialogs	87
Dialog Commands	88
Dialog Elements	89
Return Values and Labels	91
PRACTICE Macros inside Dialog Definitions	92
HELP System	93
Ways to Get Help	93
Context-Sensitive Help	94
Structure of the Help System	95
Configure the Help System	96
Recommendations for Choosing a PDF Viewer	97
Bookmarks for Help Topics	98
Troubleshooting the Help System	100
Change the Installation Path of the PDF Files	101
Winhelp Compatibility	101
Previous Releases - HELP System	102
Previous Releases - HELP Installation and Setup	102
Previous Releases - Configuring an Alternate PDF Viewer	102

Previous Releases - HELP Installation Problems	106
InterCom	108
Version Management and Licensing	110
Text Editors	111
Built-in Editors	111
Special Purpose Editor Windows	113
Edit Menu	116
External Editors	116
Configuring an External Editor	117
Working with TRACE32 and the External Editor	118
Icons	119
Built-in Icons and Icon Library	120
Inserting a Placeholder for User-Defined Icons	121
Drawing Icons	122
Interface	124
Shortcuts	125
Appendix - About the TRACE32 Software Version Numbers	129

PowerView Command Reference

PowerView Command Reference	(ide_ref.pdf)	1
History		15
AREA		17
AREA	Message windows	17
AREA.CLEAR	Clear area	18
AREA.CLOSE	Close output file	18
AREA.Create	Create or modify message area	19
AREA.Delete	Delete message area	19
AREA.List	Display a detailed list off all message areas	20
AREA.OPEN	Open output file	22
AREA.PIPE	Redirect area to pipe	23
AREA.RESet	Reset areas	23
AREA.SAVE	Save AREA window contents to file	24
AREA.Select	Select area	24
AREA.STDERR	Redirect area to stderr	25
AREA.STDOUT	Redirect area to stdout	25
AREA.view	Display message area in AREA window	26
AutoSTOre		28
AutoSTOre	Save and restore settings (history, GUI, etc.) automatically	28
BITMAPEDIT		30

BITMAPEDIT	Bitmap editor for user-defined icons	30
ChDir		31
ChDir	Change directory	31
ClipSTOre		32
ClipSTOre	Store settings to clipboard	32
CmdPOS		33
CmdPOS	Controls the position of TRACE32 in MWI window mode	33
CommandLineKEYS		35
CommandLineKEYS	Special characters	35
ComPare		36
ComPare	Compare files	36
CONNECTION		38
CONNECTION.AUTOSTART.MODE	Set connection autostart mode	39
CONNECTION.AUTOSTART.state	Connection autostart settings	39
CONNECTION.CADI.LibraryPath	Set CADI library path	40
CONNECTION.CONFIG.AutoSave	Store settings in preferences	40
CONNECTION.CONFIG.Load	Load settings from user preferences	41
CONNECTION.CONFIG.LoadDEFaults	Load default settings	41
CONNECTION.CONFIG.RESet	Reset settings	41
CONNECTION.CONFIG.SAVE	Save settings to user preferences	42
CONNECTION.ESTABLISH	Establish connection	42
CONNECTION.ETHernet.DeviceList.ADD	Add ethernet device to list	45
CONNECTION.ETHernet.DeviceList.CLEAR	Clear device list	45
CONNECTION.ETHernet.scan	Enable search for ethernet debug modules	45
CONNECTION.FindDEVices	Find TRACE32 debug hardware	45
CONNECTION.GDI.LibraryPath	Set GDI library path	46
CONNECTION.HOSTMCI.Host	Set external MCI server host	46
CONNECTION.HOSTMCI.ServerMode	Select HostMCI server mode	46
CONNECTION.IRIS.LibraryPath	Set IRIS library path	47
CONNECTION.MCD.LibraryPath	Set MCD library path	47
CONNECTION.ResetDevice	Reset debug module	47
CONNECTION.SELect.DEVice	Select TRACE32 hardware for connection	47
CONNECTION.SELect.INTERface	Select interface for connection	48
CONNECTION.SELect.state	Open connection selection window	49
CONNECTION.SIMUlator.License	Select license source for Instruction Set Simulator	49
CONNECTION.STARTUP	Initiate interactive connection process	49
CONNECTION.state	Open CONNECTION main configuration window	49
CONNECTION.TCPUSB.Host	Set TCPUSB host name and port	50
CONNECTION.TCPUSB.scan	Enable search for USB debug modules connected to a remote PC	50
CONNECTION.USB.scan	Enable search for USB debug modules	51

CONNECTION.VIEWER.License	Select license source for viewer	51
CONNECTION.Wizard.autostart	Create autostart connection script	51
CONNECTION.Wizard.CreateScript	Create custom connection script	52
CONNECTION.Wizard.Startup	Connection wizard for PowerView startup	52
COPY		53
COPY	Copy files	53
DATE		54
DATE	Display date and time	54
DEL		55
DEL	Delete file	55
DIALOG		56
DIALOG	Custom dialogs	56
Dialog Definition Programming Commands		56
DIALOG.AREA	Adds an output area to a custom dialog	98
DIALOG.DIR	Display a folder picker dialog	99
DIALOG.Disable	Disable dialog elements	100
DIALOG.Enable	Enable dialog elements	101
DIALOG.END	Close the dialog window	101
DIALOG.EXecute	Execute a dialog button	101
DIALOG.File	Pass file name from OS file dialog to PRACTICE script	102
DIALOG.File.open	Display an OS file-open dialog	103
DIALOG.File.SAVE	Display an OS file-save dialog	104
DIALOG.File.SELECT	Display an OS file-select dialog	105
DIALOG.MESSAGE	Create dialog box with an information icon	106
DIALOG.NOYES	Create dialog box with NO and YES buttons	106
DIALOG.OK	Create dialog box with an exclamation mark	107
DIALOG.Program	Interactive programming	108
DIALOG.ReProgram	Dialog programming	110
DIALOG.SELect	Programmatically focus on this dialog	110
DIALOG.Set	Modify the value of a dialog element	111
DIALOG.SetDIR	Browse for folder	113
DIALOG.SetFile	Pass file name from OS file dialog to custom dialog	114
DIALOG.SetFile.open	OS file-open dialog > file name > EDIT element	114
DIALOG.SetFile.SAVE	OS file-save dialog > file name > EDIT element	116
DIALOG.SetFile.SELECT	OS file-select dialog > file name > EDIT element	116
DIALOG.STORAGE	Stored macros in the dialog context	117
DIALOG.STORAGE.define	Define macros stored in the dialog context	117
DIALOG.STORAGE.LOAD	Load macros stored in the dialog context	118
DIALOG.STORAGE.SAVE	Update macros stored in the dialog context	118
DIALOG.view	Show dialog window	118
DIALOG.YESNO	Create dialog box with YES and NO buttons	120
DIR		121

DIR	List subdirectories and files	121
DUMP		123
DUMP	Binary file dump	123
EDIT		125
EDIT	TRACE32 editor	125
Overview EDIT		125
EDIT.CLOSE	Close a text file	126
EDIT.ENCoding	Change the file encoding	127
EDIT.EXTErn	Use specified external ASCII editor to edit file	128
EDIT.file	Edit file	129
EDIT.Find	Perform find, replace and goto operations in TRACE32 editors	132
EDIT.FORMAT	Format file contents an editor window	134
EDIT.Goto	Go to specified line	135
EDIT.InsertText	Insert text	136
EDIT.List	List editor files	136
EDIT.LOAD	Load text files	137
EDIT.OPEN	Use TRACE32 editor to edit file	138
EDIT.QUIT	Discard modifications	139
EDIT.REDO	Redo the previously undone edit/edits	139
EDIT.Replace	Open dialog window on the Replace tab	140
EDIT.REVERT	Revert file	140
EDIT.SAVE	Save a text file	141
EDIT.SElect	Select text/code in an editor window	142
EDIT.UNDO	Undo the last edit/edits	143
ERROR		144
ERROR.RESet	Reset PRACTICE error	144
EVAL		145
Eval	Evaluate expression	145
FIND		147
FIND	Search in file	147
FramePOS		148
FramePOS	Controls the position of TRACE32 in MDI window mode	148
HELP		151
HELP	Online help	151
HELP.Bookmark	Show help bookmark list	152
HELP.Bookmark.ADD	Files on bookmark list	153
HELP.Bookmark.ADD.file	Add file to bookmark list	153
HELP.Bookmark.ADD.Find	Add file to bookmark list	154
HELP.Bookmark.ADD.Index	Add file to bookmark list	155
HELP.Bookmark.DElete	Delete from bookmark list	155
HELP.Bookmark.show	Show help bookmark list	156

HELP.checkUPDATE	Automatic update check for new help-files	156
HELP.command	Command related support	156
HELP.FILTER	Filters for online help	157
HELP.FILTER.Add	Add a filter to the help filter list	158
HELP.FILTER.Delete	Delete filter from help filter list	158
HELP.FILTER.List	List all help filters	159
HELP.FILTER.RESet	Reset help filter system	159
HELP.FILTER.set	Activate/deactivate help filters for online help	160
HELP.Find	Perform a full-text search in online help	160
HELP.Index	Search in indexed terms, commands, and functions	163
HELP.OPEN	Open PDF documentation for command or function	165
HELP.PDF	Open PDF file	166
HELP.PICK	Context-sensitive help	166
HELP.PRinT	Print help files	167
HELP.PRinT.PRinTsel	Print selected files	167
HELP.PRinT.SELect	Select files to print	167
HELP.PRinT.show	Show print help files	168
HELP.PRinT.UNSELect	Unselect all print files	168
HELP.Topics	Show the structure of the online help system	169
HELP.TREE	Display command tree	170
HISTory		171
HISTory	Command history of last executed commands	171
HISTory.eXecute	Execute command history	172
HISTory.SAVE	Store command history log	172
HISTory.Set	History settings	173
HISTory.SIZE	Command history and file history	174
HISTory.SIZE.cmd	Define log size of command history	174
HISTory.SIZE.FILE	Define number of recently used files in 'File' menu	175
HISTory.type	Display command history log of last executed commands	175
IFCONFIG		176
IFCONFIG	Ethernet or USB communication	176
IFCONFIG.PROfile	Display operation profiles	176
IFCONFIG.state	Interface configuration	178
IFCONFIG.TEST	Test interface function and speed	180
InterCom		181
InterCom	Data exchange between different TRACE32 PowerView instances	181
InterCom.ENABLE	User-defined InterCom name, auto-assigned port number	182
InterCom.Evaluate	Evaluate function via InterCom system	184
InterCom.execute	Execute command via InterCom system	185
InterCom.executeNoWait	Execute command via InterCom system	187
InterCom.NAME	Assign user-defined InterCom name	187
InterCom.PING	Test InterCom system	189

InterCom.PipeCLOSE	Close named pipe	189
InterCom.PipeOPEN	Open named pipe	190
InterCom.PipeREAD	Read from named pipe	190
InterCom.PipeWRITE	Write to named pipe	191
InterCom.PORT	Assign user-defined InterCom UDP port number	191
InterCom.WAIT	Wait for remote InterCom system	193
LICENSE		194
LICENSE	Manage TRACE32 licenses	194
LICENSE.List	Display all license information	194
LICENSE.REQuest	Request a license	195
LICENSE.state	Display the currently used maintenance contract	197
LICENSE.UPDATE	Update the maintenance contract	198
LOG		199
LOG	Log TRACE32 commands and PRACTICE script calls	199
LOG.CLOSE	Close command log	200
LOG.DO	Log calls of PRACTICE scripts	200
LOG.OFF	Switch off command log	201
LOG.ON	Switch on command log	202
LOG.OPEN	Open command log file	202
LOG.toAREA	Log commands by writing them to an AREA window	204
LOG.type	Display command log	208
LS		208
LS	Display directory	208
MENU		209
MENU	Customize the user interface TRACE32 PowerView	209
MENU.AddMenu	Add one standard menu item	209
MENU.AddTool	Add a button to the main toolbar	210
MENU.Delete	Delete nested menu	211
MENU.Delete.NAME	Delete specified menu	211
MENU.PENDIng	Menu files waiting for compilation	212
MENU.PENDIng.List	List menu files waiting for compilation	212
MENU.PENDIng.RESet	Clear list of pending menu files	212
MENU.Program	Interactive programming	213
MENU.ReProgram	Menu programming	214
MENU.RESet	Default configuration	217
Programming Commands		218
MKDIR		236
MKDIR	Create new directory	236
MKTEMP	Create file or directory with unique name	237
MV		240
MV	Rename file	240

OS		241
OS	Execute host commands	241
Overview OS		241
OS.Area	Re-route host command output to AREA window	244
OS.Command	Execute a host command	245
OS.Hidden	Execute a host command in silent mode	247
OS.OPEN	Open file in default application	248
OS.screen	Call up the shell or execute host command	250
OS.SetENV	Set operating system environment variables	251
OS.Window	Re-route host command output to the OS.Window	252
PACK		253
PACK	Compress files (with LZW algorithm)	253
PATCH		254
PATCH	Binary file patching	254
PATH		255
PATH	Define search paths for files used by TRACE32 commands	255
PATH	Search path	256
PATH.Delete	Delete search path	256
PATH.DOWN	Define search path at end of list	257
PATH.List	List search path	258
PATH.RESet	Reset search path	258
PATH.Set	Define search path	259
PATH.UP	Define search path at top of list	260
PRinTer		261
PRinTer	Print and export window contents	261
PRinTer.Area	Re-route printer output to AREA window in specified format	262
PRinTer.ClipBoard	Re-route printer output to clipboard in specified format	263
PRinTer.CLOSE	Close file after multiple printer outputs	263
PRinTer.CONFIG	Print-out configuration	264
PRinTer.CONFIG.HEADER	Print window title	264
PRinTer.CONFIG.OFFSET	Specify print-out borders	264
PRinTer.CONFIG.SIZE	Specify print-out size	265
PRinTer.EXPORT	Export formatted printer output to file	266
PRinTer.FILE	Re-route printer output to a file in specified file format	270
PRinTer.FileType	Select file format	273
PRinTer.HardCopy	Make a hardcopy of the screen	274
PRinTer.OFFSET	Specify print-out borders	274
PRinTer.OPEN	Re-route multiple printer outputs to the same file	275
PRinTer.PRINT	Print to opened printer file	277
PRinTer.select	Select printer	278
PRinTer.SIZE	Specify print-out size	279
PWD		280

PWD	Change directory	280
PYthon		281
PYthon.EDIT	Open Python script in editor	281
PYthon.INSTALL	Install RCL module and Python interpreter	281
PYthon.RUN	Run Python script in dedicated window	282
QUIT		283
QUIT	Return to operating system	283
REN		284
REN	Rename file	284
RM, RMDIR		285
RM	Delete file	285
RMDIR	Remove directory	285
SCreenShot		286
SCreenShot	Save a screenshot of a window to a file	286
SETUP		288
SETUP	Setup commands	288
SETUP.API.GDB.OFF	Disable GDB server	288
SETUP.API.GDB.ON	Disable GDB server	289
SETUP.API.TCF.OFF	Disable TCF agent	289
SETUP.API.TCF.ON	Enable TCF agent	289
SETUP.API.TCF.PORT	Set TCF port	289
SETUP.ASCIITEXT	Configure ASCII text display	291
SETUP.BAKfile	Enable backup file creation	293
SETUP.COLOR	Change colors	294
SETUP.DEVNAME	Set logical device name	295
SETUP.EDITEXT	Define an external editor	296
SETUP.EDITOR	TRACE32 editor configuration	298
SETUP.EDITOR.AutoSuggest	Show input suggestions while typing	299
SETUP.EDITOR.BAKfile	Make backup copy when file is saved	300
SETUP.EDITOR.HighLight	Control syntax highlighting	300
SETUP.EDITOR.Indentation	Select indentation method	301
SETUP.EDITOR.IndentSize	Set indentation size	302
SETUP.EDITOR.IndentWithTabs	Use tabulator for indentation	303
SETUP.EDITOR.Mode	Show visible whitespace or ASCII view	303
SETUP.EDITOR.SaveChangesPrompt	Save file if edit window closed	304
SETUP.EDITOR.SmartBackspace	Backspace maintains indentation	305
SETUP.EDITOR.SmartCursor	Control cursor movement	305
SETUP.EDITOR.SmartFormat	Automatic formatting	306
SETUP.EDITOR.state	Show editor configuration dialog	307
SETUP.EDITOR.TabSize	Set tabulator size	308
SETUP.EDITOR.TrailingWhitespace	Remove trailing whitespace	308

SETUP.EDITOR.TYPE	Set editor implementation	309
SETUP.EXTension	Set default file name extensions	310
SETUP.FASTRESPONSE	Optimize for fast response times	310
SETUP.FILETYPE	File type configuration	311
SETUP.FILETYPE.DropCoMmanD	Set command for dropped files	311
SETUP.FILETYPE.ENCoding	Set encoding mode	312
SETUP.FILETYPE.EXTension	Set default file name extensions	314
SETUP.HOLDDIR	Configure working directory	317
SETUP.ICONs	Display icons in popup menus	317
SETUP.InterComACKTIMEOUT	Sets the InterCom acknowledge timeout	318
SETUP.PDEBUG	PRACTICE debug configuration settings dialog	319
SETUP.PDEBUG.BlockClose	Block window closing commands	320
SETUP.PDEBUG.BlockPosition	Block window positioning commands	320
SETUP.PDEBUG.MacroRESet	Reset PRACTICE macros after ending script	320
SETUP.PDEBUG.RESet	Reset settings to default values	321
SETUP.PDEBUG.ScriptParams	Set PRACTICE debug script parameters	321
SETUP.PDEBUG.TermScripts	Terminate all pending PRACTICE scripts	321
SETUP.PDEBUG.WindowExternal	Open debug window as external window	322
SETUP.PDEBUG.WindowOnTop	Keep debug window on top	322
SETUP.PDFViewer	Context-sensitive help via your favorite PDF viewer	323
SETUP.PDFViewer.EXEcutable	Path and executable of your PDF viewer	324
SETUP.PDFViewer.OPEN	Open a PDF of the help system	324
SETUP.PDFViewer.PRinT	Print PDF via HELP window	325
SETUP.PDFViewer.RESet	Reset the settings in SETUP.PDFViewer dialog	325
SETUP.PDFViewer.TEMPorary	Help configuration for demo purposes	326
SETUP.PDFViewer.TEMPorary.EXEcutable	PDF viewer for demo purposes	326
SETUP.PDFViewer.TEMPorary.OPEN	Open a PDF of the help system	326
SETUP.PDFViewer.TEMPorary.PRinT	Print PDF via HELP window	327
SETUP.PDFViewer.TEMPorary.RESet	Reset demo-help configuration	327
SETUP.PYthon.EXEcutable	Defines path to python interpreter	327
SETUP.QUITDO	Define quit PRACTICE script file	328
SETUP.RADIX	Radix mode	329
SETUP.RANDOM	Set seed for RANDOM() function	330
SETUP.ReDraw	Update whole screen	330
SETUP.RESOLVEDIR	Resolve symbolic links	331
SETUP.SOUND	Set sound generator mode	331
SETUP.STOPMESSAGE	Print message when STOP command is executed	331
SETUP.STOre	Configure output of the STOre commands	333
SETUP.TabSize	Configure tab width	334
SETUP.TIMEFORM	Select scientific time format	335
SETUP.UpdateRATE	Update rate for windows	336
SETUP.WARNSTOP	Configure PRACTICE stops	336
SETUP.XSLTSTYLESHEET	Reference to XSLT stylesheet for XML files	337

SHA1SUM	338
SHA1SUM	Calculate SHA1 checksum of a file 338
SILENT	339
SILENT	Suppress informational messages in AREA window 339
SOFTKEYS	340
SOFTKEYS	Toggle the buttons on the softkey bar 340
STATUSBAR	341
STATUSBAR	Toggle state line 341
STOre	342
STOre	Store settings as PRACTICE script 342
SUBTITLE	343
SUBTITLE	Define a window subtitle for AMP debugging 343
TAR	345
TAR	Pack files into an archive 345
TIMEOUT	347
TIMEOUT	Specify timeout for TRACE32 command 347
TITLE	348
TITLE	Define a main window title for a TRACE32 PowerView GUI 348
TOOLBAR	349
TOOLBAR	Toggle toolbar 349
TYPE	350
TYPE	Display text file 350
UNARchive	351
UNARchive	Linux and Microsoft libraries 351
UNARchive.extract	Extract files from Linux library and Microsoft library 351
UNARchive.Show	Extract files from library and list them in window 352
UNARchive.Table	Display table of contents of library 352
UNPACK	353
UNPACK	Expand files (with LZW algorithm) 353
UNZIP	354
UNZIP	Expand GZIP archive file (with DEFLATE algorithm) 354
VERSION	355
VERSION	TRACE32 version information 355
VERSION.ENVironment	Display environment settings 355
VERSION.HARDWARE	Display hardware versions 356
VERSION.SOFTWARE	Display software versions 356
VERSION.ThirdPartyLicenses	Display third party license information 357
VERSION.view	Display window with version info 358

WELCOME		359
WELCOME	Welcome to TRACE32	359
WELCOME.CONFIG	Configure search paths for PRACTICE demo scripts	359
WELCOME.CONFIG.ADDDIR	Add a new script search path	360
WELCOME.CONFIG.FILTER	Set the script search filter	360
WELCOME.CONFIG.ReMoveDIR	Remove a script search path	360
WELCOME.CONFIG.RESet	Reset the script search configuration	360
WELCOME.CONFIG.state	Open the welcome config window	361
WELCOME.SCRIPTS	Open the script search window	362
WELCOME.STARTUP	Open the welcome window if not disabled	362
WELCOME.view	Open the welcome window	363
Window		364
Win	Window handling (size, position, font size, etc.)	364
WinBack	Generate background window	365
WinCLEAR	Erase windows	365
WinDEFaultSIZE	Apply a user-defined default size to windows	367
WinDuplicate	Allows to open an existing window again	368
WinExt	Generate external window	369
WinFIND	Search for text in window	369
WinFreeze	Generate frozen window	371
WinLarge	Generate window with large font	372
WinMid	Generate window with regular font	373
WinOverlay	Pile up windows on top of each other	373
WinPAGE	Window pages	374
WinPAGE.Create	Create and select page	374
WinPAGE.Delete	Delete page	375
WinPAGE.List	Display an overview of all pages and their windows	376
WinPAGE.REName	Rename page	377
WinPAGE.RESet	Reset window system	377
WinPAGE.select	Select page	377
WinPAN	Specify a window cut-out	378
WinPOS	Define window dimensions and window name	379
WinPrint	Print address or record range of a window	382
WinPRT	Hardcopy of window	382
WinResist	Generate a resistant window	384
WinRESIZE	New size for window	385
WinSmall	Generate window with small font	386
WinTABS	Specify widths of re-sizable columns	386
WinTOP	Bring window to top	387
WinTrans	Generate transparent window	388
ZERO		389
ZERO.offset	Set time reference	389
ZERO.RESet	Reset to original value	390

ZIP	390
ZIP	Compress files to GZIP archive (with DEFLATE algorithm) 390
Appendix A - Help Filters	391

PRACTICE Script Language

PRACTICE Script Language User's Guide

PRACTICE Script Language User's Guide	(practice_user.pdf) 1
History	3
Why Use PRACTICE Scripts	3
Related Documents	3
PRACTICE Script Structure	4
Function	4
Difference between Variables and PRACTICE Macros	4
PRACTICE Script Elements	5
Script Flow	6
Conditional Script Flow	7
Script Nesting	7
Block Structures	8
PRACTICE Macros	9
Switching PRACTICE Macro Expansion ON or OFF	11
Parameter Passing	12
Input and Output	13
File Operations	14
Automatic Start-up Scripts	15
Logging the Call Hierarchy of PRACTICE Scripts	17
Debugging PRACTICE Scripts	19
Appendix A	21
How to Run Demo Scripts Copied from the PDF Manuals	21
Demo Scripts in the TRACE32 Demo Folder	24

PRACTICE Script Language Reference Guide

PRACTICE Script Language Reference Guide	(practice_ref.pdf) 1
History	5
Related Documents	6
A ... D	7

APPEND	Append to file	7
BEEP	Acoustic signal	7
CLOSE	Close file	8
CONTInue	Continue PRACTICE script	8
DECRYPT	Decrypts a text or binary file	9
DO	Start PRACTICE script	10
DODECRYPT	Execute encrypted PRACTICE script (*.cmm)	11
E ... F		12
ECHO	Write text and data to an AREA window (with format decoration)	12
ELSE	Conditional script execution	14
ENCRYPT	Encrypt a text or binary file	15
ENCRYPTDO	Encrypt a PRACTICE script (*.cmm)	16
ENCRYPTPER	Encrypt a PER file (*.per)	17
END	Terminate PRACTICE scripts, etc.	18
ENDDO	Return from a PRACTICE script	19
ENTER	Window-based input	20
ENTRY	Parameter passing	22
G ... H		23
GLOBAL	Create global PRACTICE macro	23
GLOBALON	Global event-controlled PRACTICE script execution	24
GOSUB	Subroutine call	30
GOTO	Local script jump	32
I ... L		33
IF	Conditional script execution	33
INKEY	Character input	34
JUMPTO	Global script jump	34
LOCAL	Create local PRACTICE macro	35
M ... O		37
ON	Event-controlled PRACTICE script execution	37
OPEN	Open data file	42
P ...		43
PARAMETERS	Parameter fetching	43
PBREAK	Breakpoints in PRACTICE script files (*.cmm)	44
PBREAK.Delete	Delete breakpoint	44
PBREAK.DISable	Disable breakpoint	44
PBREAK.ENable	Enable breakpoint	45
PBREAK.List	Display breakpoint list	46
PBREAK.OFF	TRACE32 disables breakpoint handling	47
PBREAK.ON	TRACE32 re-enables breakpoint handling	48
PBREAK.RESet	Clear all breakpoints	48
PBREAK.Set	Add breakpoint	49
PEDIT	Edit PRACTICE script	52

PLIST	List PRACTICE script	55
PMACRO	PRACTICE macros	56
PMACRO.EXPLICIT	Enforce explicit PRACTICE macro declaration	56
PMACRO.IMPLICIT	Implicit PRACTICE macro declaration	58
PMACRO.IMPLICITPRIVATE	Hide implicit macros	59
PMACRO.list	Display PRACTICE macros	60
PMACRO.LOCK	Lock PRACTICE macros	60
PMACRO.RESet	Clear current PRACTICE macros	61
PMACRO.UNLOCK	Unlock PRACTICE macros	62
PRINT	Write text and data to an AREA window (without format decoration)	63
PRINTF	Write formatted data to an AREA window	67
PRIVATE	Create private PRACTICE macro	75
PSKIP	Skip command or block in PRACTICE script	77
PSTEP	Execute single line	78
PSTEPOUT	Back to caller	79
PSTEPOVER	Step over callee and stop at the next script line	80
Q ... R		81
READ	Read from data file	81
RePeaT	Loop with check at end of loop	82
RETURN	Return from subroutine	84
RETURNVALUES	Take return values	85
RUN	Start PRACTICE script	86
S ... T		87
SCREEN	Screen updating	87
SCREEN.ALways	Refresh always	87
SCREEN.display	Refresh screen	88
SCREEN.OFF	No refresh	88
SCREEN.ON	Refresh when printing	88
SCREEN.WAIT	Update screen while waiting	89
SPRINTF	Write formatted data to a PRACTICE macro	91
STOP	Interrupt PRACTICE script	92
SUBROUTINE	Define a subroutine	93
W ... Z		94
WAIT	Wait until a condition is true or a period has elapsed	94
WHILE	Loop with check at start of loop	96
WRITE	Write to data file	97
WRITEB	Write binary data to file	98

PowerView Function Reference

PowerView Function Reference	(ide_func.pdf)	1
History		10
In This Document		11
AREA Functions		12
In This Section		12
AREA.COUNT()	Number of existing message areas	12
AREA.EXIST()	Check if message area exists	12
AREA.LINE()	Extract line from message area	13
AREA.MAXCOUNT()	Maximum number of message areas	13
AREA.NAME()	Names of existing message areas	14
AREA.SELECTed()	Name of active message area	14
AREA.SIZE.COLUMNS()	Columns of a message area	15
AREA.SIZE.LINES()	Lines of a message area	15
CLOCK Functions		16
CONFIG Function		17
CONFIG.SCREEN()	Check if screen output is switched on	17
CONNECTION Functions		18
CONNECTION.DEVice.IndexByName()	Get device index	18
CONNECTION.DEVice.IndexBySerialNumber()	Get device index	18
CONNECTION.DEVice.InUse()	Debug module currently in use	18
CONNECTION.DEVice.NAME()	Get device name of debug module	19
CONNECTION.DEVice.Number()	Number of detected debug modules	19
CONNECTION.DEVice.PORT()	Get host connection port of debug module	20
CONNECTION.DEVice.SerialNumber()	Get device serial number	20
CONNECTION.GetDriverError()	Get driver error of last connection attempt	21
CONNECTION.HOSTMCI.TestMciServer()	Get MCI server state information	21
CONNECTION.INTERface.Available()	Check connection interface availability	21
CONNECTION.STATE.ERROR()	Failed to establish connection	22
CONNECTION.STATE.Interactive()	Interactive connection state	22
CONVert Functions		23
In This Section		23
CONVert.ADDRESSTODUALPORT()	Dualport access class	23
CONVert.ADDRESSTONONSECURE()	Non-secure access class	24
CONVert.ADDRESSTOSECURE()	Secure access class	24
CONVert.BOOLTOINT()	Boolean to integer	25
CONVert.CHAR()	Integer to ASCII character	25
CONVert.FLOATTOINT()	Float to integer	26

CONVert.HEXTOINT()	Hex to integer	26
CONVert.INTTOBOOL()	Integer to boolean	27
CONVert.INTTOFLOAT()	Integer to floating point value	27
CONVert.INTTOHEX()	Integer to hex	28
CONVert.INTTOMASK()	Compose bit-mask from integer value and mask	28
CONVert.LINEAR11TOFLOAT()	LINEAR11 to float	29
CONVert.LINEAR16TOFLOAT()	LINEAR16 to float	29
CONVert.MASKMTOINT()	Bits set to don't-care in given bit-mask	30
CONVert.MASKTOINT()	Bits set to 1 in given bit-mask	31
CONVert.OCTaltoint()	Octal to decimal	31
CONVert.SignedByte()	1 byte to 8 bytes	32
CONVert.SignedLong()	4 bytes to 8 bytes	32
CONVert.SignedWord()	2 bytes to 8 bytes	33
CONVert.TIMEMSTOINT()	Time to milliseconds	33
CONVert.TIMENSTOINT()	Time to nanoseconds	34
CONVert.TIMERAWTOINT()	Time to TRACE32 timer ticks	34
CONVert.TIMESTOINT()	Time to seconds	34
CONVert.TIMEUSTOINT()	Time to microseconds	35
CONVert.TOLOWER()	String to lower case	36
CONVert.TOUPPER()	String to upper case	36
DATE Functions		37
In This Section		37
DATE.DATE()	Current date	37
DATE.DAY()	Today's date	37
DATE.MakeUnixTime()	Date to Unix timestamp	38
DATE.MONTH()	Number of current month	39
DATE.SECONDS()	Seconds since midnight	39
DATE.TIME()	Current time	39
DATE.TimeZone()	Time zone identifier and hh:mm:ss	40
DATE.UnixTime()	Seconds since Jan 1970	40
DATE.UnixTimeUS()	Microseconds since Jan 1970	40
DATE.utcOffset()	Offset of current local time to UTC	41
DATE.YEAR()	Current year	41
DIALOG Functions		42
In This Section		42
DIALOG.BOOLEAN()	Current boolean value of checkbox	42
DIALOG.EXIST()	Existence of dialog element	43
DIALOG.STRING()	Current string value of dialog element, e.g. EDIT box	44
DIALOG.STRING2()	Comma-separated list of values, e.g. from LISTBOX	45
ERROR Functions		46
In This Section		46
ERROR.CMDLINE()	Erroneous command	46

ERROR.FIRSTID()	ID of first error	46
ERROR.ID()	ID of last error message	47
ERROR.MESSAGE()	Error text	48
ERROR.OCCURRED()	Error status	48
ERROR.POSITION()	Error position	48
EVAL Functions		49
In This Section		49
EVAL()	Value of expression evaluated with Eval command	49
EVAL.ADDRESS()	Address of expression evaluated with Eval cmd.	49
EVAL.BOOLEAN()	Boolean expression evaluated with Eval cmd. boolean	49
EVAL.FLOAT()	Float value of expression evaluated with Eval cmd.	50
EVAL.PARAM()	Expression evaluated with Eval cmd.	50
EVAL.STRing()	String composed by expression evaluated with Eval cmd.	50
EVAL.TIme()	Value of time evaluated with Eval command	50
EVAL.TYPE()	Type of expression evaluated with Eval command	51
FALSE Function		52
FALSE()	Boolean expression	52
FILE Functions		53
In This Section		53
__FILE__()	Path and file name of current PRACTICE script	53
__LINE__()	Number of script line to be executed next	53
FILE.EOF()	Check if end of read-in file has been reached	53
FILE.EOFLASTREAD()	Check if last read from file reached the end of the file	54
FILE.EXIST()	Check if file exists	55
FILE.NEWHANDLE()	Get next free handle	55
FILE.OPEN()	Check if file is open	56
FILE.SUM()	Get checksum from a file	56
FILE.TYPE()	File type of loaded file	56
FORMAT Functions		58
In This Section		58
FORMAT.BINary()	Numeric to binary value (leading spaces)	58
FORMAT.CHAR()	Numeric to ASCII character (fixed length)	59
FORMAT.Decimal()	Numeric to string (leading spaces)	60
FORMAT.DecimalU()	Numeric to unsigned decimal as string (leading spaces)	61
FORMAT.DecimalUZ()	Numeric to unsigned decimal as string (leading zeros)	62
FORMAT.FLOAT()	Floating point value to string	63
FORMAT.HEX()	Numeric to hex (leading zeros)	64
FORMAT.STRing()	Output string with fixed length	65
FORMAT.TIME()	Time to string (leading spaces)	66
FORMAT.UDECIMAL()	Refer to FORMAT.DecimalU()	66
FORMAT.UDECIMALZ()	Refer to FORMAT.DecimalUZ()	67
FORMAT.UnixTime()	Format Unix timestamps	67

FOUND Functions	71
In This Section	71
FOUND()	TRUE() if search item was found 71
FOUND.COUNT()	Number of occurrences found 72
GDB Function (TRACE32 as GDB Back-End)	73
GDB.PORT()	Port number for communication via GDB interface 73
HELP Function	74
HELP.MESSAGE()	Help search item 74
HOST Functions	75
HOSTID()	Host ID 75
HOSTIP()	Host IP address 75
IFCONFIG and IFTEST Functions	76
In This Section	76
IFCONFIG.COLLISIONS()	Collisions since start-up 76
IFCONFIG.CONFIGURATION()	Connection type 77
IFCONFIG.DEVICENAME()	Name of TRACE32 device 77
IFCONFIG.ERRORS()	Errors since start-up 77
IFCONFIG.ETHernetADDRESS()	MAC address of TRACE32 device 77
IFCONFIG.IPADDRESS()	IP address of TRACE32 device 78
IFCONFIG.RESYNCS()	Resyncs since start-up 78
IFCONFIG.RETRIES()	Retries since start-up 78
IFTEST.DOWNLOAD()	Download in KByte/sec 79
IFTEST.LATENCY()	Latency in microseconds 79
IFTEST.UPLOAD()	Upload in KByte/sec 79
InterCom Functions	80
In This Section	80
InterCom.GetGlobalMacro()	Exchange strings between PowerView instances 80
InterCom.GetPracticeState()	PRACTICE run-state on other instance 81
InterCom.NAME()	InterCom name of this TRACE32 instance 81
InterCom.PING()	Check if ping is successful 82
InterCom.PODPORT()	Port number of any TRACE32 instance 83
InterCom.PODPORTNAME()	InterCom name of any TRACE32 instance 84
InterCom.PODPORTNUMBER()	Number of TRACE32 instances 85
InterCom.PORT()	Port number of this TRACE32 instance 86
LICENSE Functions	87
In This Section	87
LICENSE.DATE()	Expiration date of maintenance contract 87
LICENSE.FAMILY()	Name of the CPU family license 87
LICENSE.FEATURES()	List of features licensed 88
LICENSE.getINDEX()	Index of maintenance contract 88
LICENSE.GRANTED()	License state 89

LICENSE.HAVEFEATURE()	Checks if license is stored in debugger hardware	89
LICENSE.MSERIAL()	Serial number of the maintenance contract	90
LICENSE.MULTICORE()	Check if multicore debugging is licensed	90
LICENSE.RequiredForCPU()	License required for selected CPU	90
LICENSE.SERIAL()	Serial number of debug cable	91
LOG Function		92
LOG.DO.FILE()	Get log file used by LOG.DO	92
Mathematical Functions		93
In This Section		93
math.ABS()	Absolute value of decimal value	93
math.FABS()	Absolute value of floating point number	93
math.FCOS()	Cosine of an angle given in radian	94
math.FEXP()	Exponentiation with base e (Euler's number)	94
math.FEXP10()	Exponentiation with base 10	94
math.FINF()	Positive infinity	94
math.FLOG()	Natural logarithm of given value	95
math.FLOG10()	Logarithm to base 10 of given value	95
math.FMAX()	Return the larger one of two floating point values	95
math.FMIN()	Return the smaller one of two floating point values	96
math.FMOD()	Floating-Point Modulus	96
math.FNAN()	Not a number value	97
math.FPOW()	Y-th power of base x	97
math.FSIN()	Sine of an angle given in radian	97
math.FSQRT()	Square-root of given value	97
math.MAX()	Return the larger one of two decimal values	98
math.MIN()	Return the smaller one of two decimal values	98
math.SIGN()	Return -1 or +1 depending on argument	98
math.SIGNUM()	Return -1 or 0 or +1 depending on argument	99
math.TimeMAX()	Return the larger one of two time values	99
math.TimeMIN()	Return the smaller one of two time values	99
MENU Function		101
MENU.EXIST()	Check if menu name exists	101
NODENAME Function		102
NODENAME()	Node name of connected TRACE32 device	102
OS Functions		103
In This Section		103
OS.DIR()	Check if directory exists	104
OS.DIR.ACCESS()	Access rights to directory	104
OS.ENV()	Value of OS environment variable	105
OS.FILE.readable()	Check if file can be opened for reading	106
OS.FILE.ABSPATH()	Absolute path to file or directory	106
OS.FILE.ACCESS()	Access rights to file	107

OS.FILE.BASENAME()	Strip directory and suffix from filenames	108
OS.FILE.DATE()	Modification date and timestamp of file	108
OS.FILE.DATE2()	Modification date of file	109
OS.FILE.EXIST()	Check if file exists	109
OS.FILE.EXTENSION()	File name extension	109
OS.FILE.JOINPATH()	Join multiple paths	110
OS.FILE.LINK()	Real file name of file link	111
OS.FILE.NAME()	Extract file name from path	112
OS.FILE.PATH()	Return path of file	113
OS.FILE.REALPATH()	Canonical absolute path to file or directory	113
OS.FILE.SIZE()	File size in bytes	114
OS.FILE.TIME()	Modification timestamp of file	114
OS.FILE.UnixTime()	Unix timestamp of file	115
OS.FIRSTFILE()	First file name matching a pattern	116
OS.ID()	User ID of TRACE32 instance	117
OS.NAME()	Basic name of operating system	118
OS.NEXTFILE()	Next file name matching a pattern	119
OS.PORTAVAILABLE.TCP()	Check if TCP port is used	119
OS.PORTAVAILABLE.UDP()	Check if UDP port is used	120
OS.PresentConfigurationFile()	Name of used TRACE32 configuration file	120
OS.PresentDemoDirectory()	Demo directory for the current architecture	121
OS.PresentExecutableDirectory()	Directory of current TRACE32 exe.	121
OS.PresentExecutableFile()	Path and file name of current TRACE32 exe.	121
OS.PresentHomeDirectory()	Path of the home directory	122
OS.PresentHELPDirectory()	Path of the TRACE32 online help directory	122
OS.PresentLicenseFile()	Current TRACE32 license file	122
OS.PresentPracticeDirectory()	Directory of currently executed script	123
OS.PresentPracticeFile()	Path and file name of currently executed script	123
OS.PresentSystemDirectory()	TRACE32 system directory	123
OS.PresentTemporaryDirectory()	TRACE32 temporary directory	124
OS.PresentWorkingDirectory()	Current working directory	124
OS.RETURN()	Return code of the last executed operating system command	125
OS.TIMER()	OS timer in milliseconds	125
OS.TMPFILE()	Name for a temporary file	125
OS.VERSION()	Type of the host operating system	127
OS.Window.LINE()	Get line from an OS.Window window	130
PATH Functions		131
In This Section		131
PATH.NUMBER()	Number of path entries	131
PATH.PATH()	Search path entry	131
ProcessID Function		133
ProcessID()	Process identifier of a TRACE32 PowerView instance	133

PRACTICE Functions	134
In This Section	134
PRACTICE.ARG() Return value of GOSUB, DO, RETURN, and ENDDO	134
PRACTICE.ARG.SIZE() Number of passed or returned arguments	135
PRACTICE.CALLER.FILE() File name of the script/subscript caller	136
PRACTICE.CALLER.LINE() Line number in caller script	136
PRACTICE.CoMmanD.AVAILable() Check if command is available	137
PRACTICE.FUNction.AVAILable() Check if function is available	138
PRINTER Function	139
PRINTER.FILENAME() Path and file name of next print operation	139
RADIX Function	140
RADIX() Current radix setting	140
RANDOM Functions	141
RANDOM() Pseudo random number	141
RANDOM.RANGE() Pseudo random number from specified range	141
RANDOM.RANGE.HEX() Pseudo hex random number from specified range	142
RCL Function	143
RCL.PORT() UDP Port number of remote API interface	143
RCL.TCP.NrUsedCons() Number of remote API clients connected via TCP	143
RCL.TCP.PORT() TCP Port number of remote API interface	144
SOFTWARE Functions	145
In This Section	145
SOFTWARE.64BIT() Check if TRACE32 executable is 64-bit	145
SOFTWARE.BUILD() Upper build number	145
SOFTWARE.BUILD.BASE() Lower build number	145
SOFTWARE.VERSION() Release build or nightly build, etc.	146
STRing Functions	147
In This Section	147
STRing.CHAR() Extract a character	147
STRing.ComPare() Check if string matches pattern	148
STRing.ComPare.Alpha() Compare strings by unicode	148
STRing.COUNT() Substring occurrences	149
STRing.CUT() Cut string from left or right	150
STRing.ESCAPEQuotes() Double quote character inside string	150
STRing.FIND() Check if search characters are found within string	151
STRing.LENgtH() Length of string	152
STRing.LoWeR() String to lowercase	152
STRing.MID() Extract part of string	152
STRing.ReplacE() Modified string after search operation	154
STRing.SCAN() Offset of the found string	155
STRing.SCANAndExtract() Extract remaining string after search string	156

STRing.SCANBack()	Offset of the found string	157
STRing.SPLIT()	Return element from string list	158
STRing.TOKEN()	Extract token from string	161
STRing.TRIM()	String without leading and trailing whitespaces	163
STRing.UPPeR()	String to uppercase	164
TCF Functions (TRACE32 as TCF Agent)		165
In This Section		165
TCF.PORT()	Port number of TCF interface	165
TCF.DISCOVERY()	Check if TCF discovery is enabled	165
TEST Function		166
TEST.TIMEISVALID()	Check if time value is valid	166
TIMEOUT Function		167
TIMEOUT()	Check if command was fully executed	167
TITLE Function		169
TITLE()	Caption of the TRACE32 main window	169
TRUE Function		170
TRUE()	Boolean expression	170
WARNINGS Function		170
WARNINGS()	Check if warning occurred during command execution	170
WINdow Functions		171
In This Section		171
WINdow.COMMAND()	Command string displayed in window	171
WINdow.EXIST()	Check if window name exists	172
WINdow.LIST()	Generate a comma-separated list of window names	172
WINdow.POSition()	Information on window position and dimension	173
WINPAGE.CURRENT()	Get name of currently selected window page	174
WINPAGE.EXIST()	Check if window page exists	174
WINPAGE.LIST()	Generate comma-separated list of page names	175

General Function Reference

General Function Reference	(general_func.pdf)	1
History		24
In This Document		26
How This Document is Organized		26
Difference between Functions and Commands in TRACE32		27
Purpose of Functions		28
How to Use Functions		29
Which Return Values of Functions can be Printed?		31
Related Documents		31

ACCESS Functions	32
In This Section	32
ACCESS.isGUEST()	TRUE if access class belongs to guest 32
ACCESS.isHYPERVISOR()	TRUE if access class belongs to hypervisor 33
ADDRESS Functions	34
In This Section	34
ADDRESS.ACCESS()	Access class as ordinal number 34
ADDRESS.ACCESS.CMP()	Compare access classes 34
ADDRESS.ACCESS.CMPSTRICT()	Compare access classes, strict 35
ADDRESS.EXPANDACCESS()	Fully qualified access class 35
ADDRESS.INSTR.LEN()	Length of instruction 36
ADDRESS.isDATA()	Check if memory class refers to data 36
ADDRESS.isGUEST()	TRUE if address is guest address 37
ADDRESS.isHYPERVISOR()	TRUE if address is hypervisor address 37
ADDRESS.isINTERMEDIATE()	Check if intermediate address 38
ADDRESS.isNONSECURE()	TRUE if non-secure (TrustZone) access 38
ADDRESS.isNONSECUREEX()	TRUE if non-secure access 39
ADDRESS.MACHINEID()	Extract machine ID 39
ADDRESS.MAU()	Minimal addressable unit size (MAU) 40
ADDRESS.OFFSET()	Address without class 41
ADDRESS.isONCHIP()	TRUE if on-chip address area 41
ADDRESS.isPHYSICAL()	TRUE if physical address 41
ADDRESS.isPROGRAM()	TRUE if program address 42
ADDRESS.isSECURE()	TRUE if secure (TrustZone) access 42
ADDRESS.isSECUREEX()	TRUE if secure access 43
ADDRESS.RANGE.BEGIN()	Lowest address value of address range 44
ADDRESS.RANGE.END()	Highest address value of address range 44
ADDRESS.RANGE.SIZE()	Size of address range 45
ADDRESS.SEGMENT()	Segment of an address 46
ADDRESS.STRACCESS()	Access class of an address 46
Analyzer Functions	47
In This Section	48
Analyzer()	Check if Analyzer command group is available 48
Analyzer.CONFIG.<powertrace>()	Check if specified PowerTrace connected 49
Analyzer.COUNTER.EVENT()	Get value of trigger program event counter 49
Analyzer.COUNTER.TIME()	Get value of trigger program time counter 50
Analyzer.DSEL()	For internal usage only 50
Analyzer.FIRST()	Get record number of first trace record 50
Analyzer.FLOW.ERRORS()	Get number of flow errors / hard errors 50
Analyzer.FLOW.FIFOFULL()	Get number of FIFO overflows 51
Analyzer.FOCUS.EYE()	Check quality of data eye 53
Analyzer.ISCHANNELUP()	Check if serial link is established 54
Analyzer.MAXSIZE()	Get max. size of trace buffer in records 55

Analyzer.MODE()	Get Analyzer recording mode	55
Analyzer.MODE.FLOW()	Check if Analyzer operates as flowtrace	55
Analyzer.PCIE.CONFIG()	Value of register field from PCIe configuration	56
Analyzer.PCIE.ISCONFIGURED()	TRUE if prerequisites are fulfilled	57
Analyzer.PCIE.Register()	Value of 32-bit register from PCIe configuration	57
Analyzer.PROBEREVISION()	Get revision of StarCore NEXUS probe	58
Analyzer.RECORDS()	Get number of used trace records	58
Analyzer.RECORD.ADDRESS()	Get address recorded in trace record	58
Analyzer.RECORD.DATA()	Get data recorded in trace record	59
Analyzer.RECORD.OFFSET()	Get address in trace record as number	59
Analyzer.RECORD.TIME()	Get timestamp of trace record	60
Analyzer.REF()	Get record number of reference record	61
Analyzer.SIZE()	Get current trace buffer size in records	61
Analyzer.STATE()	Get state of Analyzer	61
Analyzer.THRESHOLD()	Get threshold voltage of parallel preprocessor	62
Analyzer.TraceCONNECT()	Name of trace sink of the SoC	62
Analyzer.TRACK.RECORD()	Get record number matching search	63
Analyzer.TRIGGER.TIME()	Time of trigger point in trace	63
ARM Function		65
ARMARCHVERSION()	ARM architecture version of CPU	65
Advanced Register Trace (ART) Functions		66
In This Section		66
ART.FIRST()	Get record number of first trace record	66
ART.MAXSIZE()	Get max. size of trace buffer in records	67
ART.MODE()	Get ART recording mode	67
ART.RECORD.ADDRESS()	Get address recorded in trace record	67
ART.RECORD.OFFSET()	Get address in trace record as number	68
ART.RECORD.TIME()	Get timestamp of trace record	68
ART.RECORDS()	Get number of used trace records	68
ART.REF()	Get record number of reference record	68
ART.SIZE()	Get current trace buffer size in records	69
ART.STATE()	Get state of ART trace	69
ART.TRACK.RECORD()	Get record number matching search	69
AUTOFOCUS Functions		70
In This Section		70
AUTOFOCUS()	TRUE if AutoFocus preprocessor attached	70
AUTOFOCUS.OK()	TRUE if command execution successful	70
AUTOFOCUS.FREQUENCY()	Frequency of trace-port clock	70
AVX Functions		71
In This Section		71
AVX()	Content of AVX register	71
AVX512()	Content of AVX512 register	72

Break Functions	73
In This Section	73
Break.Alpha.EXIST()	TRUE if Alpha breakpoint exists 73
Break.Beta.EXIST()	TRUE if Beta breakpoint exist 73
Break.Charly.EXIST()	TRUE if Charly breakpoint exists 74
Break.Program.EXIST()	TRUE if enabled program breakpoint exists 74
Break.ReadWrite.EXIST()	TRUE if enabled data address breakpoint exists 74
BMC Functions (Benchmark Counter)	75
In This Section	75
BMC.CLOCK()	Frequency of core clock 75
BMC.COUNTER()	Value of a benchmark counter 75
BMC.COUNTER.BYNAME()	Value of a benchmark counter 76
BMC.COUNTER.CORE()	Value of a benchmark counter 76
BMC.COUNTER.BYNAME.CORE()	Value of a benchmark counter 77
BMC.OVERFLOW()	TRUE if benchmark counter overflow 77
BMC.OVERFLOW.BYNAME()	TRUE if benchmark counter overflow 77
BMC.OVERFLOW.CORE()	TRUE if benchmark counter overflow 78
BMC.OVERFLOW.BYNAME.CORE()	TRUE if benchmark counter overflow 78
Boundary Scan Description Language (BSDL) Functions	79
In This Section	79
BSDL.CHECK.BYPASS()	Chain bypass test 79
BSDL.CHECK.FLASHCONF()	Flash configuration test 79
BSDL.CHECK.IDCODE()	Chain IDCODE test 79
BSDL.GetDRBit()	Data register bit 80
BSDL.GetPortLevel()	Port level value 80
CABLE Functions	81
In This Section	81
CABLE.GalvanicISolation()	Cable has galvanic isolation 81
CABLE.GalvanicISolation.FIRMWARE()	Adapter firmware version 81
CABLE.GalvanicISolation.SERIAL()	Serial number of adapter 81
CABLE.NAME()	Name of debug cable 82
CABLE.SERIAL()	Serial number of debug cable 82
CABLE.TWOWIRE()	TRUE if two-wire debugging supported 82
CACHE Functions	83
In This Section	83
CACHE.DC.DIRTY()	Dirty-flag of L1 Data Cache Line 83
CACHE.DC.DIRTYMASK()	Dirty-flag mask of L1 Data Cache Line 84
CACHE.DC.LRU()	LRU information of L1 Data Cache Line 84
CACHE.DC.TAG()	Address Tag of L1 Data Cache Line 84
CACHE.DC.VALID()	Valid-flag of L1 Data Cache Line 85
CACHE.DC.VALIDMASK()	Valid-flag mask of L1 Data Cache Line 85
CACHE.IC.DIRTY()	Dirty-flag of L1 Unified Cache Line 86

CACHE.IC.DIRTYMASK()	Dirty-flag mask of L1 Unified Cache Line	86
CACHE.IC.LRU()	LRU information of L1 Instruction Cache Line	86
CACHE.IC.TAG()	Address Tag of L1 Instruction Cache Line	87
CACHE.IC.VALID()	Valid-flag of L1 Instruction Cache Line	87
CACHE.IC.VALIDMASK()	Valid-flag mask of L1 Instruction Cache Line	87
CACHE.L2.DIRTY()	Dirty-flag of L2 Cache Line	88
CACHE.L2.DIRTYMASK()	Dirty-flag mask of L2 Cache Line	88
CACHE.L2.LRU()	LRU information of L2 Cache Line	88
CACHE.L2.SHARED()	Shared-flag of L2 Cache Line	89
CACHE.L2.SHAREDMASK()	Shared-flag mask of L2 Cache Line	89
CACHE.L2.TAG()	Address Tag of L2 Cache Line	89
CACHE.L2.VALID()	Valid-flag of L2 Cache Line	90
CACHE.L2.VALIDMASK()	Valid-flag mask of L2 Cache Line	90
CACHE.L3.DIRTY()	Dirty-flag of L3 Cache Line	90
CACHE.L3.DIRTYMASK()	Dirty-flag of L3 Cache Line	91
CACHE.L3.LRU()	LRU information of L3 Cache Line	91
CACHE.L3.TAG()	Address Tag of L3 Cache Line	91
CACHE.L3.VALID()	Valid-flag of L3 Cache Line	92
CACHE.L3.VALIDMASK()	Valid-flag mask of L3 Cache Line	92
CAnalyzer Functions		93
In This Section		93
CAnalyzer()	Check if CAnalyzer command group is available	94
CAnalyzer.BOTHcables()	TRUE if both debug cables are plugged	94
CAnalyzer.CableTYPE()	Type of adapter	94
CAnalyzer.DebugCable()	CombiProbe whisker cable is A or B	95
CAnalyzer.FEATURE()	Query features of CAnalyzer hardware	95
CAnalyzer.FIRST()	Get record number of first trace record	97
CAnalyzer.MAXSIZE()	Get max. size of trace buffer in records	97
CAnalyzer.PIN()	Status of trace pins	98
CAnalyzer.RECORD.ADDRESS()	Get address recorded in trace record	98
CAnalyzer.RECORD.DATA()	Get data recorded in trace record	98
CAnalyzer.RECORD.OFFSET()	Get address in trace record as number	99
CAnalyzer.RECORD.TIME()	Get timestamp of trace record	99
CAnalyzer.RECORDS()	Get number of used trace records	99
CAnalyzer.REF()	Get record number of reference record	99
CAnalyzer.SIZE()	Get current trace buffer size in records	100
CAnalyzer.STATE()	Get state of Compact Analyzer	100
CAnalyzer.TraceCLOCK()	Get trace port frequency	101
CAnalyzer.TraceCONNECT()	Name of trace sink of the SoC	102
CAnalyzer.TracePort()	CombiProbe whisker cable is A or B	102
CAnalyzer.TRACK.RECORD()	Get record number matching search	102
CERBEURS Functions		103
CERBERUS.IOINFO()	IOINFO of Cerberus module	103

CERBERUS.IOINFO.IFLCK()	TRUE if IF_LCK bit in Cerberus INONFO set	103
CHIP Functions		104
CHIP.EmulationDevice()	TRUE if emulation device	104
CHIP.STEPping()	Major silicon step of an TriCore AURIX device	104
CIProbe Functions (Analog Probe for CombiProbe or μTrace)		105
In This Section		105
CIProbe()	TRUE if Compact Analyzer hardware	105
CIProbe.ADC.ENABLE()	TRUE if channel is enabled	105
CIProbe.ADC.SHUNT()	Get shunt-resistor value	105
CIProbe.MAXSIZE()	Get max. size of trace buffer in records	106
CIProbe.RECORDS()	Get number of used trace records	106
CIProbe.SIZE()	Get current trace buffer size in records	106
CIProbe.STATE()	Get state of Compact Analyzer for CIProbe	107
CIProbe.TRACK.RECORD()	Get record number matching search	107
CMI Function		108
CMIBASE()	Base addresses of CMI modules	108
COMPONENT Functions		109
In This Section		109
COMPONENT.AVAILABLE()	TRUE if debug/trace peripherals available on CPU	109
COMPONENT.BASE()	Base address of debug/trace peripherals	110
COMPONENT.NAME()	User-defined name of debug/trace peripherals	110
COMPONENT.TYPE()	Type of debug/trace peripherals	111
COMPONENTNAME()	Name of debug/trace peripheral	111
COMPONENTNUMBER()	Number of valid debug/trace peripherals	112
CORE Functions		113
In This Section		113
CONFIGNUMBER()	Number of cores configured in TRACE32	114
CORE()	Get the selected core	114
CORE.ISACTIVE()	TRUE if this core is active	115
CORE.ISASSIGNED()	TRUE if physical core is assigned to debug session	116
CORE.LOGICALTOPHYSICAL()	This is the physical core number	117
CORE.NAMES()	Physical core names assigned to TRACE32	118
CORENAME()	Name of core within selected chip	119
CORE.NUMBER()	Number of logical cores	119
CORE.PHYSICALTOLOGICAL()	Logical core number of physical core	121
Count Functions		122
In This Section		122
Count.Frequency()	Frequency of last measurement	122
Count.LEVEL()	Level of frequency counter input	122
Count.Time()	Time of last measurement	123
Count.VALUE()	Samples of the Count.GO command	123

COverage Functions	124
In This Section	124
COverage.BDONE()	Byte count of all executed instructions 125
COverage.IDLE()	TRUE if all trace data for code coverage are processed 125
COverage.LOAD.KEY()	Key from last ACD file 126
COverage.Percentage()	Percentage of code coverage 126
COverage.SCOPE()	Degree of code coverage 126
COverage.SourceMetric()	Active code coverage criterion 128
COverage.TreeWalk()	Walk symbol tree 129
CPU Functions	130
In This Section	130
CPU.ADDRESS()	Start address of memory section 130
CPU.ADDRESS.PhysicalINDEX()	Section start address of given core 130
CPU.FEATURE()	TRUE if CPU feature exists 131
CPU.PINCOUNT()	For internal usage only 136
CPUBONDOUT()	Name of boundout processor 136
CPUCOREVERSION()	Core or architecture version of CPU 136
CPUDERIVATE()	Main part of processor name 137
CPUFAMILY()	Family name of processor 137
CPUHELP()	For internal usage only 137
CPUIS()	TRUE if search string matches processor name 138
CPUIS64BIT()	TRUE if 64-bit architecture 138
DAP Functions	139
In This Section	139
DAP.Available()	TRUE if debugging via DAP is supported 139
DAP.USER<x>()	Status of the DAP user pin 139
Data Functions	140
In This Section	140
Data.<value_width>()	Memory contents in default endianness 140
Data.<value_width>.<endianness>()	Mem. contents in specified byte order 143
Data.<value_width>.<access_width>()	Mem. contents in specified width 145
Data.AL.ERRORS()	Get number of errors detected by Data.AllocList 146
Data.Float()	Get floating point number 146
Data.STRING()	Get zero-terminated string 147
Data.STRINGN()	Get zero-terminated string with a maximum length 148
Data.SUM()	Get checksum 148
Data.SWAP.<value_width>.<swap_width>()	Swap byte groups in word 149
Data.WSTRING()	Get zero-terminated wide string 150
Data.WSTRING.BigEndian()	Get big-endian wide string 150
Data.WSTRING.LittleEndian()	Get little-endian wide string 151
DEBUGGER Function	152
DEBUGGER.FEATURE()	Check debugger feature 152

DEBUGMODE Function		153
DEBUGMODE()	Current debug mode	153
DISASSEMBLE Function		154
DISASSEMBLE.ADDRESS()	Disassembled instruction at address	154
DONGLEID Function		155
DONGLEID()	Serial number of USB WibuKey	155
ELA Function (ARM Coresight Embedded Logic Analyzer)		156
ELABASE()	ELA base address	156
DPP Function (C166/ST10 only)		156
DPP()	Content of DPP register	156
EPOC Functions		157
In This Section		157
EPOC.DATAADDRESS()	Start address of data area (EPOC debugger)	157
EPOC.ENTRYPOINT()	Entry address of debug task	157
EPOC.TEXTADDRESS()	Start address of code area (EPOC debugger)	157
ERROR Functions (target-dependent)		158
ERROR.ADDRESS()	Address of last occurred memory access error	158
ETM Functions		159
In This Section		159
ETM()	TRUE if ETM trace is available	159
ETM.ADDRCOMP()	For internal usage only	160
ETM.ADDRCOMPTOTAL()	Number of ETM address comparator pair	160
ETM.COUNTERS()	Number of ETM counters	160
ETM.DATACOMP()	Number of ETM data comparators	160
ETM.EXTIN()	Number of internal ETM inputs	161
ETM.EXTOUT()	Number of external ETM outputs	161
ETM.FIFOFULL()	ETM fifofull logic	161
ETM.MAP()	Number of ETM memory map decoders	161
ETM.PROTOCOL()	Version of ETM protocol	162
ETM.SEQUENCER()	Number of ETM sequencers	162
ETM.TraceCore()	TRUE if the core is traced	162
EXTENDED Function (Z80 only)		163
EXTENDED()	TRUE if register CBAR > 0	163
FDX Function		164
FDX.INSTRING()	Content at FDX memory address	164
FDX.TargetSTALLS()	Monitor FDX communication stalls on the target	164
FLAG Functions		165
In This Section		165
FLAG()	TRUE if hardware flag system available	165
FLAG.READ()	FLAG memory bytes with read access bit	165

FLAG.WRITE()	FLAG memory bytes with write access bit	165
FLASH Functions		166
In This Section		166
FLASH.CFI.SIZE()	Size of FLASH devices	167
FLASH.CFI.WIDTH()	Data bus width of FLASH devices	167
FLASH.CLock.Frequency()	FLASH clock value	167
FLASH.ID()	FLASH manufacturer and device ID	168
FLASH.List.STATE.PENDING()	Number of pending sectors	169
FLASH.List.TYPE()	FLASH family code of FLASH list entry	169
FLASH.ProgramMODE()	FLASH programming modes	170
FLASH.ProgramMODE.OPTION()	FLASH programming options	171
FLASH.SECTOR.BEGIN()	Start address	172
FLASH.SECTOR.END()	End address	172
FLASH.SECTOR.EXIST()	TRUE if sector exists	172
FLASH.SECTOR.EXTRAvalue()	Extra value of FLASH.Create	173
FLASH.SECTOR.NEXT()	Address of next sector	174
FLASH.SECTOR.OTP()	TRUE if OTP sector	174
FLASH.SECTOR.OPTION()	Options of a FLASH sector	175
FLASH.SECTOR.RANGE()	Address range of a FLASH sector	176
FLASH.SECTOR.SIZE()	Size in bytes	176
FLASH.SECTOR.STATE()	FLASH programming state	176
FLASH.SECTOR.TYPE()	FLASH family code of sector	177
FLASH.SECTOR.WIDTH()	Width of FLASH sector	178
FLASH.TARGET.BUILD()	Build number of FLASH algorithm file	178
FLASH.TARGET.CODERANGE()	Code range of FLASH algorithm	179
FLASH.TARGET.DATARANGE()	Data range of FLASH algorithm	179
FLASH.TARGET.FILE()	Name of FLASH algorithm file	179
FLASH.UNIT()	Unit number of FLASH sector	180
FLASH.UNIT.BEGIN()	Unit start address	180
FLASH.UNIT.END()	Unit end address	180
FLASH.UNIT.EXIST()	TRUE if unit exists	181
FLASH.UNIT.NEXT()	Number of next unit	181
FLASHFILE Functions		182
In This Section		182
FLASHFILE.GETBADBLOCK.COUNT()	Number of bad blocks	182
FLASHFILE.GETBADBLOCK.NEXT()	Address of bad block	182
FLASHFILE.SPAREADDRESS()	Address of spare area	183
FPU Functions (Floating Point Unit)		184
In This Section		184
FPU()	FPU register contents	184
FPUCR()	FPU control register contents	184
FPU.RAW()	FPU register raw contents	184

FXU Function		185
FXU()	Content of FXU register	185
GROUP Function		185
GROUP.EXIST()	TRUE if group exists	185
Hardware Functions		186
In This Section		186
hardware.COMBIPROBE()	TRUE if CombiProbe	186
hardware.ESI()	TRUE if EPROM Simulator	186
hardware.ICD()	TRUE if TRACE32 debug hardware	187
hardware.POWERDEBUG()	TRUE if TRACE32 PowerDebug hardware	187
hardware.POWERINTEGRATOR()	TRUE if a PowerIntergrator	187
hardware.POWERINTEGRATOR2()	TRUE if a PowerIntegrator II	187
hardware.POWERNEXUS()	TRUE is a NEXUS Adapter	188
hardware.POWERPROBE()	TRUE is a PowerProbe	188
hardware.POWERTRACE()	TRUE if a PowerTrace Module	188
hardware.POWERTRACE2()	TRUE if a PowerTrace II	188
hardware.POWERTRACE2LITE()	TRUE if a PowerTrace II LITE	189
hardware.POWERTRACE3()	TRUE if a PowerTrace III	189
hardware.POWERTRACEPX()	TRUE if a PowerTrace PX	189
hardware.POWERTRACESERIAL()	TRUE if a PowerTrace Serial	189
hardware.POWERTRACESERIAL2()	TRUE if a PowerTrace Serial II	190
hardware.QUADPROBE()	TRUE if QuadProbe	190
hardware.UTRACE()	TRUE if μ Trace	190
HVX Function		191
HVX()	Content of HVX register	191
I2C Functions		192
In This Section		192
I2C.DATA()	Data read by I2C.TRANSFER	192
I2C.PIN()	Pin status	192
ID Functions		193
In This Section		193
ID.CABLE()	Hardware ID of debug cable	193
ID.POWERTRACEAUXPORT()	Hardware ID of device at PT aux port	193
ID.PREPROcessor()	Hardware ID of preprocessor	194
ID.SERialPort1()	Type-ID of Adapter or Preprocessor at PowerTrace Serial	195
ID.WHISKER()	ID of whisker cable	196
IDCODE()	ID code of TAP in JTAG chain	199
IDCODENUMBER()	Number of detected TAPs	199
Integrator Functions		200
In This Section		200
Integrator()	TRUE if PowerIntegrator	200

Integrator.FIRST()	Get record number of first trace record	200
Integrator.ADC.ENABLE()	Bitmask of enabled analog channels	201
Integrator.ADC.SHUNT()	Shunt-resistor value	201
Integrator.ANALOG()		201
Integrator.COUNTER.EVENT()	Get value of trigger program event counter	201
Integrator.COUNTER.EXTERN()	Value of trigger program external counter	202
Integrator.COUNTER.TIME()	Get value of trigger program time counter	202
Integrator.DIALOGDSEL()	For internal usage only	202
Integrator.DIALOGDSELGET()	For internal usage only	202
Integrator.DSEL()	For internal usage only	203
Integrator.FIND.PI_CHANNEL()	For internal usage only	203
Integrator.FIND.PI_WORD()	TRUE if signal word is defined	203
Integrator.FLAG()	Check state of trigger program FLAG	203
Integrator.GET()	Value of channel	204
Integrator.MAXSIZE()	Get max. size of trace buffer in records	204
Integrator.PROBE()	For internal usage only	204
Integrator.PROGRAMFILENAME()	File name of trigger program	204
Integrator.RECORD.DATA()	Get data recorded in trace record	205
Integrator.RECORD.TIME()	Get timestamp of trace record	205
Integrator.RECORDS()	Get number of used trace records	205
Integrator.REF()	Get record number of reference record	205
Integrator.SIZE()	Get current trace buffer size in records	206
Integrator.STATE()	Get state of the Integrator	206
Integrator.TRACK.RECORD()	Get record number matching search	206
Integrator.USB()	For internal usage only	207
INTERFACE Functions		208
In This Section		208
INTERFACE.CADI()	TRUE if connection to target is via CADI interface	208
INTERFACE.GDB()	TRUE if connection to target is via GDB interface	209
INTERFACE.GDI()	TRUE if connection to target via GDI interface	209
INTERFACE.HOST()	TRUE if application is debugged on host	209
interface.HOSTMCI()	TRUE if TRACE32 debug driver runs on host	209
INTERFACE.IRIS()	TRUE if connection to target is via IRIS interface	210
INTERFACE.MCD()	TRUE if connection to target via MCD interface	210
INTERFACE.NAME()	Name of debugger	210
INTERFACE.QNX()	TRUE if PBI=QNX	210
INTERFACE.SIM()	TRUE if simulator	211
IOBASE Functions		212
In This Section		212
IOBASE()	Base address of internal I/O's	212
IOBASE.ADDRESS()	Base address of internal I/O's with access class	212
IOBASE2()	Second base address of internal I/O's	212

IProbe Functions	213
In This Section	214
IProbe()	TRUE if IPROBE 214
IProbe.ADC.ENABLE()	TRUE if channel is enabled 214
IProbe.ADC.SHUNT()	Shunt resistor value of channel 215
IProbe.ANALOG()	TRUE if Analog Probe is plugged 216
IProbe.FIRST()	Get record number of first trace record 216
IProbe.GET()	Value of channel 216
IProbe.MAXSIZE()	Get max. size of trace buffer in records 217
IProbe.PROBE()	217
IProbe.RECORD.DATA()	Get data recorded in trace record 218
IProbe.RECORD.TIME()	Get timestamp of trace record 218
IProbe.RECORDS()	Get number of used trace records 219
IProbe.REF()	Get record number of reference record 220
IProbe.SIZE()	Get current trace buffer size in records 220
IProbe.STATE()	Get state of IProbe 221
IProbe.TRACK.RECORD()	Get record number matching search 221
JTAG Functions	222
In This Section	222
JTAG.MIPI34()	Query special MIPI34 pins 222
JTAG.PIN()	Level of JTAG signal 223
JTAG.SEQuence.RESULT()	Get result of JTAG sequence 223
JTAG.SEQuence.EXIST()	Check if a JTAG sequence exists 223
JTAG.SEQuence.LOCKED()	Check if a JTAG sequence is locked 224
JTAG.SHIFT()	TDO output of JTAG shift 224
JTAG.X7EFUSE.RESULT()	Result of JTAG.X7EFUSE command 225
JTAG.X7EFUSE.CNTL()	CNTL flags read by JTAG.X7EFUSE command 226
JTAG.X7EFUSE.DNA()	DNA value read by JTAG.X7EFUSE command 226
JTAG.X7EFUSE.KEY()	AES key read by JTAG.X7EFUSE command 227
JTAG.X7EFUSE.USER()	User code read by JTAG.X7EFUSE command 227
JTAG.XUSEFUSE.RESULT()	Result of JTAG.XUSEFUSE command 228
JTAG.XUSEFUSE.CNTL()	CNTL value read by JTAG.XUSEFUSE command 228
JTAG.XUSEFUSE.DNA()	DNA value read by JTAG.XUSEFUSE command 229
JTAG.XUSEFUSE.KEY()	AES key read by JTAG.XUSEFUSE command 229
JTAG.XUSEFUSE.RSA()	RSA hash read by JTAG.XUSEFUSE command 230
JTAG.XUSEFUSE.SEC()	SEC value read by JTAG.XUSEFUSE command 230
JTAG.XUSEFUSE.USER()	User code read by JTAG.XUSEFUSE command 231
JTAG.XUSEFUSE.USER128()	128 bit User code read by JTAG.XUSEFUSE 231
LOGGER Functions	232
In This Section	232
LOGGER.FIRST()	Get record number of first trace record 232
LOGGER.RECORD.ADDRESS()	Get address recorded in trace record 233
LOGGER.RECORD.DATA()	Get data recorded in trace record 233

LOGGER.RECORD.OFFSET()	Get address in trace record as number	233
LOGGER.RECORD.TIME()	Get timestamp of trace record	234
LOGGER.RECORDS()	Get number of used trace records	234
LOGGER.REF()	Get record number of reference record	234
LOGGER.SIZE()	Get current trace buffer size in records	234
LOGGER.STATE()	Get state of Logger trace	235
MachO Format Function (Apple)		236
MACHO.LASTUUID()	Universally unique identifier of MachO file	236
MAP Functions		237
In This Section		237
MAP.ROMSIZE()	Size of the defined ROM	237
MCDS Functions		238
In This Section		238
MCDS.MODULE.NAME()	Name of MCDS module	239
MCDS.MODULE.NUMBER()	Number-part of MCDS module ID	239
MCDS.MODULE.REVIsion()	Revision-part of MCDS module ID	239
MCDS.MODULE.TYPE()	Type-part of MCDS module ID	240
MCDS.STATE()	MCDS module is switched on/off	240
MCDS.TraceBuffer.LowerGAP()	Trace buffer lower gap	241
MCDS.TraceBuffer.SIZE()	Trace buffer size	242
MCDS.TraceBuffer.UpperGAP()	Trace buffer upper gap	242
MMU Functions (Memory Management Unit)		243
In This Section		243
MMU()	Value of MMU register	243
MMU.DEFAULTTPT()	Base address of default page table	244
MMU.DEFAULTTTRANS.<range>()	Query MMU setup	245
MMU.FORMAT()	Currently selected MMU format	247
MMU.FORMAT.DETECTED()	Auto-detection of page table format	248
MMU.FORMAT.DETECTED.ZONE()	Auto-detection of page table format	249
MMX Function (MultiMedia eXtension)		250
MMX()	Value of MMX register	250
MONITOR Function		250
MONITOR()	TRUE if debugger is running as monitor	250
NEXUS Functions		251
In This Section		251
NEXUS()	TRUE if Nexus trace is supported	251
NEXUS.RTTBUILD()	RTT build register	251
NEXUS.PortMode()	Current PortMode setting	252
NEXUS.PortSize()	Current PortSize setting	252
Onchip Functions		253
In This Section		253

Onchip()	TRUE if the onchip trace is available	253
Onchip.FIRST()	Get record number of first trace record	253
Onchip.FLOW.ERRORS()	Get number of flow errors / hard errors	253
Onchip.FLOW.FIFOFULL()	Get number of FIFO overflows	254
Onchip.MAXSIZE()	Get max. size of trace buffer in records	255
Onchip.RECORD.ADDRESS()	Get address recorded in trace record	255
Onchip.RECORD.DATA()	Get data recorded in trace record	255
Onchip.RECORD.OFFSET()	Get address in trace record as number	255
Onchip.RECORD.TIME()	Get timestamp of trace record	256
Onchip.RECORDS()	Get number of used trace records	256
Onchip.REF()	Get record number of reference record	256
Onchip.SIZE()	Get current trace buffer size in records	256
Onchip.STATE()	Get state of Onchip trace	257
Onchip.TraceCONNECT()	Name of trace sink of the SoC	257
Onchip.TRACK.RECORD()	Get record number matching search	258
PBI Function		259
PBI()	Name of used debug back-end	259
PCI Functions		260
In This Section		260
PCI.Read.B()	Byte from PCI register	260
PCI.Read.L()	Long from PCI register	260
PCI.Read.W()	Word from PCI register	260
PER Functions		261
In This Section		261
PER.<width>()	Memory contents in default endianness	261
PER.<width>.<endianness>()	Memory contents in specified endianness	262
PER.ADDRESS()	Address of register(field)	263
PER.ADDRESS.<sub_cmd>()	Check access security in PER file	264
PER.ARG()	Argument of PER.view command	264
PER.ARG.ADDRESS()	Address argument of PER.view command	265
PER.BASE()	Last BASE address	265
PER.Buffer.<width>()	Value from buffer	266
PER.EVAL()	Value of expression in PER file	267
PER.FILENAME()	PER file name	267
PER.SAVEINDEX()	Value from indexed register	268
PER.VALUE()	Value of register(field)	268
PER.VALUE.STRING()	Value of BITFLD as string	269
PERF Functions (Performance)		270
In This Section		270
PERF.MEMORY.HITS()	Number of memory samples	270
PERF.MEMORY.SnoopAddress()	Snoop memory address	271
PERF.MEMORY.SnoopSize()	Snoop size	271

PERF.METHOD()	Recording method	271
PERF.MODE()	Get Performance Analyzer recording mode	272
PERF.PC.HITS()	Number of PC samples	272
PERF.RATE()	Number of snoops per second	272
PERF.RunTime()	Retained time for program run	273
PERF.SNOOPFAILS()	Number of snoop fails	273
PERF.STATE()	Get state of Performance Analyzer	273
PERF.TASK.HITS()	Number of task samples	274
Port Analyzer Functions		275
In This Section		275
PORT.GET()	Value of channel	275
PORT.MAXSIZE()	Get max. size of trace buffer in records	275
PORT.RECORDS()	Get number of used trace records	275
PORT.REF()	Get record number of reference record	276
PORT.SIZE()	Get current trace buffer size in records	276
PORT.STATE()	Get state of Port Analyzer	276
PORT.TRACK.RECORD()	Get record number matching search	276
PORTANALYZER()		277
PORTSHARING Function		277
PORTSHARING()	Current setting of PortSHaRing	277
POWER Functions		278
In This Section		278
PowerProbe Functions		279
In This Section		279
PROBE.COUNTER.EVENT()	Get value of trigger program event counter	279
PROBE.COUNTER.EXTERN()	Get value of trigger program external counter	279
PROBE.COUNTER.TIME()	Get value of trigger program time counter	279
Probe.FIRST()	Get record number of first trace record	280
PROBE.FLAG()	Check state of trigger program FLAG	280
PROBE.GET()	Value of channel	280
PROBE.MAXSIZE()	Get max. size of trace buffer in records	280
PROBE.RECORD.DATA()	Get data recorded in trace record	281
PROBE.RECORD.TIME()	Get timestamp of trace record	281
PROBE.RECORDS()	Get number of used trace records	281
PROBE.REF()	Get record number of reference record	281
PROBE.SIZE()	Get current trace buffer size in records	282
PROBE.STATE()	Get state of PowerProbe	282
PROBE.TRACK.RECORD()	Get record number matching search	283
Program Pointer Function		284
PP()	Address of program pointer (access class, space ID, program counter)	284
Register Functions		285

Register()	Content of register	285
Register.LIST()	First / next register name	286
Register.Valid()	Valid register value	287
RTS Functions		288
In This Section		288
RTS.ERROR()	Check for flowerrors during RTS processing	288
RTS.NOCODE()	Check for RTS NOCODE error	288
RTS.FIFOFULL()	Check for FIFO full error in RTS	289
RTS.RECORD()	Find record causing an error in RTS	289
RTS.RECORDS()	Get number of trace records transferred to RTS	289
RTS.BUSY()	Check if RTS is busy	289
RunTime Functions		290
In This Section		290
RunTime.ACCURACY()	Accuracy of run-time counter	290
RunTime.ACTUAL()		290
RunTime.LAST()		290
RunTime.LASTRUN()		291
RunTime.REFA()		291
RunTime.REFB()		291
SMMU Functions		292
SMMU.BaseADDRESS()	Base address of SMMU	292
SMMU.StreamID2SMRG()	Find match for stream ID	292
SNOOPer Functions		294
In This Section		294
SNOOPer.FIRST()	Get record number of first trace record	294
SNOOPer.MAXSIZE()	Get max. size of trace buffer in records	295
SNOOPer.RECORD.ADDRESS()	Get address recorded in trace record	295
SNOOPer.RECORD.DATA()	Get data recorded in trace record	295
SNOOPer.RECORD.OFFSET()	Get address in trace record as number	295
SNOOPer.RECORD.TIME()	Get timestamp of trace record	296
SNOOPer.RECORDS()	Get number of used trace records	296
SNOOPer.REF()	Get record number of reference record	296
SNOOPer.SIZE()	Get current trace buffer size in records	296
SNOOPer.STATE()	Get state of SNOOPer trace	297
STATE Functions (Target State)		298
In This Section		298
STATE.HALT()		298
STATE.OSLK()		298
STATE.POWER()		299
STATE.PROCESSOR()		300
STATE.RESET()		301
STATE.RUN()		301

STATE.TARGET()	State of target displayed in TRACE32 state line	301
SPE Function		301
SPE()	Content from SPE register	301
SSE Function		302
SSE()	Segment from SSE register	302
Stimuli Generator Function		303
hardware.STG()	TRUE if Stimuli Generator hardware	303
sYmbol Functions		304
In This Section		304
sYmbol.AutoLOAD.CHECK()	Update option for the symbol autoloader	304
sYmbol.AutoLOAD.CHECKCMD()	Load command for symbol autoloader	304
sYmbol.AutoLOAD.CONFIG()	Used sub-command	305
sYmbol.BEGIN()	First address of symbol	305
sYmbol.COUNT()	Number of symbols	306
sYmbol.ECA.BINary.GAPNUMBER()	Number of observability gaps	306
sYmbol.END()	Last address of symbol	306
sYmbol.EPILOG()	Address of return point	307
sYmbol.EXIST()	TRUE if symbol exists	308
sYmbol.EXIT()	Exit address of function	308
sYmbol.FUNCTION()	Function name	309
sYmbol.IMPORT()	Import file names	309
sYmbol.ISFUNCTION()	TRUE if symbol is function	309
sYmbol.ISVARIABLE()	TRUE if symbol is variable	310
sYmbol.LANGUAGE()	Selected high-level language	311
sYmbol.List.MAP.<x>()	Information about address ranges on the target	311
sYmbol.LIST.PROGRAM()	Path and file name of binary files	312
sYmbol.List.PROGRAM.<x>()	Information about loaded programs	313
sYmbol.List.SEction.<x>()	Information about section ranges	314
sYmbol.LIST.SOURCE()	File location of source file	316
sYmbol.MATCHES()	Number of occurrences	316
sYmbol.NAME()	Symbol path and name based on address	317
sYmbol.NAME.AT()	Resolve ambiguous symbols based on address	317
sYmbol.NEXT.BEGIN()	Start address of next symbol	318
sYmbol.RANGE()	Address range of symbol	318
sYmbol.SEARCHFILE()	Absolute path of source file	318
sYmbol.SECADDRESS()	Start address of section	320
sYmbol.SECEND()	End address of section	320
sYmbol.SECEXIST()	Check for existence of a section	320
sYmbol.SECNAME()	Section name	321
sYmbol.SECPRANGE()	Physical address range of section	321
sYmbol.SECRANGE()	Logical address range of section	321
sYmbol.SIZEOF()	Size of debug symbol	322

sYmbol.SOURCEFILE()	Name of source file	322
sYmbol.SOURCELINE()	HLL-line number of address	323
sYmbol.SOURCEPATH()	TRUE if path is search path	324
sYmbol.STATE()	Value from sYmbol.state window	324
sYmbol.TRANSPOSE()	Transpose program and module names	324
sYmbol.TYPE()	Type of symbol	325
sYmbol.VARNAME()	Name of variable or structure element	326
SYStem Functions		327
In This Section		327
SYStem.ACCESS.DENIED()	TRUE if memory access is denied	328
SYStem.AMBA()	TRUE if AMBA bus mode is selected	328
SYStem.BigEndian()	TRUE if target core runs in big endian mode	328
SYStem.CADlconfig.RemoteServer()		329
SYStem.CADlconfig.Traceconfig()		330
SYStem.CONFIG.<tap_position>()		331
SYStem.CONFIG.DEBUGPORT()		331
SYStem.CONFIG.DEBUGPORTTYPE()		331
SYStem.CONFIG.JTAGTAP()	Return the JTAG PRE and POST settings	332
SYStem.CONFIG.ListCORE()		335
SYStem.CONFIG.ListSIM()		336
SYStem.CONFIG.Slave()		336
SYStem.CONFIG.TAPState()		337
SYStem.CPU()	Name of processor	337
SYStem.GTL.CALLCOUNTER()	Amount of calls to GTL library	338
SYStem.GTL.CONNECTED()	Connection status	338
SYStem.GTL.CYCLECOUNTER()	load GTL interface for bit banging protocol	338
SYStem.GTL.LIBname()	Name of GTL library	338
SYStem.GTL.ModelINFO()	Info string from GTL API	339
SYStem.GTL.ModelNAME()	Model Name	339
SYStem.GTL.PLUGINVERSION()	Version number	339
SYStem.GTL.TransactorNAME()	Transactor name	340
SYStem.GTL.TransactorTYPE()	Transactor type	340
SYStem.GTL.VENDORID()	Vendor ID	340
SYStem.GTL.VERSION()	Version number	341
SYStem.HOOK()		341
SYStem.IMASKASM()		341
SYStem.IMASKHLL()		341
SYStem.INSTANCE()	Index of TRACE32 PowerView instance	342
SYStem.INSTANCECOUNT()	Count of GUIs connected to a PowerDebug	342
SYStem.IRISconfig.RemoteServer()		343
SYStem.JtagClock()		343
SYStem.LittleEndian()		343
SYStem.MCDCommand.ResultString()		344

SYStem.MCDconfig.LIBrary()		344
SYStem.Mode()		344
SYStem.NOTRAP()	1 if the option NOTRAP is active	345
SYStem.Option.DUALPORT()	State of like-named command	345
SYStem.Option.MACHINESPACES()	State of like-named command	345
SYStem.Option.MMUSPACES()	State of like-named command	346
SYStem.Option.EnReset()	State of like-named command	346
SYStem.Option.ResBreak()	State of like-named command	346
SYStem.Option.SPILLLOCation()	State of like-named command	347
SYStem.Option.ZoneSPACES()	State of like-named command	347
SYStem.RESetBehavior()	Current setting of RESetBehavior	348
SYStem.Up()	TRUE if debugger has access to debug resources	348
SYStem.USECORE()		350
SYStem.USEMASK()		351
TASK Functions		352
In This Section		352
TASK()	Name of current task	353
TASK.ACCESS()	Access class	353
TASK.ACCESS.ZONE()	Access class zone	353
TASK.BACK()	Background task number	353
TASK.CONFIG()	OS Awareness configuration information	354
TASK.CONFIGFILE()	Path of loaded OS Awareness	354
TASK.COUNT()	Number of tasks	354
TASK.CURRENT.MACHINEID()	ID of current machine	355
TASK.CURRENT.SPACEID()	ID of current MMU space	355
TASK.CURRENT.TASK()	Magic value of current task	355
TASK.CURRENT.TASKNAME()	Name of current task	355
TASK.FIRST()	First task in list	356
TASK.FORE()	Foreground task number	356
TASK.ID()	ID of task	356
TASK.MACHINE.ACCESS()	Default access class	356
TASK.MACHINE.ID()	ID of machine	357
TASK.MACHINE.NAME()	Name of machine	357
TASK.MACHINE.VTTB()	VTTB of machine	358
TASK.MAGIC()	Task magic number	358
TASK.MAGICADDRESS()	'magic address'	359
TASK.MAGICRANGE()	Range of 'magic address'	359
TASK.MAGICSIZE()	Size of 'magic address'	359
TASK.NAME()	Name of task	360
TASK.NEXT()	Next task in list	360
TASK.ORTIFILE()	Path of loaded ORTI file	361
TASK.SPACE.COUNT()	Number of spaces	361
TASK.SPACEID()	Space ID of task	362

TERM Functions (Terminal Window)	363
In This Section	363
TERM.LINE()	Get line from terminal window 363
TERM.NEWHANDLE()	Get next free terminal handle 363
TERM.READBUSY()	TRUE as long as TERM.READ is in progress 364
TERM.RETURNCODE()	Get returncode from terminal routine 364
TERM.TRIGGERED()	Get trigger state of terminal window 365
TPIU Functions	366
In This Section	366
TPIU.PortMode()	Port mode setting 366
TPIU.PortSize()	Port size setting 366
TPIU.SWVPrescaler()	SWVPrescaler value 367
TPUBASE Function	367
TPUBASE.ADDRESS()	Address of TPU 367
Trace Functions	368
In This Section	368
Trace.FIRST()	Get record number of first trace record 368
Trace.FLOW()	TRUE if trace method is flow trace 369
Trace.FLOW.ERRORS()	Get number of flow errors / hard errors 369
Trace.FLOW.FIFOFULL()	Get number of FIFO overflows 370
Trace.MAXSIZE()	Get max. size of trace buffer in records 370
Trace.METHOD()	Currently configured trace method 371
Trace.METHOD.Analyzer()	TRUE if the trace method is Analyzer 371
Trace.METHOD.ART()	TRUE if the trace method is ART 371
Trace.METHOD.CAnalyzer()	TRUE if the trace method is CAnalyzer 371
Trace.METHOD.FDX()	TRUE if the trace method is FDX 372
Trace.METHOD.HAnalyzer()	TRUE if the trace method is HAnalyzer 372
Trace.METHOD.Integrator()	TRUE if the trace method uses the Integrator 372
Trace.METHOD.IProbe()	TRUE if the trace method uses the IProbe 372
Trace.METHOD.LA()	TRUE if the trace method is LA 373
Trace.METHOD.LOGGER()	TRUE if the trace method is LOGGER 373
Trace.METHOD.ONCHIP()	TRUE if the trace method is ONCHIP 373
Trace.METHOD.Probe()	TRUE if trace method uses the PowerProbe 373
Trace.METHOD.SNOOPer()	TRUE if the trace method is SNOOPer 374
Trace.RECORD.ADDRESS()	Get address recorded in trace record 374
Trace.RECORD.DATA()	Get data recorded in trace record 374
Trace.RECORD.OFFSET()	Get address in trace record as number 375
Trace.RECORD.TIME()	Get timestamp of trace record 375
Trace.RECORDS()	Get number of used trace records 375
Trace.SIZE()	Get current trace buffer size in records 376
Trace.STATE()	Get state of PowerTrace hardware 376
Trace.STATistic.COUNT()	Number of occurrences of selected function 377

Trace.STATistic.EXIST()	TRUE if function exists in trace statistics	377
Trace.STATistic.FIRST()	Record number of start point for statistic analysis	377
Trace.STATistic.IMAX()	Longest time between function entry and exit	377
Trace.STATistic.IMIN()	Shortest time between function entry and exit	378
Trace.STATistic.Internal()	Time spent within the selected function	378
Trace.STATistic.LAST()	Record number of end point for statistic analysis	378
Trace.STATistic.MAX()	Maximum time of selected function	378
Trace.STATistic.MIN()	Minimum time of selected function	379
Trace.STATistic.Total()	Total time of selected function	379
Trace.TraceCONNECT()	Name of trace sink of the SoC	380
TRACEPORT Function		381
In This Section		381
TRACEPORT.LaneCount()	Number of serial lanes	381
TRACK Functions		382
In This Section		382
TRACK.ADDRESS()	Get tracking address	382
TRACK.COLUMN()	Number of column where the found item starts	382
TRACK.LINE()	Number of line containing the found item	382
TRACK.RECORD()	Number of record containing the found item	384
TRACK.STRING()	Current selection in a TRACE32 window	384
TRACK.TIME()	Timestamp of current tracking record	385
TRANS Functions (Debugger Address Translation)		386
In This Section		386
TRANS.LIST.NUMBER()	Number of TRANS.List entries	386
TRANS.LIST.LOGRANGE()	Query TRANS.List entry	387
TRANS.LIST.PHYSADDR()	Query TRANS.List entry	388
TRANS.LIST.TYPE()	Query TRANS.List entry	389
TRANS.ENABLE()	TRUE if address translation is enabled	390
TRANS.INTERMEDIATE()	Convert a guest logical address	390
TRANS.INTERMEDIATE.VALID()	TRUE if address translation is valid	391
TRANS.LINEAR()	Convert logical to linear address	391
TRANS.LINEAR.VALID()	TRUE if address translation is valid	392
TRANS.LOGICAL()	Convert physical to logical address	392
TRANS.LOGICAL.VALID()	TRUE if address translation is valid	393
TRANS.PHYSICAL()	Convert logical to physical address	393
TRANS.PHYSICAL.VALID()	TRUE if address translation is valid	396
TRANS.TABLEWALK()	TRUE if address translation table walk is ON	396
TSS Function		397
TSS()	TSS base address	397
Var Functions		398
In This Section		398
Var.ADDRESS()	Address of HLL expression	398

Var.BITPOS()	Bit position inside a C bit field	398
Var.BITSIZE()	Size of bit field element	399
Var.END()	Last address of HLL expression	399
Var.EXIST()	TRUE if HLL expression exists	400
Var.FVALUE()	Contents of HLL expression	401
Var.ISBIT()	TRUE if HLL expression is a bit field element	401
Var.RANGE()	Address range of HLL expression	402
Var.SIZEOF()	Size of HLL expression	402
Var.STRing()	Zero-terminated string or variable contents	403
Var.TYPEOF()	Type of HLL expression	403
Var.VALUE()	Value of HLL expression	404
VCO Function		405
VCO()	Frequency of VCO generator	405
VERSION Functions		406
In This Section		406
VERSION.BUILD()	Upper build number	407
VERSION.BUILD.BASE()	Lower build number	407
VERSION.CABLE()	Hardware version of debug cable	408
VERSION.DATE()	Version date YYYY/MM	408
VERSION.ENVironment()	TRACE32 environment setting	408
VERSION.FirmWare.DEBUG()	Version number of firmware	409
VERSION.SERIAL()	Serial number	409
VERSION.SERIAL.CABLE()	First serial number of debug cable	409
VERSION.SERIAL.DEBUG()	Serial number of debug module	410
VERSION.SERIAL.Integrator()	Serial number of PowerIntegrator	410
VERSION.SERIAL.NEXUSadapter()	Serial number of nexus adapter	410
VERSION.SERIAL.PREPROcessor()	Serial number of preprocessor	410
VERSION.SERIAL.POWERPROBE()	Serial number of PowerProbe	411
VERSION.SERIAL.POWERTRACEAUXPORT()	S/N of device at PT aux port	411
VERSION.SERIAL.SERialPort1()	S/N of device at Serial Port 1 of PT Serial	411
VERSION.SERIAL.WHISKER()	S/N of whiskers at CombiProbe or μ Trace	411
VERSION.SERIAL.TRACE()	Serial number of trace module	412
VERSION.SOFTWARE()	Release build or nightly build, etc.	413
VERSION.SOFTWARE.TYPE()	Software build type	414
VPU Functions		415
In This Section		415
VPU()	Value of VPU register	415
VPUCR()	Value of VRSAVE or VSCR register	415

Stimuli Generator Function Reference

Stimuli Generator Function Reference	(stg_func.pdf)	1

History	3
Functions	3

Application Notes for PRACTICE

Converter from GEL to PRACTICE

Converter from GEL to PRACTICE	(converter_gel.pdf)	1
Introduction		3
Brief Overview of Documents for New Users		3
Launching Converter		4
Using @prog, @data and @io		5
Using Menuitem, Hotmenu, Dialog and Slider		6
Recognizing Types of Identifiers in GEL		6
Functions Parameters		7
Preprocessor		7
Callback GEL Functions		7
Built-in GEL Functions		7
Using PRACTICE Commands from GEL Script		9
Converter-specific Reserved Words		9
Target CPU Register Names		10
Troubleshooting		10

General Commands

General Commands Reference Guide A

General Commands Reference Guide A	(general_ref_a.pdf)	1
History		11
AET		12
AET.CLEAR	Clear AET settings	12
AET.DataTrace	Configure AET data-trace	12
AET.GatedClock	Use trace port clock when no data is sent	13
AET.OFF	Switch AET off	13
AET.ON	Switch AET on	14
AET.PATTERN	Enable AET pattern generator	14

AET.PortClock	Select AET port mode	14
AET.PortMode	Select AET port mode	14
AET.PortSize	Select AET trace port width	15
AET.ReadWriteBreak	Control read/write breakpoints	15
AET.Register	Display the AET unit registers	16
AET.RESet	Reset AET settings	16
AET.SHADOW	Set AET shadow memory address	16
AET.STALL	Stall processor to prevent FIFO overflow	17
AET.state	Display AET settings	17
AET.SyncPeriod	Set synchronization frequency	17
AET.TagDataTrace	Tag AET data trace	18
AET.TimingTrace	Select AET trace timestamp information	18
AET.Trace	Control generation of trace information	19
AET.TraceID	Change the default ID for an AET trace source	19
AET.TracePriority	Define priority of AET	19
Analyzer		20
Analyzer	Trace method Analyzer, recording, and analysis commands	20
Analyzer-specific Trace Commands		22
Analyzer.Mode	Set the trace operation mode	22
Analyzer.PIPECompression	Enable compression in PIPE mode	24
Analyzer.Program	Program trigger unit	24
Analyzer.RecordAutoFill	Precision of run-time measurements	25
Analyzer.REMAP	Remap trace port channels	25
Analyzer.REMAP.CLOCK	Select input clock	26
Analyzer.REMAP.RESet	Reset pinout configuration	26
Analyzer.REMAP.state	Display remap configuration window	26
Analyzer.ReProgram	Program trigger unit	27
Analyzer.SAMPLE	Set AutoFocus sample time offset	27
Analyzer.TOut	Trigger output line	28
Analyzer.TraceCLOCK	Improve timestamps on PowerTrace	28
Generic Analyzer Trace Commands		29
Analyzer.ACCESS	Define access path to program code for trace decoding	29
Analyzer.Arm	Arm the trace	29
Analyzer.AutoArm	Arm automatically	29
Analyzer.AutoFocus	Calibrate AUTOFOCUS preprocessor	29
Analyzer.AutoInit	Automatic initialization	29
Analyzer.AutoStart	Automatic start	29
Analyzer.BookMark	Set a bookmark in trace listing	30
Analyzer.BookMarkToggle	Toggles a single trace bookmark	30
Analyzer.Chart	Display trace contents graphically	30
Analyzer.Chart.Address	Time between program events as a chart	30
Analyzer.Chart.AddressGROUP	Address group time chart	30

Analyzer.Chart.ChildTREE	Display callee context of a function as chart	30
Analyzer.Chart.DatasYmbol	Analyze pointer contents graphically	30
Analyzer.Chart.DistriB	Distribution display graphically	31
Analyzer.Chart.Func	Function activity chart	31
Analyzer.Chart.GROUP	Group activity chart	31
Analyzer.Chart.INTERRUPT	Display interrupt chart	31
Analyzer.Chart.INTERRUPTTREE	Display interrupt nesting	31
Analyzer.Chart.Line	Graphical HLL lines analysis	31
Analyzer.Chart.MODULE	Code execution broken down by module as chart	31
Analyzer.Chart.Nesting	Show function nesting at cursor position	32
Analyzer.Chart.PAddress	Which instructions accessed data address	32
Analyzer.Chart.PROGRAM	Code execution broken down by program	32
Analyzer.Chart.PsYmbol	Shows which functions accessed data address	32
Analyzer.Chart.RUNNABLE	Runnable activity chart	32
Analyzer.Chart.sYmbol	Symbol analysis	32
Analyzer.Chart.TASK	Task activity chart	32
Analyzer.Chart.TASKFunc	Task related function run-time analysis (legacy)	33
Analyzer.Chart.TASKINFO	Context ID special messages	33
Analyzer.Chart.TASKINTR	Display ISR2 time chart (ORTI)	33
Analyzer.Chart.TASKKernel	Task run-time chart with kernel markers (flat)	33
Analyzer.Chart.TASKORINTERRUPT	Task and interrupt activity chart	33
Analyzer.Chart.TASKORINTRState	Task and ISR2 state analysis	33
Analyzer.Chart.TASKSRV	Service routine run-time analysis	33
Analyzer.Chart.TASKState	Task state analysis	34
Analyzer.Chart.TASKVSINTERRUPT	Time chart of interrupted tasks	34
Analyzer.Chart.TASKVSINTR	Time chart of task-related interrupts	34
Analyzer.Chart.TREE	Display function chart as tree view	34
Analyzer.Chart.Var	Variable chart	34
Analyzer.Chart.VarState	Variable activity chart	34
Analyzer.CLOCK	Clock to calculate time out of cycle count information	34
Analyzer.ComPare	Compare trace contents	35
Analyzer.ComPareCODE	Compare trace with memory	35
Analyzer.CustomTrace	Custom trace	35
Analyzer.CustomTraceLoad	Load a DLL for trace analysis/Unload all DLLs	35
Analyzer.DISable	Disable the trace	35
Analyzer.DRAW	Plot trace data against time	35
Analyzer.DRAW.channel	Plot no-data values against time	35
Analyzer.DRAW.Data	Plot data values against time	36
Analyzer.DRAW.Var	Plot variable values against time	36
Analyzer.EXPORT	Export trace data for processing in other applications	36
Analyzer.EXPORT.ARTI	Export trace data as ARTI for CP	36
Analyzer.EXPORT.ARTIAP	Export trace data as ARTI for AP	36
Analyzer.EXPORT.Ascii	Export trace data as ASCII	36

Analyzer.EXPORT.Bin	Export trace data as binary file	36
Analyzer.EXPORT.BRANCHFLOW	Export branch events from trace data	37
Analyzer.EXPORT.CSVFunc	Export the function nesting to a CSV file	37
Analyzer.EXPORT.cycles	Export trace data	37
Analyzer.EXPORT.Func	Export function nesting	37
Analyzer.EXPORT.MDF	Export trace data as MDF	37
Analyzer.EXPORT.MTV	Export in MCDS Trace Viewer format	37
Analyzer.EXPORT.TASK	Export task switches	37
Analyzer.EXPORT.TASKEVENTS	Export task event to CSV	38
Analyzer.EXPORT.TracePort	Export trace packets as recorded at trace port	38
Analyzer.EXPORT.VCD	Export trace data in VCD format	38
Analyzer.EXPORT.VERILOG	Export trace data in VERILOG format	38
Analyzer.EXPORT.VHDL	Export trace data in VHDL format	38
Analyzer.ExtractCODE	Extract code from trace	38
Analyzer.FILE	Load a file into the file trace buffer	38
Analyzer.Find	Find specified entry in trace	39
Analyzer.FindAll	Find all specified entries in trace	39
Analyzer.FindChange	Search for changes in trace flow	39
Analyzer.FindProgram	Advanced trace search	39
Analyzer.FindReProgram	Activate advanced existing trace search program	39
Analyzer.FindViewProgram	State of advanced trace search programming	39
Analyzer.FLOWPROCESS	Process flowtrace	39
Analyzer.FLOWSTART	Restart flowtrace processing	40
Analyzer.Get	Display input level	40
Analyzer.GOTO	Move cursor to specified trace record	40
Analyzer.Init	Initialize trace	40
Analyzer.JOINFILE	Concatenate several trace recordings	40
Analyzer.List	List trace contents	40
Analyzer.ListNesting	Analyze function nesting	40
Analyzer.ListVar	List variable recorded to trace	40
Analyzer.LOAD	Load trace file for offline processing	41
Analyzer.MERGEFILE	Combine two trace files into one	41
Analyzer.OFF	Switch off	41
Analyzer.PipeWRITE	Connect to a named pipe to stream trace data	41
Analyzer.PlatformCLOCK	Set clock for platform traces	41
Analyzer.PortFilter	Specify utilization of trace memory	41
Analyzer.PortSize	Set external port size	41
Analyzer.PortType	Specify trace interface	42
Analyzer.PROfile	Rolling live plots of trace data	42
Analyzer.PROfile.CTU	Display complex trigger unit counter profile	42
Analyzer.PROfileChart	Profile charts	42
Analyzer.PROfileChart.Address	Address profile chart	42
Analyzer.PROfileChart.AddressGROUP	Address group time chart	42

Analyzer.PROfileChart.AddressRate	Address rate profile chart	42
Analyzer.PROfileChart.COUNTER	Display a profile chart	43
Analyzer.PROfileChart.DatasYmbol	Analyze pointer contents graphically	43
Analyzer.PROfileChart.DlStance	Time interval for a single event	43
Analyzer.PROfileChart.DistriB	Distribution display in time slices	43
Analyzer.PROfileChart.DURation	Time between two events	43
Analyzer.PROfileChart.GROUP	Group profile chart	43
Analyzer.PROfileChart.INTERRUPT	Display interrupt profile chart	43
Analyzer.PROfileChart.Line	HLL-line profile chart	44
Analyzer.PROfileChart.MODULE	Module profile chart	44
Analyzer.PROfileChart.PAddress	Which instructions accessed data address	44
Analyzer.PROfileChart.PROGRAM	Program profile chart	44
Analyzer.PROfileChart.PsYmbol	Which functions accessed data address	44
Analyzer.PROfileChart.Rate	Event frequency	44
Analyzer.PROfileChart.RUNNABLE	Runnable profile chart	44
Analyzer.PROfileChart.sYmbol	Dynamic program behavior graphically (flat)	45
Analyzer.PROfileChart.TASK	Dynamic task behavior graphically (flat)	45
Analyzer.PROfileChart.TASKINFO	Context ID special messages	45
Analyzer.PROfileChart.TASKINTR	ISR2 profile chart (ORTI)	45
Analyzer.PROfileChart.TASKKernel	Task profile chart with kernel markers	45
Analyzer.PROfileChart.TASKORINTERRUPT	Task and interrupt profile chart	45
Analyzer.PROfileChart.TASKSRV	Profile chart of OS service routines	45
Analyzer.PROfileChart.TASKVSINTERRUPT	Interrupted tasks	46
Analyzer.PROfileChart.TASKVSINTR	Profile chart for task-related interrupts	46
Analyzer.PROfileChart.Var	Variable profile chart	46
Analyzer.PROfileSTATistic	Statistical analysis in a table versus time	46
Analyzer.PROfileSTATistic.Address	Statistical analysis for addresses	46
Analyzer.PROfileSTATistic.AddressGROUP	Stat. for address groups	46
Analyzer.PROfileSTATistic.COUNTER	Statistical analysis for counter	46
Analyzer.PROfileSTATistic.DatasYmbol	Statistic analysis for pointer content	47
Analyzer.PROfileSTATistic.DistriB	Distribution statistical analysis	47
Analyzer.PROfileSTATistic.GROUP	Statistical analysis for groups	47
Analyzer.PROfileSTATistic.INTERRUPT	Statistical analysis for interrupts	47
Analyzer.PROfileSTATistic.Line	Statistical analysis for HLL lines	47
Analyzer.PROfileSTATistic.MODULE	Statistical analysis for modules	47
Analyzer.PROfileSTATistic.PAddress	Which instr. accessed data address	47
Analyzer.PROfileSTATistic.PROGRAM	Statistical analysis for programs	48
Analyzer.PROfileSTATistic.PsYmbol	Which functions accessed data address	48
Analyzer.PROfileSTATistic.RUNNABLE	Statistical analysis for runnables	48
Analyzer.PROfileSTATistic.sYmbol	Statistical analysis for symbols	48
Analyzer.PROfileSTATistic.TASK	Statistical analysis for tasks	48
Analyzer.PROfileSTATistic.TASKINFO	Context ID special messages	48
Analyzer.PROfileSTATistic.TASKINTR	Statistical analysis for ISR2 (ORTI)	48

Analyzer.PROfileSTATistic.TASKKernel	Stat. analysis with kernel markers	49
Analyzer.PROfileSTATistic.TASKORINTERRUPT	Interrupts and tasks	49
Analyzer.PROfileSTATistic.TASKSRV	Analysis of OS service routines	49
Analyzer.PROfileSTATistic.TASKVSINTERRUPT	Interrupted tasks	49
Analyzer.PROTOcol	Protocol analysis	49
Analyzer.PROTOcol.Chart	Graphic display for user-defined protocol	49
Analyzer.PROTOcol.Draw	Graphic display for user-defined protocol	49
Analyzer.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	50
Analyzer.PROTOcol.Find	Find in trace buffer for user-defined protocol	50
Analyzer.PROTOcol.list	Display trace buffer for user-defined protocol	50
Analyzer.PROTOcol.PROfileChart	Profile chart for user-defined protocol	50
Analyzer.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	50
Analyzer.PROTOcol.STATistic	Display statistics for user-defined protocol	50
Analyzer.REF	Set reference point for time measurement	50
Analyzer.RESet	Reset command	51
Analyzer.SAVE	Save trace for postprocessing in TRACE32	51
Analyzer.SelfArm	Automatic restart of trace recording	51
Analyzer.ShowFocus	Display data eye for AUTOFOCUS preprocessor	51
Analyzer.ShowFocusClockEye	Display clock eye	51
Analyzer.ShowFocusEye	Display data eye	51
Analyzer.SIZE	Define buffer size	51
Analyzer.SnapShot	Restart trace capturing once	52
Analyzer.SPY	Adaptive stream and analysis	52
Analyzer.state	Display trace configuration window	52
Analyzer.STATistic	Statistic analysis	52
Analyzer.STATistic.Address	Time between up to 8 program events	52
Analyzer.STATistic.AddressDIStance	Time interval for single program event	52
Analyzer.STATistic.AddressDURation	Time between two program events	52
Analyzer.STATistic.AddressGRouP	Address group run-time analysis	53
Analyzer.STATistic.ChildTREE	Show callee context of a function	53
Analyzer.STATistic.COLOr	Assign colors to function for colored graphics	53
Analyzer.STATistic.CYcle	Analyze cycle types	53
Analyzer.STATistic.DatasYmbol	Analyze pointer contents numerically	53
Analyzer.STATistic.DIStance	Time interval for a single event	53
Analyzer.STATistic.DistriB	Distribution analysis	53
Analyzer.STATistic.DURation	Time between two events	54
Analyzer.STATistic.FIRST	Start point for statistic analysis	54
Analyzer.STATistic.Func	Nesting function runtime analysis	54
Analyzer.STATistic.FuncDURation	Statistic analysis of single function	54
Analyzer.STATistic.FuncDURationInternal	Statistic analysis of single func.	54
Analyzer.STATistic.GROuP	Group run-time analysis	54
Analyzer.STATistic.Ignore	Ignore false records in statistic	54
Analyzer.STATistic.INTERRUPT	Interrupt statistic	55

Analyzer.STATistic.InterruptIsFunction	Statistics interrupt processing	55
Analyzer.STATistic.InterruptIsKernel	Statistics interrupt processing	55
Analyzer.STATistic.InterruptIsKernelFunction	Statistics interrupt processing	55
Analyzer.STATistic.InterruptIsTaskswitch	Statistics interrupt processing	55
Analyzer.STATistic.INTERRUPTTREE	Display interrupt nesting	55
Analyzer.STATistic.LAST	End point for statistic analysis	55
Analyzer.STATistic.Line	High-level source code line analysis	56
Analyzer.STATistic.LINKage	Per caller statistic of function	56
Analyzer.STATistic.Measure	Analyze the performance of a single signal	56
Analyzer.STATistic.MODULE	Code execution broken down by module	56
Analyzer.STATistic.PAddress	Which instructions accessed data address	56
Analyzer.STATistic.ParentTREE	Show the call context of a function	56
Analyzer.STATistic.PROCESS	Re-process statistics	56
Analyzer.STATistic.PROGRAM	Code execution broken down by program	57
Analyzer.STATistic.PsYmbol	Shows which functions accessed data address	57
Analyzer.STATistic.RUNNABLE	Runnable runtime analysis	57
Analyzer.STATistic.RUNNABLEDuration	Runnable duration analysis	57
Analyzer.STATistic.Sort	Specify sorting criteria for statistic commands	57
Analyzer.STATistic.sYmbol	Flat run-time analysis	57
Analyzer.STATistic.TASK	Task activity statistic	57
Analyzer.STATistic.TASKFunc	Task related function run-time analysis	58
Analyzer.STATistic.TASKINFO	Context ID special messages	58
Analyzer.STATistic.TASKINTR	ISR2 statistic (ORTI)	58
Analyzer.STATistic.TASKKernel	Task analysis with kernel markers (flat)	58
Analyzer.STATistic.TASKLOCK	Analyze lock accesses from tasks	58
Analyzer.STATistic.TASKORINTERRUPT	Statistic of interrupts and tasks	58
Analyzer.STATistic.TASKORINTRState	Task and ISR2 statistic analysis	58
Analyzer.STATistic.TASKSRV	Analysis of time in OS service routines	59
Analyzer.STATistic.TASKState	Performance analysis	59
Analyzer.STATistic.TASKStateDuration	Task state runtime analysis	59
Analyzer.STATistic.TASKTREE	Tree display of task specific functions	59
Analyzer.STATistic.TASKVSINTERRUPT	Statistic of interrupts, task-related	59
Analyzer.STATistic.TASKVSINTR	ISR2 statistic (ORTI), task related	59
Analyzer.STATistic.TREE	Tree display of nesting function run-time analysis	59
Analyzer.STATistic.Use	Use records	60
Analyzer.STATistic.Var	Statistic of variable accesses	60
Analyzer.STREAMCompression	Select compression mode for streaming	60
Analyzer.STREAMFILE	Specify temporary streaming file path	60
Analyzer.STREAMFileLimit	Set size limit for streaming file	60
Analyzer.STREAMLOAD	Load streaming file from disk	60
Analyzer.STREAMSAVE	Save streaming file to disk	60
Analyzer.TDelay	Trigger delay	61
Analyzer.TERMination	Use trace line termination of preprocessor	61

Analyzer.TestFocus	Test trace port recording	61
Analyzer.TestFocusClockEye	Scan clock eye	61
Analyzer.TestFocusEye	Check signal integrity	61
Analyzer.TestUtilization	Tests trace port utilization	61
Analyzer.THreshold	Optimize threshold for trace lines	61
Analyzer.Timing	Waveform of trace buffer	62
Analyzer.TraceCONNECT	Select on-chip peripheral sink	62
Analyzer.TRACK	Set tracking record	62
Analyzer.TSElect	Select trigger source	62
Analyzer.View	Display single record	62
Analyzer.ZERO	Align timestamps of trace and timing analyzers	62
APU		63
APU	Auxiliary processing unit	63
APU.Break	APU breakpoints	63
APU.Break.Delete	Delete APU breakpoint	63
APU.Break.direct	Stop the APU	64
APU.Break.Init	Initialize APU breakpoint system	64
APU.Break.List	List APU breakpoints	64
APU.Break.RESet	Reset APU breakpoint system	65
APU.Break.Set	Set permanent APU breakpoint	65
APU.command	Execute APU specific command	66
APU.Data	APU data command group	66
APU.Data.dump	Data memory display	66
APU.Data.List	Symbolic display	67
APU.Data.LOAD	Load file	67
APU.Data.Set	Data memory modification	68
APU.Go	Start the APU	68
APU.GREP	Search for string	69
APU.List	View program	69
APU.ListHll	View program source	70
APU.LOAD	Load APU library	70
APU.Register	Show APU register window	70
APU.Register.Set	Register modification	71
APU.Register.view	Register display	71
APU.RESet	Reset APU core	72
APU.Step	Single-stepping	72
APU.StepHll	HLL single-stepping	72
APU.View	Display APU peripherals	73
ART		74
ART	Trace method for Advanced Register Trace	74
ART-specific Trace Commands		76
ART.Mode	Set the trace operation mode	76

ART Trace Commands		77
ART.Arm	Arm the trace	77
ART.AutoArm	Arm automatically	77
ART.AutoInit	Automatic initialization	77
ART.BookMark	Set a bookmark in trace listing	77
ART.Chart	Display trace contents graphically	77
ART.ComPare	Compare trace contents	77
ART.DISable	Disable the trace	78
ART.DRAW	Plot trace data against time	78
ART.EXPORT	Export trace data for processing in other applications	78
ART.FILE	Load a file into the file trace buffer	78
ART.Find	Find specified entry in trace	78
ART.FindAll	Find all specified entries in trace	78
ART.FindChange	Search for changes in trace flow	78
ART.GOTO	Move cursor to specified trace record	78
ART.Init	Initialize trace	79
ART.List	List trace contents	79
ART.ListNesting	Analyze function nesting	79
ART.LOAD	Load trace file for offline processing	79
ART.OFF	Switch off	79
ART.PROfileChart	Profile charts	79
ART.PROTOcol	Protocol analysis	79
ART.PROTOcol.Chart	Graphic display for user-defined protocol	80
ART.PROTOcol.Draw	Graphic display for user-defined protocol	80
ART.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	80
ART.PROTOcol.Find	Find in trace buffer for user-defined protocol	80
ART.PROTOcol.list	Display trace buffer for user-defined protocol	80
ART.PROTOcol.PROfileChart	Profile chart for user-defined protocol	80
ART.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	80
ART.PROTOcol.STATistic	Display statistics for user-defined protocol	81
ART.REF	Set reference point for time measurement	81
ART.RESet	Reset command	81
ART.SAVE	Save trace for postprocessing in TRACE32	81
ART.SelfArm	Automatic restart of trace recording	81
ART.SIZE	Define buffer size	81
ART.SnapShot	Restart trace capturing once	81
ART.state	Display trace configuration window	81
ART.STATistic	Statistic analysis	82
ART.Timing	Waveform of trace buffer	82
ART.TRACK	Set tracking record	82
ART.View	Display single record	82
ART.ZERO	Align timestamps of trace and timing analyzers	82
AutoSTOre		83

AutoSTOre	Save and restore settings (history, GUI, etc.) automatically	83
AVX		85
AVX	AVX registers (Advanced Vector Extension)	85
AVX.Init	Initialize AVX registers	85
AVX.Set	Modify AVX registers	85
AVX.view	Display AVX registers	86
AVX512		87
AVX512	AVX512 registers (Advanced Vector Extension)	87
AVX512.Init	Initialize AVX512 registers	87
AVX512.Set	Modify AVX512 registers	87
AVX512.view	Display AVX512 registers	88

General Commands Reference Guide B

General Commands Reference Guide B	(general_ref_b.pdf)	1
History		8
BMC		9
BMC	Benchmark counters	9
BMC.<counter>	Benchmark counters	10
BMC.<counter>.EVENT	Assign event to counter	10
BMC.<counter>.FORMAT	Counter value format	10
BMC.<counter>.RATIO	Set two counters in relation	11
BMC.<counter>.SIZE	Specify counter size	11
BMC.Attach	BMC attach	12
BMC.Autolnit	Automatic initialization	12
BMC.CLOCK	Provide core clock for cycle counter	12
BMC.Init	Initialize counters	13
BMC.PROfile	Display counter changes per second	13
BMC.PROfileChart	Profile chart with benchmark counter	14
BMC.PROfileChart.AddressGROUP	Address group profile chart with BMC	14
BMC.PROfileChart.DatasYmbol	Pointer profile chart with BMC	15
BMC.PROfileChart.DistriB	Distribution display with BMC	15
BMC.PROfileChart.GROUP	Group profile chart with BMC	15
BMC.PROfileChart.Line	Source code line profile chart with BMC	16
BMC.PROfileChart.MODULE	Module profile chart with BMC	16
BMC.PROfileChart.PROGRAM	Program profile chart with BMC	17
BMC.PROfileChart.sYmbol	Symbol profile chart with BMC	18
BMC.PROfileChart.TASK	Task profile chart with BMC	18
BMC.PROfileChart.TASKINFO	Data trace via context ID with BMC	18
BMC.PROfileChart.TASKINTR	ISR2 profile chart with BMC	19
BMC.PROfileChart.TASKKernel	Task profile chart with BMC	19
BMC.PROfileChart.TASKORINTERRUPT	Task and interrupts with BMC	19

BMC.PROfileChart.TASKSRV	OS service routines profile chart with BMC	20
BMC.PROfileChart.TASKVSINTR	Task related intr. profile chart with BMC	20
BMC.PROfileSTATistic	Statistical analysis vs. time with benchmark counter	21
BMC.PROfileSTATistic.Address	Address statistical analysis with BMC	21
BMC.PROfileSTATistic.AddressGROUP	Address group statistic with BMC	22
BMC.PROfileSTATistic.DatasYmbol	Pointer profile statistic with BMC	22
BMC.PROfileSTATistic.DistriB	Distribution statistical analysis with BMC	22
BMC.PROfileSTATistic.GROUP	Group profile statistic with BMC	23
BMC.PROfileSTATistic.INTERRUPT	Interrupt profile statistic with BMC	23
BMC.PROfileSTATistic.Line	High-level code line profile statistic with BMC	24
BMC.PROfileSTATistic.MODULE	Module profile statistic with BMC	24
BMC.PROfileSTATistic.PROGRAM	Program profile statistic with BMC	24
BMC.PROfileSTATistic.RUNNABLE	Runnable profile statistic with BMC	25
BMC.PROfileSTATistic.sYmbol	Symbol profile statistic with BMC	25
BMC.PROfileSTATistic.TASK	Task profile statistic with BMC	25
BMC.PROfileSTATistic.TASKINFO	Data trace via context ID with BMC	26
BMC.PROfileSTATistic.TASKINTR	ISR2 profile statistic with BMC	26
BMC.PROfileSTATistic.TASKKernel	Task profile statistic with BMC	27
BMC.PROfileSTATistic.TASKORINTERRUPT	Task or interrupt with BMC	27
BMC.PROfileSTATistic.TASKSRV	OS service routines profile stat. with BMC	27
BMC.RESet	Reset benchmark counter configuration	29
BMC.SnoopSet	Assign event counter to SNOOPer trace	29
BMC.state	Display BMC configuration window	32
BMC.STATistic	Statistic analysis with benchmark counter	35
BMC.STATistic.ChildTREE	Function callee context with BMC	35
BMC.STATistic.DistriB	Distribution analysis with BMC	36
BMC.STATistic.Func	Nesting function run-time with BMC	36
BMC.STATistic.GROUP	Group run-time analysis with BMC	36
BMC.STATistic.LINKage	Per caller function statistic with BMC	37
BMC.STATistic.MODULE	Module statistic with BMC	37
BMC.STATistic.ParentTREE	Statistic for call context with BMC	37
BMC.STATistic.PROGRAM	Program statistic with BMC	38
BMC.STATistic.sYmbol	Flat run-time analysis with BMC	38
BMC.STATistic.TASK	Statistic for tasks with BMC	39
BMC.STATistic.TASKINFO	Statistic for context ID messages with BMC	39
BMC.STATistic.TASKINTR	Statistic for ISR2 with BMC	39
BMC.STATistic.TASKKernel	Statistic for tasks with BMC	40
BMC.STATistic.TASKORINTERRUPT	Tasks and interrupts with BMC	40
BMC.STATistic.TASKSRV	Statistic for OS service routines with BMC	41
BMC.STATistic.TREE	Tree nesting function run-time with BMC	41
BookMark		42
BookMark	Address and trace bookmarks	42
Overview BookMark		42

BookMark.CHange	Edit the settings of a bookmark	43
BookMark.Create	Create a new address bookmark	44
BookMark.Delete	Delete an existing bookmark	45
BookMark.EditRemark	Add/edit remark of a bookmark	46
BookMark.EXPORT	Export bookmarks	47
BookMark.EXPORT.ADDRESS	Export bookmarks for specified addresses	47
BookMark.EXPORT.preset	Export bookmarks to an XML file	47
BookMark.EXPORT.SOURCE	Export bookmarks for specified source files	49
BookMark.EXPORT.sYmbol	Export bookmarks for specified symbols	49
BookMark.List	List all bookmarks	51
BookMark.RESet	Delete all bookmarks	52
BookMark.Toggle	Toggles a single address bookmark	53
Break		54
Break	Stopping the program execution	54
Breakpoints		54
Break.Asm	Stop program/set temporary breakpoint and switch to Asm mode	56
Break.CLEAR	Reset complex triggers	57
Break.CONFIG	Configuration of breakpoint behavior and breakpoint scope	58
Break.CONFIG.AlwaysAlive	Alive Onchip breakpoints	58
Break.CONFIG.InexactAddress	Inexact address range breakpoint	58
Break.CONFIG.InexactData	Inexact data value breakpoint	59
Break.CONFIG.InexactResume	Resuming on inexact breakpoints	60
Break.CONFIG.InexactTrigger	Inexact trigger breakpoints	60
Break.CONFIG.MatchASID	Use ASID specific breakpoints	61
Break.CONFIG.MatchMachine	Use machine specific breakpoints	62
Break.CONFIG.MatchZone	Use zone specific breakpoints	62
Break.CONFIG.METHOD	Breakpoints implementation	64
Break.CONFIG.state	Breakpoint configuration window	65
Break.CONFIG.UseContextID	Context ID specific breakpoints	65
Break.CONFIG.UseMachineID	Machine ID specific breakpoints	66
Break.CONFIG.VarConvert	Convert breakpoints on scalar variables	68
Break.Delete	Delete breakpoints	69
Break.DeletePATtern	Delete breakpoints allowing wildcards	70
Break.direct	Stop program execution or set temporary breakpoints	71
Break.DISable	Disable breakpoints	73
Break.ENABLE	Enable breakpoints	74
Break.Hll	Stop program/set temporary breakpoint and switch to HLL mode	75
Break.Init	Initialize breakpoints	76
Break.List	Display list of breakpoints	76
Break.Mix	Stop program/set temporary breakpoint and switch to MIX mode	78
Break.MONitor	Switch back to stop mode debugging	79
Break.PASS	Define pass condition for breakpoint	79
Break.PATtern	Set temporary breakpoints allowing wildcards	80

Break.Program	CTL interactive programming	80
Break.ReProgram	Activate existing CTL program file	81
Break.REQuest	Request a program break	81
Break.RESet	Delete all breakpoints and reset the TRACE32 break system	81
Break.Set	Set breakpoints	82
Break.SetFunc	Mark HLL functions	106
Break.SetLine	Mark HLL lines	108
Break.SetMONitor	Switch to run mode debugging at the next “Go”	108
Break.SetPATtern	Set breakpoints allowing wildcards	108
Break.SetTask	Stop the program execution when task is scheduled	110
Break.ViewProgram	Show state of the CTL trigger unit	110
BSDL		111
BSDL	Boundary scan description language	111
BSDL.BYPASSall	Check bypass mode	112
BSDL.CHECK	Enable test result checking	112
BSDL.FILE	Load a BSDL file	112
BSDL.FLASH	Flash programming	113
BSDL.FLASH.IFCheck	Check flash interface definition	113
BSDL.FLASH.IFDefine	Define flash interface	115
BSDL.FLASH.IFMap	Map flash interface	116
BSDL.FLASH.INIT	Initialize flash interface	117
BSDL.HARDRESET	TAP reset via TRST	117
BSDL.IDCODEall	Check ID codes	118
BSDL.LINKAGE	Create a bypass device	118
BSDL.LoadDR	Load data register from file	119
BSDL.MOVEDOWN	Move selected chip downwards	120
BSDL.MOVEUP	Move selected chip upwards	121
BSDL.ParkState	Select JTAG parking state	121
BSDL.RESet	Reset boundary scan configuration	122
BSDL.RUN	Run JTAG sequence	122
BSDL.RUNTCK	Toggle TCK	122
BSDL.SAMPLEall	Sample all signals	123
BSDL.SELect	Select a chip	123
BSDL.SET	Set chip parameters	124
BSDL.SetAndRun	Immediate data register takeover	130
BSDL.SOFTRESET	TAP reset via TMS	131
BSDL.state	Display BSDL chain configuration window	132
BSDL.StepPauseDR	Special DR shift	133
BSDL.SToreDR	Store data register to file	134
BSDL.TwoStepDR	Single/double data register shift	135
BSDL.UNLOAD	Unload a chip from chain	135
BTrace		136
BTrace	Script-controlled trace sink	136

BTrace-specific Trace Commands	137
BTrace.<specific_cmds>	Overview of BTrace-specific commands 137
BTrace.Mode	Set the trace operation mode 137
BTrace.PUSH	Push trace data 137
BTrace.state	Display BTrace configuration window 140
Generic BTrace Trace Commands	141
BTrace.Arm	Arm the trace 141
BTrace.AutoArm	Arm automatically 141
BTrace.AutoInit	Automatic initialization 141
BTrace.BookMark	Set a bookmark in trace listing 141
BTrace.Chart	Display trace contents graphically 141
BTrace.ComPare	Compare trace contents 141
BTrace.DISable	Disable the trace 142
BTrace.DRAW	Plot trace data against time 142
BTrace.EXPORT	Export trace data for processing in other applications 142
BTrace.FILE	Load a file into the file trace buffer 142
BTrace.Find	Find specified entry in trace 142
BTrace.FindAll	Find all specified entries in trace 142
BTrace.FindChange	Search for changes in trace flow 142
BTrace.GOTO	Move cursor to specified trace record 142
BTrace.Init	Initialize trace 143
BTrace.List	List trace contents 143
BTrace.ListNesting	Analyze function nesting 143
BTrace.LOAD	Load trace file for offline processing 143
BTrace.OFF	Switch off 143
BTrace.PROfileChart	Profile charts 143
BTrace.PROTOcol	Protocol analysis 143
BTrace.PROTOcol.Chart	Graphic display for user-defined protocol 144
BTrace.PROTOcol.Draw	Graphic display for user-defined protocol 144
BTrace.PROTOcol.EXPORT	Export trace buffer for user-defined protocol 144
BTrace.PROTOcol.Find	Find in trace buffer for user-defined protocol 144
BTrace.PROTOcol.list	Display trace buffer for user-defined protocol 144
BTrace.PROTOcol.PROfileChart	Profile chart for user-defined protocol 144
BTrace.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol 144
BTrace.PROTOcol.STATistic	Display statistics for user-defined protocol 145
BTrace.REF	Set reference point for time measurement 145
BTrace.RESet	Reset command 145
BTrace.SAVE	Save trace for postprocessing in TRACE32 145
BTrace.SIZE	Define buffer size 145
BTrace.STATistic	Statistic analysis 145
BTrace.Timing	Waveform of trace buffer 145
BTrace.TRACK	Set tracking record 145
BTrace.View	Display single record 146

General Commands Reference Guide C

General Commands Reference Guide C	(general_ref_c.pdf)	1
History		12
CACHE		13
CACHE	View and modify CPU cache contents	13
CACHE.CLEAN	Clean CACHE	13
CACHE.ComPare	Compare CACHE with memory	14
CACHE.DUMP	Dump CACHE	15
CACHE.FLUSH	Clean and invalidate CACHE	16
CACHE.GET	Get CACHE contents	17
CACHE.INFO	View all information related to an address	17
CACHE.INVALIDATE	Invalidate CACHE	18
CACHE.List	List CACHE contents	18
CACHE.ListFunc	List cached functions	19
CACHE.ListLine	List cached source code lines	20
CACHE.ListModule	List cached modules	20
CACHE.ListVar	List cached variables	21
CACHE.LOAD	Load previously stored cache contents	22
CACHE.RELOAD	Reload previously loaded cache contents	22
CACHE.SAVE	Save cache contents for postprocessing	22
CACHE.SNAPSHOT	Take cache snapshot for comparison	23
CACHE.UNLOAD	Unload previously loaded cache contents	24
CACHE.view	Display cache control register	25
CAnalyzer		26
CAnalyzer	Trace features of Compact Analyzer	26
CAnalyzer - Compact Analyzer specific Trace Commands		28
CAnalyzer.<specific_cmds>	Overview of CAnalyzer-specific commands	28
CAnalyzer.CLOCKDelay	Set clock delay	28
CAnalyzer.CLOSE	Close named pipes	28
CAnalyzer.DecodeMode	Define how to decode the received trace data	29
CAnalyzer.I2C	I2C control	30
CAnalyzer.PipeLOAD	Load a previously saved file	30
CAnalyzer.PipeRePlay	Replay a previously recorded stream	30
CAnalyzer.PipeSAVE	Define a file that stores received data	31
CAnalyzer.PipeWRITE	Define a named pipe as trace sink	31
CAnalyzer.SAMPLE	Set sample time offset	32
CAnalyzer.ShowFocus	Display data eye	33
CAnalyzer.ShowFocusClockEye	Show clock eye	36
CAnalyzer.ShowFocusEye	Show data eyes	37

CAalyzer.TERMination	Configure parallel trace termination	39
CAalyzer.TOut	Route trigger to PODBUS (CombiProbe/μTrace)	39
CAalyzer.TraceCLOCK	Configure the trace port frequency	40
CAalyzer.TracePORT	Select which trace port is used	41
CAalyzer.WRITE	Define a file as trace sink	42
Generic CAalyzer Trace Commands		43
CAalyzer.ACCESS	Define access path to program code for trace decoding	43
CAalyzer.Arm	Arm the trace	43
CAalyzer.AutoArm	Arm automatically	43
CAalyzer.AutoFocus	Calibrate AUTOFOCUS preprocessor	43
CAalyzer.AutoInit	Automatic initialization	43
CAalyzer.BookMark	Set a bookmark in trace listing	43
CAalyzer.BookMarkToggle	Toggles a single trace bookmark	44
CAalyzer.Chart	Display trace contents graphically	44
CAalyzer.CLOCK	Clock to calculate time out of cycle count information	44
CAalyzer.ComPar	Compare trace contents	44
CAalyzer.ComParCODE	Compare trace with memory	44
CAalyzer.CustomTrace	Custom trace	44
CAalyzer.CustomTraceLoad	Load a DLL for trace analysis/Unload all DLLs	44
CAalyzer.DISable	Disable the trace	45
CAalyzer.DRAW	Plot trace data against time	45
CAalyzer.EXPORT	Export trace data for processing in other applications	45
CAalyzer.ExtractCODE	Extract code from trace	45
CAalyzer.FILE	Load a file into the file trace buffer	45
CAalyzer.Find	Find specified entry in trace	45
CAalyzer.FindAll	Find all specified entries in trace	45
CAalyzer.FindChange	Search for changes in trace flow	45
CAalyzer.FindProgram	Advanced trace search	46
CAalyzer.FindReProgram	Activate advanced existing trace search program	46
CAalyzer.FindViewProgram	State of advanced trace search programming	46
CAalyzer.FLOWPROCESS	Process flowtrace	46
CAalyzer.FLOWSTART	Restart flowtrace processing	46
CAalyzer.Get	Display input level	46
CAalyzer.GOTO	Move cursor to specified trace record	46
CAalyzer.Init	Initialize trace	47
CAalyzer.JOINFILE	Concatenate several trace recordings	47
CAalyzer.List	List trace contents	47
CAalyzer.ListNesting	Analyze function nesting	47
CAalyzer.ListVar	List variable recorded to trace	47
CAalyzer.LOAD	Load trace file for offline processing	47
CAalyzer.MERGEFILE	Combine two trace files into one	47
CAalyzer.Mode	Set the trace operation mode	47
CAalyzer.OFF	Switch off	48

CAalyzer.PortFilter	Specify utilization of trace memory	48
CAalyzer.PortType	Specify trace interface	48
CAalyzer.PROfileChart	Profile charts	48
CAalyzer.PROfileSTATistic	Statistical analysis in a table versus time	48
CAalyzer.PROTOcol	Protocol analysis	48
CAalyzer.PROTOcol.Chart	Graphic display for user-defined protocol	48
CAalyzer.PROTOcol.Draw	Graphic display for user-defined protocol	49
CAalyzer.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	49
CAalyzer.PROTOcol.Find	Find in trace buffer for user-defined protocol	49
CAalyzer.PROTOcol.list	Display trace buffer for user-defined protocol	49
CAalyzer.PROTOcol.PROfileChart	Profile chart for user-defined protocol	49
CAalyzer.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	49
CAalyzer.PROTOcol.STATistic	Display statistics for user-defined protocol	49
CAalyzer.REF	Set reference point for time measurement	50
CAalyzer.RESet	Reset command	50
CAalyzer.SAVE	Save trace for postprocessing in TRACE32	50
CAalyzer.SelfArm	Automatic restart of trace recording	50
CAalyzer.SIZE	Define buffer size	50
CAalyzer.SnapShot	Restart trace capturing once	50
CAalyzer.SPY	Adaptive stream and analysis	50
CAalyzer.state	Display trace configuration window	50
CAalyzer.STATistic	Statistic analysis	51
CAalyzer.STREAMCompression	Select compression mode for streaming	51
CAalyzer.STREAMFILE	Specify temporary streaming file path	51
CAalyzer.STREAMFileLimit	Set size limit for streaming file	51
CAalyzer.STREAMLOAD	Load streaming file from disk	51
CAalyzer.STREAMSAVE	Save streaming file to disk	51
CAalyzer.TDelay	Trigger delay	51
CAalyzer.TestFocus	Test trace port recording	52
CAalyzer.TestFocusClockEye	Scan clock eye	52
CAalyzer.TestFocusEye	Check signal integrity	52
CAalyzer.TestUtilization	Tests trace port utilization	52
CAalyzer.THreshold	Optimize threshold for trace lines	52
CAalyzer.Timing	Waveform of trace buffer	52
CAalyzer.TraceCONNECT	Select on-chip peripheral sink	52
CAalyzer.TRACK	Set tracking record	53
CAalyzer.TSElect	Select trigger source	53
CAalyzer.View	Display single record	53
CAalyzer.ZERO	Align timestamps of trace and timing analyzers	53
CIProbe		54
CIProbe	Trace with Analog Probe and CombiProbe/?Trace (MicroTrace)	54
CIProbe-specific Trace Commands		56
CIProbe.<specific_cmds>	Overview of CIProbe-specific commands	56

CIProbe.ALOWerLIMit	Set lower trigger/filter comparator value	56
CIProbe.ATrigEN	Enable/disable trigger contribution of a channel	56
CIProbe.ATrigMODE	Set trigger/filter condition	58
CIProbe.AUPPerLIMit	Set upper trigger/filter comparator value	59
CIProbe.Mode	Set trace operation mode	59
CIProbe.state	Display CIProbe configuration window	60
CIProbe.TDelay	Define trigger delay	60
CIProbe.TOut	Route CIProbe trigger to PODBUS	61
CIProbe.TSElect	Route PODBUS trigger to CIProbe	62
CIProbe.TSYNC.SElect	Select trigger input pin and edge or state	62
Generic CIProbe Trace Commands		64
CIProbe.Arm	Arm the trace	64
CIProbe.AutoArm	Arm automatically	64
CIProbe.AutoInit	Automatic initialization	64
CIProbe.BookMark	Set a bookmark in trace listing	64
CIProbe.BookMarkToggle	Toggles a single trace bookmark	64
CIProbe.Chart	Display trace contents graphically	64
CIProbe.ComPare	Compare trace contents	65
CIProbe.DISable	Disable the trace	65
CIProbe.DisConfig	Trace disassembler configuration	65
CIProbe.DRAW	Plot trace data against time	65
CIProbe.EXPORT	Export trace data for processing in other applications	65
CIProbe.FILE	Load a file into the file trace buffer	65
CIProbe.Find	Find specified entry in trace	65
CIProbe.FindAll	Find all specified entries in trace	65
CIProbe.FindChange	Search for changes in trace flow	66
CIProbe.Get	Display input level	66
CIProbe.GOTO	Move cursor to specified trace record	66
CIProbe.Init	Initialize trace	66
CIProbe.List	List trace contents	66
CIProbe.ListNesting	Analyze function nesting	66
CIProbe.ListVar	List variable recorded to trace	66
CIProbe.LOAD	Load trace file for offline processing	66
CIProbe.OFF	Switch off	67
CIProbe.PROfile	Rolling live plots of trace data	67
CIProbe.PROfile.channel	Display profile of signal probe channels	67
CIProbe.PROfileChart	Profile charts	67
CIProbe.PROfileSTATistic	Statistical analysis in a table versus time	67
CIProbe.PROTOcol	Protocol analysis	67
CIProbe.PROTOcol.Chart	Graphic display for user-defined protocol	67
CIProbe.PROTOcol.Draw	Graphic display for user-defined protocol	68
CIProbe.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	68
CIProbe.PROTOcol.Find	Find in trace buffer for user-defined protocol	68

CIProbe.PROTOcol.list	Display trace buffer for user-defined protocol	68
CIProbe.PROTOcol.PROfileChart	Profile chart for user-defined protocol	68
CIProbe.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	68
CIProbe.PROTOcol.STATistic	Display statistics for user-defined protocol	68
CIProbe.REF	Set reference point for time measurement	69
CIProbe.RESet	Reset command	69
CIProbe.SAVE	Save trace for postprocessing in TRACE32	69
CIProbe.SIZE	Define buffer size	69
CIProbe.SnapShot	Restart trace capturing once	69
CIProbe.SPY	Adaptive stream and analysis	69
CIProbe.STATistic	Statistic analysis	69
CIProbe.STREAMCompression	Select compression mode for streaming	69
CIProbe.STREAMFILE	Specify temporary streaming file path	70
CIProbe.STREAMFileLimit	Set size limit for streaming file	70
CIProbe.Timing	Waveform of trace buffer	70
CIProbe.TRACK	Set tracking record	70
CIProbe.View	Display single record	70
CIProbe.ZERO	Align timestamps of trace and timing analyzers	70
ClipStore		71
ClipSTOre	Store settings to clipboard	71
CLOCK		72
CLOCK	Display date and time	72
CLOCK.BACKUP	Set backup clock frequency	72
CLOCK.DATE	Alias for DATE command	73
CLOCK.OFF	Disable clock frequency computation	73
CLOCK.ON	Enable clock frequency computation	73
CLOCK.OSCillator	Set board oscillator frequency	74
CLOCK.Register	Display PLL related registers	74
CLOCK.RESet	Reset CLOCK command group settings	74
CLOCK.state	Display clock frequencies	75
CLOCK.SYSCLock	Set external clock frequency	75
CLOCK.VCOBase	Set 'VCOBase' clock frequency	76
CLOCK.VCOBaseERAY	Set 'FlexRay VCOBase' clock frequency	76
CMI		77
CMI	Clock management interface	77
CMN		78
CMN	Coherent mesh network	78
CMN<trace> - Trace Data Analysis		79
CMN<trace>	Command groups for CMN<trace>	79
Overview CMN<trace>		79
CMNAnalyzer	Analyze CMN information recorded by TRACE32 PowerTrace	80
CMNCAalyzer	Analyze CMN information recorded by CombiProbe	80

CMNHAnalyzer	Analyze CMN information captured by the host analyzer	81
CMNLA	Analyze CMN information from binary source	81
CMNOnchip	Analyze CMN information captured in target onchip memory	81
CORE		83
CORE	Cores in an SMP system	83
Overview CORE		83
CORE.ADD	Add core/thread to the SMP system	84
CORE.ASSIGN	Assign a set of physical cores/threads to the SMP system	85
CORE.List	List information about cores	91
CORE.NUMber	Assign a number of cores/threads to the SMP system	92
CORE.ReMove	Remove core from the SMP system	93
CORE.select	Change currently selected core	93
CORE.SHOWACTIVE	Show active/inactive cores in an SMP system	94
CORE.SINGLE	Select single core for debugging	95
Count		97
Count	Universal counter	97
Overview Count		97
Count.AutoInit	Automatic counter reset	99
Count.Gate	Gate time	99
Count.GO	Start measurement	100
Count.Init	Reset counter	100
Count.Mode	Mode selection	101
Count.OUT	Forward counter input signal to trigger system/output	103
Count.PROfile	Graphic counter display	103
Count.RESet	Reset command	105
Count.Select	Select input source	105
Count.state	State display	106
COverage		107
COverage	Trace-based code coverage	107
COverage.ADD	Add trace contents to code coverage system	107
COverage.Delete	Set code coverage tagging to never	108
COverage.EXPORT	Export code coverage information	109
COverage.EXPORT.CBA	Export coverage results in CBA format	110
COverage.EXPORT.CSV	Export coverage results in CSV format	111
COverage.EXPORT.JSON	Export code coverage results in JSON format	111
COverage.EXPORT.JSONE	Export code coverage in extended JSON format	112
COverage.EXPORT.ListCalleEs	Export the function callees	113
COverage.EXPORT.ListCalleEs.<sub_cmd>	Export callees information	114
COverage.EXPORT.ListCalleRs	Export the function callers	115
COverage.EXPORT.ListCalleRs.<sub_cmd>	Export callers information	116
COverage.EXPORT.ListFunc	Export code coverage results at function level	117
COverage.EXPORT.ListFunc.<sub_cmd>	Export function	117

COVerage.EXPORT.ListInlineBlock	Export inlined code blocks	122
COVerage.EXPORT.ListInlineBlock.<sub_cmd>	Export cov. inlined	123
COVerage.EXPORT.ListLine	Export HLL lines	124
COVerage.EXPORT.ListLine.<sub_cmd>	Export HLL lines information	125
COVerage.EXPORT.ListModule	Export modules	126
COVerage.EXPORT.ListModule.<sub_cmd>	Export modules information	126
COVerage.EXPORT.ListVar	Export HLL variables	127
COVerage.EXPORT.ListVar.<sub_cmd>	Export HLL variables information	127
COVerage.INFO	Information about conditional instructions	128
COVerage.Init	Clear coverage database	129
COVerage.List	Coverage display	129
COVerage.ListCalleEs	Display coverage for callees function	130
COVerage.ListCalleEs.<sub_cmd>	Display coverage for callees function	130
COVerage.ListCalleRs	Display coverage for callers function	133
COVerage.ListCalleRs.<sub_cmd>	Display coverage for callers function	133
COVerage.ListFunc	Display coverage for functions	136
COVerage.ListFunc.<sub_cmd>	Display coverage for HLL function	136
COVerage.ListInlineBlock	Display coverage for inlined block	140
COVerage.ListInlineBlock.<sub_cmd>	Display coverage for inlined block	140
COVerage.ListLine	Display coverage for HLL lines	143
COVerage.ListLine.<sub_cmd>	Display coverage for HLL lines	143
COVerage.ListModule	Display coverage for modules	145
COVerage.ListModule.<sub_cmd>	Display coverage for modules	145
COVerage.ListVar	Display coverage for variable	148
COVerage.ListVar.<sub_cmd>	Display coverage for variables	148
COVerage.LOAD	Load coverage database from file	151
COVerage.MAP	Map the coverage to a different range	152
COVerage.METHOD	Select code coverage method	153
COVerage.Mode	Activate code coverage for virtual targets	154
COVerage.OFF	Deactivate coverage	154
COVerage.ON	Activate coverage	155
COVerage.Option	Set coverage options	156
COVerage.Option.BLOCKMode	Enable/disable line block mode	156
COVerage.Option.ITrace	Enable instruction trace processing	157
COVerage.Option.SourceMetric	Select code coverage metric	157
COVerage.Option.StaticInfo	Perform code coverage precalculations	159
COVerage.RESet	Clear coverage database	160
COVerage.SAVE	Save coverage database to file	160
COVerage.Set	Coverage modification	161
COVerage.state	Configure coverage	162
COVerage.TreeWalkSETUP	Prepare a tree with code coverage symbols	163
COVerage.TreeWalkSETUP.<sub_cmd>	Prepare a coverage symbol tree	163
CTS		165

CTS	Context tracking system (CTS)	165
Trace-based Debugging		166
Full High-Level Language Trace Display		167
Reconstruction of Trace Gaps (TRACE32-ICD)		167
CTS Commands		168
CTS.CACHE	CTS cache analysis	168
CTS.CACHE.Allocation	Define the cache allocation technique	170
CTS.CACHE.CYcles	Define counting method for cache analysis	171
CTS.CACHE.DefineBus	Define bus interface	171
CTS.CACHE.L1Architecture	Define architecture for L1 cache	173
CTS.CACHE.LFSR	Linear-feedback shift register for random generator	173
CTS.CACHE.ListAddress	Address based cache analysis	174
CTS.CACHE.ListFunc	Function based cache analysis	175
CTS.CACHE.ListLine	HLL line based cache analysis	176
CTS.CACHE.ListModules	Module based cache analysis	176
CTS.CACHE.ListRequests	Display request for a single cache line	177
CTS.CACHE.ListSet	Cache set based cache analysis	178
CTS.CACHE.ListVar	Variable based cache analysis	178
CTS.CACHE.MMUArchitecture	Define MMU architecture for cache control	179
CTS.CACHE.Mode	Define memory coherency strategy	180
CTS.CACHE.Replacement	Define the replacement strategy	181
CTS.CACHE.RESet	Reset settings of CTS cache window	182
CTS.CACHE.SETS	Define the number of cache sets	182
CTS.CACHE.Sort	Define sorting for all list commands	182
CTS.CACHE.state	Display settings of CTS cache analysis	183
CTS.CACHE.Tags	Define address mode for cache lines	184
CTS.CACHE.TLBArchitecture	Define architecture for the TLB	185
CTS.CACHE.View	Display the results for the cache analysis	186
CTS.CACHE.ViewBPU	Display statistic for branch prediction unit	190
CTS.CACHE.ViewBus	Display statistics for the bus utilization	191
CTS.CACHE.ViewStalls	Display statistics for idles/stalls	192
CTS.CACHE.WAYS	Define number of cache ways	193
CTS.CACHE.Width	Define width of cache line	194
CTS.CAPTURE	Copy real memory to the virtual memory for CTS	194
CTS.Chart.ChildTREE	Display callee context of a function as chart	195
CTS.Chart.Func	Function activity chart	195
CTS.Chart.INTERRUPT	Display interrupt chart	195
CTS.Chart.INTERRUPTTREE	Display interrupt nesting	196
CTS.Chart.Nesting	Show function nesting at cursor position	196
CTS.Chart.RUNNABLE	Runnable activity chart	196
CTS.Chart.sYmbol	Execution time at different symbols as chart	197
CTS.Chart.TASK	Task activity chart	197
CTS.Chart.TASKINFO	Chart for context ID special messages	198

CTS.Chart.TASKINTR	Display ISR2 time chart (ORTI)	198
CTS.Chart.TASKKernel	Display task time chart with kernel markers (ORTI)	198
CTS.Chart.TASKORINTERRUPT	Task and interrupt activity chart	199
CTS.Chart.TASKSRV	Service routine run-time analysis	199
CTS.Chart.TASKVSINTERRUPT	Time chart of interrupted tasks	199
CTS.Chart.TASKVSINTR	Time chart of task-related interrupts	200
CTS.Chart.TREE	Display function chart as tree view	200
CTS.EXPORT	Export trace data	201
CTS.FixedControl	Execution time at different symbols as chart	201
CTS.GOTO	Select the specified record for CTS (absolute)	201
CTS.INCremental	CTS displays intermediate results while processing	202
CTS.Init	Restart CTS processing	202
CTS.List	List trace contents	203
CTS.ListNesting	Analyze function nesting	205
CTS.Mode	Operation mode	205
CTS.OFF	Switch off trace-based debugging	206
CTS.ON	Switch on trace-based debugging	206
CTS.PROCESS	Process cache analysis	206
CTS.PROfileChart	Profile charts	207
CTS.PROfileChart.CACHE	Display cache analysis results graphically	207
CTS.PROfileChart.sYmbol	Dynamic program behavior as profile chart	208
CTS.PROfileChart.TASK	Task profile chart	209
CTS.PROfileChart.TASKINFO	Profile chart for context ID special messages	209
CTS.PROfileChart.TASKINTR	ISR2 profile chart	209
CTS.PROfileChart.TASKKernel	Task profile chart with kernel markers	210
CTS.PROfileChart.TASKORINTERRUPT	Task and interrupt profile chart	210
CTS.PROfileChart.TASKSRV	OS service routines profile chart	210
CTS.PROfileChart.TASKVSINTR	Task-related interrupts profile chart	211
CTS.RESet	Reset the CTS settings	212
CTS.SELectiveTrace	Trace contains selective trace information	212
CTS.SKIP	Select the specified record for CTS (relative)	212
CTS.SmartTrace	CTS smart trace	213
CTS.state	Display CTS settings	214
CTS.STATistic	Nesting function runtime analysis	216
CTS.STATistic.ChildTREE	Show callee context of a function	216
CTS.STATistic.Func	Nesting function runtime analysis	216
CTS.STATistic.GROUP	Group run-time analysis	217
CTS.STATistic.INTERRUPT	Interrupt statistic	217
CTS.STATistic.INTERRUPTTREE	Interrupt nesting	217
CTS.STATistic.LINKage	Per caller statistic of function	218
CTS.STATistic.MODULE	Code execution broken down by module	218
CTS.STATistic.ParentTREE	Show the call context of a function	218
CTS.STATistic.PROGRAM	Code execution broken down by program	219

CTS.STATistic.RUNNABLE	Runnable runtime analysis	219
CTS.STATistic.sYmbol	Flat run-time analysis	219
CTS.STATistic.TASK	Task statistic	220
CTS.STATistic.TASKINFO	Statistic for context ID special messages	220
CTS.STATistic.TASKINTR	ISR2 statistic (ORTI)	220
CTS.STATistic.TASKKernel	Task statistic with kernel markers	221
CTS.STATistic.TASKORINTERRUPT	Task and interrupt statistic	221
CTS.STATistic.TASKSRV	OS service routines statistic	221
CTS.STATistic.TASKVSINTERRUPT	Statistic of interrupts, task-related	222
CTS.STATistic.TREE	Tree display of nesting function run-time analysis	222
CTS.TAKEOVER	Take memory/registers reconstructed by CTS over to target	223
CTS.UNDO	Revert last CTS command	223
CTS.UseConst	Use constants for the CTS processing	223
CTS.UseDataTrace	Use sampling cycles for CTS	224
CTS.UseFinalContext	Use the CPU registers for CTS	224
CTS.UseFinalMemory	Use memory contents for CTS	225
CTS.UseSIM	Use instruction set simulator for CTS	226
CTS.UseStartMemory	Use virtual memory contents as initial values for CTS	227

General Commands Reference Guide D

General Commands Reference Guide D	(general_ref_d.pdf)	1
History		8
Data		9
Data	Memory access	9
Overview Data		9
Data.AllocList	Static memory allocation analysis	13
Data.Assemble	Built-in assembler	18
Data.ATTACH	Attach data sequence	20
Data.ATTACH.CONDition	Define attach condition	21
Data.ATTACH.CORE	Select core for attach sequence	21
Data.ATTACH.OFF	Switch attach sequence off	22
Data.ATTACH.ON	Switch attach sequence on	22
Data.ATTACH.RESet	Reset attach data sequence	22
Data.ATTACH.SELect	Increment the index number to the next sequence	23
Data.ATTACH.SEQuence	Define attach data sequence	24
Data.ATTACH.state	Attach data state display	24
Data.BDTAB	Display buffer descriptor table	25
Data.BENCHMARK	Determine cache/memory bandwidth	26
Data.CHAIN	Display linked list	30
Data.CHAINFind	Search in linked list	33
Data.CLEARVM	Clear the TRACE32 virtual memory (VM:)	34
Data.ComPare	Compare memory	35

Data.COPY	Copy memory	37
Data.CSA	Display linked list of CSA entries	39
Data.DRAW	Graphical memory display of arrays	40
Data.DRAFFT	Graphical display of fast fourier transformation	44
Data.DRAWXY	Graphical display of xy-graphs	47
Data.dump	Memory dump	50
Data.EPILOG	Automatic data modification on program execution halt	62
Data.EPILOG.CONDition	Define condition for data epilog	63
Data.EPILOG.CORE	Select core for data epilog	64
Data.EPILOG.OFF	Switch data epilog off	64
Data.EPILOG.ON	Switch data epilog on	65
Data.EPILOG.RESet	Reset all data epilogs	65
Data.EPILOG.SELect	Increment the index number to the next data epilog	66
Data.EPILOG.SEQuence	Define epilog sequence	67
Data.EPILOG.state	Display data epilogs	68
Data.EPILOG.TARGET	Define epilog target call	69
Data.Find	Search in memory	70
Data.FindCODE	Execute command on specified code type	72
Data.GOTO	Specify reference address for address tracking	73
Data.GREP	Search for string	75
Data.IMAGE	Display image data	76
Data.In	Read port	80
Data.LOAD	Load file	81
Format Specific Data.LOAD Commands and Options		95
Data.LOAD.AIF	Load Arm image file	95
Data.LOAD.AOUT	Load a.out file	96
Data.LOAD.ASAP2	Load ASAP2 file	96
Data.LOAD.Ascii	Load ASCII file	97
Data.LOAD.AsciiDump	Load ASCII file generated from Data.dump window	97
Data.LOAD.AsciiHex	Load hex file	98
Data.LOAD.AsciiOct	Load octal file	98
Data.LOAD.AVocet	Load AVOCET file	99
Data.LOAD.BDX	Load BDX file	99
Data.LOAD.Binary	Load binary file	100
Data.LOAD.Bound	Load BOUND file	103
Data.LOAD.CCSDAT	Load CCSDAT file	103
Data.LOAD.CDB	Load SDCC CDB file format	104
Data.LOAD.COFF	Load COFF file	106
Data.LOAD.ColonHex	Load colon hex file	107
Data.LOAD.COMFOR	Load COMFOR (TEKTRONIX) file	108
Data.LOAD.CORE	Load Linux core dump file	109
Data.LOAD.COSMIC	Load COSMIC file	109
Data.LOAD.CrashDump	Load MS Windows Crash Dump file	110

Data.LOAD.DAB	Load DAB file	111
Data.LOAD.DBX	Load a.out file	112
Data.LOAD.Elif	Load ELF file	113
Data.LOAD.ESTFB	Load EST flat binary	125
Data.LOAD.eXe	Load EXE file	125
Data.LOAD.FIASCO	Load FIASCO BB5 file	127
Data.LOAD.HiCross	Load HICROSS file	127
Data.LOAD.HiTech	Load HITECH file	128
Data.LOAD.HP	Load HP-64000 file	129
Data.LOAD.ICoff	Load ICOFF file	130
Data.LOAD.Ieee	Load IEEE-695 file	131
Data.LOAD.IntelHex	Load INTEL-HEX file	133
Data.LOAD.LDR	Load META-LDR file	133
Data.LOAD.MachO	Load 'Mach-O' file	134
Data.LOAD.MAP	Load MAP file	136
Data.LOAD.MCDS	Load MCDS file	137
Data.LOAD.MCoff	Load MCOFF file	137
Data.LOAD.OAT	Load OAT file	138
Data.LOAD.Omf	Load OMF file	139
Data.LOAD.Omf2	Load OMF-251 files	142
Data.LOAD.OriginHex	Load special hex files	142
Data.LOAD.PureHex	Load hex-byte file	143
Data.LOAD.REAL	Load R.E.A.L. file	143
Data.LOAD.ROF	Load OS-9 file	144
Data.LOAD.S1record	Load S1-Record file	145
Data.LOAD.S2record	Load S2-Record file	146
Data.LOAD.S3record	Load S3-Record file	146
Data.LOAD.S4record	Load S4-Record file	147
Data.LOAD.SAUF	Load SAUF file	147
Data.LOAD.SDS	Load SDSI file	148
Data.LOAD.SPARSE	Load SPARSE file	148
Data.LOAD.sYm	Load symbol file	149
Data.LOAD.SysRof	Load RENESAS SYSROF file	150
Data.LOAD.TEK	Load TEKTRONIX file	151
Data.LOAD.TekHex	Load TEKTRONIX HEX file	151
Data.LOAD.Ubrof	Load UBROF file	152
Data.LOAD.VersaDos	Load VERSADOS file	153
Data.LOAD.XCoff	Load XCOFF file	153
Data.MSYS	M-SYSTEMS FLASHDISK support	154
Data.Out	Write port	154
Data.PATTERN	Fill memory with pattern	155
Data.Print	Display multiple areas	158
Data.PROfile	Graphical display of data value	161

Data.PROGRAM	Editor for writing assembler program	163
Data.PROLOG	Automatic data modification on program execution start	164
Data.PROLOG.CONDITION	Define PROLOG condition	165
Data.PROLOG.CORE	Select core for data prolog	166
Data.PROLOG.OFF	Switch data prolog off	166
Data.PROLOG.ON	Switch data prolog on	167
Data.PROLOG.RESet	Reset all data prologs	167
Data.PROLOG.SELect	Increment the index number to the next data prolog	168
Data.PROLOG.SEQuence	Define prolog sequence	169
Data.PROLOG.state	Display data prologs	170
Data.PROLOG.TARGET	Define PROLOG target call	171
Data.REF	Display current values	172
Data.ReProgram	Assemble instructions into memory	173
Data.ReRoute	Reroute function call	173
Data.SAVE.<format>	Save data in file with specified format	174
Data.SAVE.Ascii	Save ASCII file	176
Data.SAVE.AsciiHex	Save hex file	176
Data.SAVE.AsciiOct	Save octal file	178
Data.SAVE.BDX	Save BDX file	179
Data.SAVE.Binary	Save binary file	179
Data.SAVE.CCSDAT	Save CCSDAT file	180
Data.SAVE.DAB	Save DAB file	180
Data.SAVE.ElF	Save ELF file	181
Data.SAVE.ESTFB	Save EST flat binary file	181
Data.SAVE.IntelHex	Save INTEL-HEX file	182
Data.SAVE.OmF	Save OMF file	182
Data.SAVE.PureHex	Save pure HEX file	183
Data.SAVE.S1record	Save S1-record file	184
Data.SAVE.S2record	Save S2-record file	186
Data.SAVE.S3record	Save S3-record file	186
Data.SAVE.S4record	Save S4-record file	187
Data.Set	Modify memory	188
Data.SOFTEPILOG	Automated sequence after setting software breakp.	191
Data.SOFTEPILOG.CONDITION	Define condition for data softepilog	192
Data.SOFTEPILOG.CORE	Select core for data softepilog	192
Data.SOFTEPILOG.OFF	Switch data softepilog off	193
Data.SOFTEPILOG.ON	Switch data softepilog on	193
Data.SOFTEPILOG.RESet	Reset all data softepilogs	193
Data.SOFTEPILOG.SELect	Increment the index number to the next epilog	193
Data.SOFTEPILOG.SEQuence	Define softepilog sequence	194
Data.SOFTEPILOG.state	Display data softepilogs	194
Data.SOFTPROLOG	Automated sequence before setting software breakp.	195
Data.SOFTPROLOG.CONDITION	Define condition for data softprolog	196

Data.SOFTPROLOG.CORE	Select core for data softprolog	197
Data.SOFTPROLOG.OFF	Switch data softprolog off	197
Data.SOFTPROLOG.ON	Switch data softprolog on	197
Data.SOFTPROLOG.RESet	Reset all data softprolog	198
Data.SOFTPROLOG.SELect	Increment the index number to the next prolog	198
Data.SOFTPROLOG.SEQuence	Define softprolog sequence	198
Data.SOFTPROLOG.state	Display data softprologs	199
Data.STANDBY	Standby data-sequences	200
Data.STANDBY.CONDition	Define condition	202
Data.STANDBY.CORE	Assign sequence to core	203
Data.STANDBY.OFF	Switch all sequences off	203
Data.STANDBY.ON	Switch all sequences on	204
Data.STANDBY.RESet	Clear all settings	204
Data.STANDBY.SELect	Increment index number for next sequence	205
Data.STANDBY.SEQuence	Define sequence	206
Data.STANDBY.state	Open configuration window	207
Data.STARTUP	Startup data sequence	208
Data.STARTUP.CONDition	Define startup condition	209
Data.STARTUP.CORE	Select core for startup sequence	210
Data.STARTUP.OFF	Switch startup sequence off	211
Data.STARTUP.ON	Switch startup data sequence on	211
Data.STARTUP.RESet	Reset startup data sequence	211
Data.STARTUP.SELect	Increment the index number to the next sequence	212
Data.STARTUP.SEQuence	Define startup data sequence	213
Data.STARTUP.state	Startup data state display	214
Data.STRING	ASCII display	215
Data.SUM	Memory checksum	215
Data.TABLE	Display arrays	218
Data.TAG	Tag code for analysis	221
Data.TAGFunc	Tag code for analysis	221
Data.Test	Memory integrity test	223
Data.TestList	Test for memory type	226
Data.TIMER	Periodical data sequence	227
Data.TIMER.CONDition	Define timer condition	228
Data.TIMER.CORE	Select core for timer sequence	229
Data.TIMER.ERRORSTOP	Stop data timer on errors	230
Data.TIMER.OFF	Switch timer off	230
Data.TIMER.ON	Switch timer on	230
Data.TIMER.RESet	Reset timer	231
Data.TIMER.SELect	Increment the index number to the next sequence	231
Data.TIMER.SEQuence	Define timer sequence	232
Data.TIMER.state	Timer state display	233
Data.TIMER.TARGET	Define timer target call	234

Data.TIMER.Time	Define period for timer	234
Data.UNTAGFunc	Remove code tags	235
Data.UPDATE	Target memory cache update	235
Data.USRACCESS	Prepare USR access	236
Data.VECTOR	Display memory as vectors	237
Data.View	Display memory	239
Data.WRITESTRING	Write string to PRACTICE file	241
DCI		242
DCI	Direct Connect Interface (DCI)	242
DQMTTrace		243
DTM		244
DTM	DTM trace sources (Data Trace Module)	244
DTM.CLOCK	Set core clock frequency for timing measurements	244
DTM.CycleAccurate	Cycle accurate tracing	244
DTM.Mode	Define DTM mode	244
DTM.OFF	Disable DTM	245
DTM.ON	Enable DTM	245
DTM.Register	Display DTM registers	245
DTM.RESet	Reset DTM settings	245
DTM.TraceID	Change the default ID for a DTM trace source	246
DTM.TracePriority	Define priority of DTM	246
DTM<trace> - Trace Data Analysis		247
DTM<trace>	Command groups for DTM<trace>	247
Overview DTM<trace>		247
DTMAnalyzer	Analyze DTM information recorded by TRACE32 PowerTrace	248
DTMCAalyzer	Analyze DTM information recorded by CombiProbe	248
DTMHAalyzer	Analyze DTM information captured by the host analyzer	249
DTMLA	Analyze DTM information from binary source	249
DTMONchip	Analyze DTM information captured in target onchip memory	250
DTMTrace	Method-independent analysis of DTM trace data	250

General Commands Reference Guide E

General Commands Reference Guide E	(general_ref_e.pdf)	1
ELA		8
ELA	Embedded logic analyzer (ELA)	8
Overview ELA		8
ELA.ATBTrigger	Use ATB to transfer trace trigger to trace sink	11
ELA.CLEAR	Clear ELA.Set settings	11
ELA.CLOCK	ELA sample rate	11
ELA.OFF	Switch ELA off	12
ELA.ON	Switch ELA on	12

ELA.PortRoute	Set up trace hardware	12
ELA.Register	Display the ELA registers	13
ELA.RESet	Reset ELA settings	13
ELA.SELect	Select signal group	14
ELA.Set	Set ELA registers	14
ELA.state	Display ELA configuration window	21
ELA.SyncPeriod	Set synchronization frequency	21
ELA.TimeStampCLOCK	External clock frequency	22
ELA.TimeStamps	Emit global timestamp packets	22
ELA.TimeStampThreshold	Set granularity for occurrence of timestamps	23
ELA.Trace	Control generation of trace information	23
ELA.TraceID	Change the default ID for an ELA trace source	24
ELA.TracePREDICT	Enable/disable prediction	24
ELA.TracePriority	Define priority of ELA	24
ELA<trace> - Trace Data Analysis		25
ELAAalyzer	Analyze ELA information recorded by TRACE32 PowerTrace	25
ELACAnalyzer	Analyze ELA information recorded by Compact Analyzer	25
ELAHAnalyzer	Analyze ELA information captured by the host analyzer	26
ELALA	Analyze ELA information from binary source	26
ELAOnchip	Analyze ELA information captured in target onchip memory	26
ELATrace	Method-independent analysis of ELA trace data	27
ETA		28
ETA	Energy test analysis for energy profiling	28
Overview ETA		28
ETA.DRAW	Line chart	33
ETA.List	Lists the ETA trace data	36
ETA.ListNesting	Displays the function call nesting	37
ETA.PROfileChart	Power consumption by function as function of time	38
ETA.PROfileChart.AddressGROUP	Energy per GROUP graphically	39
ETA.PROfileChart.DatasYmbol	Symbolic statistics for data as a chart	40
ETA.PROfileChart.DistriB	Graphical distribution analysis	41
ETA.PROfileChart.GROUP	Energy per GROUP graphically	42
ETA.PROfileChart.Line	Energy per high-level language line graphically	43
ETA.PROfileChart.MODULE	Energy per module graphically	43
ETA.PROfileChart.POWER	Power consumption per channel graphically	44
ETA.PROfileChart.PROGRAM	Energy per program graphically	47
ETA.PROfileChart.sYmbol	Energy for all program symbols graphically	47
ETA.PROfileChart.TASK	Energy consumption per task graphically	49
ETA.PROfileChart.TASKINFO	Energy per data trace message via context ID	49
ETA.PROfileChart.TASKINTR	Energy consumption per ISR2 graphically	50
ETA.PROfileChart.TASKKernel	Energy consumption per ISR2 graphically	50
ETA.PROfileChart.TASKORINTERRUPT	Energy per task/interrupt	51
ETA.PROfileChart.TASKSRV	Energy consumption per service routine	51

ETA.PROfileChart.TASKVSINTR	Energy per task-related interrupts	52
ETA.PROfileSTATistic	Energy analysis in a table versus time	53
ETA.PROfileSTATistic.Address	Statistics about addresses	53
ETA.PROfileSTATistic.AddressGROUP	Energy per GROUP as a table	54
ETA.PROfileSTATistic.DatasYmbol	Statistics about data symbols	54
ETA.PROfileSTATistic.DistriB	Distribution statistical analysis	56
ETA.PROfileSTATistic.GROUP	Energy per GROUP as a table	56
ETA.PROfileSTATistic.INTERRUPT	Energy per interrupt as a table	58
ETA.PROfileSTATistic.Line	Energy per high-level language line as a table	58
ETA.PROfileSTATistic.MODULE	Energy per module as a table	59
ETA.PROfileSTATistic.PROGRAM	Energy per program as a table	59
ETA.PROfileSTATistic.RUNNABLE	Energy per runnable as a table	60
ETA.PROfileSTATistic.sYmbol	Energy for all program symbols as a table	60
ETA.PROfileSTATistic.TASK	Energy consumption per TASK as a table	62
ETA.PROfileSTATistic.TASKINFO	Energy per data trace via context ID	62
ETA.PROfileSTATistic.TASKINTR	Energy statistics about ISR2 as a table	63
ETA.PROfileSTATistic.TASKKernel	Energy consumption as a table	63
ETA.PROfileSTATistic.TASKORINTERRUPT	Energy per task/interrupt	64
ETA.PROfileSTATistic.TASKSRV	Energy analysis of service routines	64
ETA.RESet	Reset command	65
ETA.SELect	Select the power channels to be analyzed	66
ETA.state	Opens the ETA configuration window	69
ETA.STATistic	Statistical energy analysis	71
ETA.STATistic.ChildTREE	All children of a function as a tree	72
ETA.STATistic.DistriB	Distribution analysis	73
ETA.STATistic.Func	Function energy analysis	74
ETA.STATistic.GROUP	Group analysis	75
ETA.STATistic.LINKage	Linkage analysis	77
ETA.STATistic.MODULE	Module analysis	78
ETA.STATistic.ParentTREE	Parents of a function	79
ETA.STATistic.PROGRAM	Program analysis	80
ETA.STATistic.sYmbol	Statistical analysis of energy consumption	81
ETA.STATistic.TASK	Task energy analysis	83
ETA.STATistic.TASKINFO	Energy per data trace message via context ID	84
ETA.STATistic.TASKINTR	Energy of interrupt service routines	85
ETA.STATistic.TASKKernel	Energy consumption of tasks and kernel	86
ETA.STATistic.TASKORINTERRUPT	Task/interrupt energy analysis	87
ETA.STATistic.TASKSRV	Energy analysis of service routines	88
ETA.STATistic.TREE	Energy analysis as tree	89
ETM		90
EVENTS		91
EVENTS.List	List the events trace data	91
EVENTS.ListNesting	Show program nesting	92

EVENTS.PROfileChart	Profile chart for events	93
EVENTS.PROfileChart.AddressGROUP	Event profile chart for groups	93
EVENTS.PROfileChart.ALL	Event profile chart for program run	94
EVENTS.PROfileChart.DatasYmbol	Symbolic statistics for data as a chart	94
EVENTS.PROfileChart.DistriB	Distribution statistical analysis	94
EVENTS.PROfileChart.GROUP	Event profile chart for groups	95
EVENTS.PROfileChart.Line	Events per high-level language line graphically	95
EVENTS.PROfileChart.MODULE	Event profile chart for modules	96
EVENTS.PROfileChart.PROGRAM	Event profile chart for programs	96
EVENTS.PROfileChart.sYmbol	Event for all program symbols graphically	96
EVENTS.PROfileChart.TASK	Events per task graphically	97
EVENTS.PROfileChart.TASKINFO	Events per context ID message	98
EVENTS.PROfileChart.TASKINTR	Events profile chart for ISR2 (ORTI)	98
EVENTS.PROfileChart.TASKKernel	Event profile chart with kernel marker	99
EVENTS.PROfileChart.TASKORINTERRUPT	EVENTS per task/interrupt	99
EVENTS.PROfileChart.TASKSRV	Events for OS service routines	100
EVENTS.PROfileChart.TASKVSINTR	Events for task-related interrupts	100
EVENTS.PROfileSTATistic	Profile statistics for events	101
EVENTS.PROfileSTATistic.Address	Events per address as profile statistic	101
EVENTS.PROfileSTATistic.AddressGROUP	Events per address GROUP	102
EVENTS.PROfileSTATistic.ALL	Event profile statistic for program run	102
EVENTS.PROfileSTATistic.DatasYmbol	Symbolic statistics for data	103
EVENTS.PROfileSTATistic.DistriB	Distribution statistical analysis	103
EVENTS.PROfileSTATistic.GROUP	Events per GROUP as profile statistic	104
EVENTS.PROfileSTATistic.INTERRUPT	Events per interrupt as table	104
EVENTS.PROfileSTATistic.Line	Events per high-level language line as table	104
EVENTS.PROfileSTATistic.MODULE	Events per module as profile statistic	105
EVENTS.PROfileSTATistic.PROGRAM	Events per program	105
EVENTS.PROfileSTATistic.RUNNABLE	Events per runnable as table	106
EVENTS.PROfileSTATistic.sYmbol	Events for all program symbols as table	106
EVENTS.PROfileSTATistic.TASK	Events per task as table	107
EVENTS.PROfileSTATistic.TASKINFO	Events per context ID message	107
EVENTS.PROfileSTATistic.TASKINTR	Events per ISR2 (ORTI) as table	108
EVENTS.PROfileSTATistic.TASKKernel	Events per task as table	108
EVENTS.PROfileSTATistic.TASKORINTERRUPT	Events per task as table	109
EVENTS.PROfileSTATistic.TASKSRV	Events per OS service routine	109
EVENTS.STATistic	Statistic for events	110
EVENTS.STATistic.ChildTREE	Events for the callee context of a function	110
EVENTS.STATistic.Func	Events for functions numerically	110
EVENTS.STATistic.GROUP	Events statistic for groups	110
EVENTS.STATistic.LINKAge	Per caller event statistic of function	110
EVENTS.STATistic.MODULE	Events for modules numerically	111
EVENTS.STATistic.ParentTREE	Event statistic for call context of a function	111

EVENTS.STATistic.PROGRAM	Events for programs numerically	111
EVENTS.STATistic.sYmbol	Events for all program symbols numerically	111
EVENTS.STATistic.TASK	Events per task numerically	112
EVENTS.STATistic.TASKINFO	Events per context ID message numerically	112
EVENTS.STATistic.TASKINTR	Events per ISR2 numerically	112
EVENTS.STATistic.TASKKernel	Events task analysis with kernel markers	112
EVENTS.STATistic.TASKSRV	Events per OS service routine numerically	113
EVENTS.STATistic.TREE	Tree display of nesting functions with events	113
EXTension		114
EXTension	Extend the TRACE32 debugger with custom features	114
EXTension.ACCESS	Control memory access	114
EXTension.CONFIG	Configure extension	115
EXTension.DEBUG	Debug outputs of extension	115
EXTension.DELETE	Delete loaded extension	115
EXTension.LOAD	Load extension	116
EXTension.MaxVSize	Set max. vertical size of extension windows	117
EXTension.ORTI.Delete	Unload ORTI file	117
EXTension.ORTI.LOAD	Load ORTI file	118
EXTension.ORTI.RESet	Unload all ORTI files	119
EXTension.RESet	Reset extension definition	119
EXTension.SETDIR	Set the extension directory	120
EXTension.TimeOut	Set timeout of extension	120

General Commands Reference Guide F

General Commands Reference Guide F	(general_ref_f.pdf)	1
History		7
FDX		8
FDX	Trace method FDX	8
FDX-specific Command		9
FDX.ADDRESS	Specify memory space for FDX traces	9
FDX.CLEAR	Clear FDX communication buffers	9
FDX.CLOSE	Close FDX files	9
FDX.DISableChannel	Disable FDX communication	10
FDX.ENABLEChannel	Enable FDX communication	10
FDX.InChannel	Inchannel state display	11
FDX.METHOD	Select communication channel	11
FDX.Mode	Set the trace operation mode	13
FDX.Out	Send FDX data	13
FDX.OutChannel	Outchannel state display	14
FDX.PipeREAD	Define named pipe for input channel	14
FDX.PipeWRITE	Define named pipe for output channel	15

FDX.Rate	Select sampling rate	15
FDX.READ	Define FDX input file	15
FDX.Timestamp	Configure timestamp usage of FDX trace	16
FDX.TraceChannel	Define FDX trace channel	16
FDX.WRITE	Define FDX output file	17
Generic FDX Trace Commands		18
FDX.Arm	Arm the trace	18
FDX.AutoArm	Arm automatically	18
FDX.AutoInit	Automatic initialization	18
FDX.BookMark	Set a bookmark in trace listing	18
FDX.Chart	Display trace contents graphically	18
FDX.ComPare	Compare trace contents	18
FDX.DISable	Disable the trace	19
FDX.DRAW	Plot trace data against time	19
FDX.EXPORT	Export trace data for processing in other applications	19
FDX.FILE	Load a file into the file trace buffer	19
FDX.Find	Find specified entry in trace	19
FDX.FindAll	Find all specified entries in trace	19
FDX.FindChange	Search for changes in trace flow	19
FDX.GOTO	Move cursor to specified trace record	19
FDX.Init	Initialize trace	20
FDX.List	List trace contents	20
FDX.ListNesting	Analyze function nesting	20
FDX.ListVar	List variable recorded to trace	20
FDX.LOAD	Load trace file for offline processing	20
FDX.OFF	Switch off	20
FDX.PROfileChart	Profile charts	20
FDX.PROTOcol	Protocol analysis	20
FDX.PROTOcol.Chart	Graphic display for user-defined protocol	21
FDX.PROTOcol.Draw	Graphic display for user-defined protocol	21
FDX.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	21
FDX.PROTOcol.Find	Find in trace buffer for user-defined protocol	21
FDX.PROTOcol.list	Display trace buffer for user-defined protocol	21
FDX.PROTOcol.PROfileChart	Profile chart for user-defined protocol	21
FDX.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	21
FDX.PROTOcol.STATistic	Display statistics for user-defined protocol	22
FDX.REF	Set reference point for time measurement	22
FDX.RESet	Reset command	22
FDX.SAVE	Save trace for postprocessing in TRACE32	22
FDX.SelfArm	Automatic restart of trace recording	22
FDX.SIZE	Define buffer size	22
FDX.SnapShot	Restart trace capturing once	22
FDX.state	Display trace configuration window	22

FDX.STATistic	Statistic analysis	23
FDX.Timing	Waveform of trace buffer	23
FDX.Timing	Waveform of trace buffer	23
FDX.TRACK	Set tracking record	23
FDX.View	Display single record	23
FDX.ZERO	Align timestamps of trace and timing analyzers	23
FIFO		24
FIFO	Display on-chip trace FIFO	24
FLASH		25
FLASH	Memory mapped FLASH memories	25
FLASH.AUTO	Auto programming of FLASH	25
FLASH.BSDLaccess	Enables FLASH access via boundary scan	27
FLASH.CFI	Generate FLASH declaration by CFI	27
FLASH.CHANGEtype	Changes the FLASH type	32
FLASH.CLock	Setup input clock for processor internal flash	33
FLASH.Create	Declare FLASH	34
FLASH.CreateALIAS	Create address alias	39
FLASH.Delete	Delete entry in FLASH declaration table	40
FLASH.EPILOG	Automatic data modification on FLASH operation	41
FLASH.EPILOG.CONDition	Define condition for FLASH epilog	41
FLASH.EPILOG.CORE	Select core for FLASH epilog	41
FLASH.EPILOG.OFF	Switch FLASH epilog off	42
FLASH.EPILOG.ON	Switch FLASH epilog on	42
FLASH.EPILOG.RESet	Reset all FLASH epilogs	42
FLASH.EPILOG.SELect	Increment the index number to the next epilog	43
FLASH.EPILOG.SEQuence	Define FLASH epilog sequence	43
FLASH.EPILOG.state	Display FLASH epilogs	44
FLASH.Erase	Erase FLASH	45
FLASH.GETID	Get FLASH IDs	46
FLASH.HOOKSCRIPT	PRACTICE script based FLASH programming prolog	47
FLASH.List	Display FLASH definition table	49
FLASH.LOCK	Lock FLASH	50
FLASH.MultiProgram	Simultaneous programming of flash sectors	52
FLASH.OFFSET	Change FLASH control address	52
FLASH.Program	Program FLASH	53
FLASH.PROLOG	Automatic data modification on FLASH operation	55
FLASH.PROLOG.CONDition	Define condition for FLASH prolog	55
FLASH.PROLOG.CORE	Select core for FLASH prolog	55
FLASH.PROLOG.OFF	Switch FLASH prolog off	56
FLASH.PROLOG.ON	Switch FLASH prolog on	56
FLASH.PROLOG.RESet	Reset all FLASH prologs	56
FLASH.PROLOG.SELect	Increment the index number to the next prolog	57
FLASH.PROLOG.SEQuence	Define FLASH prolog sequence	57

FLASH.PROLOG.state	Display FLASH prologs	58
FLASH.ReProgram	Re-program FLASH	59
FLASH.RESet	Reset FLASH declaration table	60
FLASH.SPI	FLASH SPI command group	61
FLASH.SPI.CFI	Generate SPI FLASH sector declaration by CFI	61
FLASH.SPI.CMD	Send data to SPI FLASH device	63
FLASH.SPI.GETSFDP	Read FLASH discovery parameters	66
FLASH.SPI.RESetMemory	Reset SPI FLASH volatile register	66
FLASH.state	FLASH programming dialog	67
FLASH.TARGET	Define target controlled algorithm	68
FLASH.TARGET2	Define second target controlled algorithm	75
FLASH.UNLOCK	Unlock FLASH	76
FLASH.UNSECUREerase	Unsecure a device	78
FLASHFILE		79
FLASHFILE	Non-memory mapped FLASH devices	79
FLASHFILE.BSDLaccess	Enables FLASH access via boundary scan	80
FLASHFILE.BSDLFLASHTYPE	Define FLASH type	80
FLASHFILE.CONFIG	Inform TRACE32 about the FLASH register addresses	81
FLASHFILE.COPY	Copy to FLASH	82
FLASHFILE.COPYSPARE	Copy to spare area of NAND FLASH	82
FLASHFILE.Create	Declaration of flash memories: create a block/sector	84
FLASHFILE.Delete	Delete block in FLASH declaration table	85
FLASHFILE.DUMP	Dump FLASH	85
FLASHFILE.Erase	Erase FLASH	86
FLASHFILE.GETBADBLOCK	Get the bad block addresses	87
FLASHFILE.GETEXTCSD	Get the extended CSD register	87
FLASHFILE.GETID	Get ID values of FLASH device	88
FLASHFILE.GETONFI	Display ONFI	88
FLASHFILE.List	List blocks or sectors of FLASH memory	89
FLASHFILE.LOAD	Load files to FLASH	90
FLASHFILE.LOAD.binary	Write FLASH	90
FLASHFILE.LOAD.Elf	Load ELF file	93
FLASHFILE.LOAD.IntelHex	Load Intel hex file	93
FLASHFILE.LOAD.JSON	Load "flasher_args.json" file	94
FLASHFILE.LOAD.SPARSE	Load SPARSE file	95
FLASHFILE.LOAD.Srecord	Load an 'Srecord' file	96
FLASHFILE.LOADALL	Load to main area and spare area	96
FLASHFILE.LOADECC	Load ECC file to spare area	97
FLASHFILE.LOADSPARE	Write NAND FLASH spare area	97
FLASHFILE.LOCK	Lock the FLASH device	98
FLASHFILE.MMC.GETHealth	eMMC health state	99
FLASHFILE.MSYSDLL	Access an M-Systems DiskOnChip flash device	99
FLASHFILE.PATTERN	Erase and fill flash memory	99

FLASHFILE.ReProgram	Re-program FLASH	100
FLASHFILE.RESet	Reset FLASHFILE declaration within TRACE32	101
FLASHFILE.SAVE	Save FLASH	102
FLASHFILE.SAVEALL	Save the main area and the spare area	102
FLASHFILE.SAVEECC	Save error correction code (ECC) to file	103
FLASHFILE.SAVEECC.BCH	Save ECC with BCH algorithm	103
FLASHFILE.SAVEECC.hamming	Save ECC with Hamming algorithm	106
FLASHFILE.SAVEECC.ReedSolomon	Save ECC with Reed-S. algorithm	109
FLASHFILE.SAVESPARE	Read NAND FLASH spare area	111
FLASHFILE.Set	Modify FLASH data	111
FLASHFILE.SETEXTCSD	Modify the extended CSD register	112
FLASHFILE.SPI	FLASHFILE SPI command group	113
FLASHFILE.SPI.CFI	Generate SPI FLASH sector declaration by CFI	113
FLASHFILE.SPI.CMD	Send data to SPI FLASH device	114
FLASHFILE.SPI.GETSFDP	Read FLASH discovery parameters	116
FLASHFILE.SPI.RESetMemory	Reset volatile register values	116
FLASHFILE.TARGET	Define target controlled algorithm	117
FLASHFILE.TEST	Non-memory mapped FLASH test	118
FLASHFILE.UNLOCK	Unlock FLASH device	119
FPU		120
FPU	Access to FPU registers	120
FPU.Init	Initialize FPU registers	120
FPU.OFF	FPU access off	121
FPU.ON	FPU access on	121
FPU.RESet	Reset command	121
FPU.Set	Modify FPU registers	121
FPU.TARGET	Define FPU access agent	122
FPU.view	Display FPU registers	122
Frame		123
Frame	Call-tree and context	123
Frame.CONFIG	Fine-tune stack unwinding	123
Frame.CONFIG.Asm	Frame back-trace mode	123
Frame.CONFIG.EABI	Use chained frame pointers	124
Frame.CONFIG.EPILOG	Use epilog code for frame display	124
Frame.CONFIG.PROLOG	Use prolog code for frame display	125
Frame.CONFIG.RELOAD	Generate frame information again	125
Frame.CONFIG.SignalHandler	Stack unwinding	125
Frame.CONFIG.sYmbol	Use symbol code for frame display	126
Frame.COPY	Copy to TRACE32 registers	127
Frame.Down	Show state one level down in stack nesting	127
Frame.GOTO	Change source code view temporarily	127
Frame.Init	Initialize the processor registers	128
Frame.REDO	Recover from UNDO registers	131

Frame.SkipFunc	Change view to previous/subsequent function	131
Frame.SkipLine	Change view to previous/subsequent HLL line	132
Frame.SWAP	Swap TRACE32 registers	132
Frame.TASK	Change view to specified task	132
Frame.UNDO	Recover previous registers	134
Frame.Up	Show state one level up in stack nesting	134
Frame.view	Display stack frame	136
FXU		139
FXU	FXU registers (extended floating point unit)	139
FXU.Init	Initialize FXU registers	139
FXU.Set	Modify FXU registers	139
FXU.view	Open FXU register window	140

General Commands Reference Guide G

General Commands Reference Guide G	(general_ref_g.pdf)	1
History		4
GLOBALON		5
GLOBALON	Global event-controlled PRACTICE script execution	5
Go		10
Go	Debug control, program execution, and real-time emulation	10
Debug Control for Debuggers		10
Go.Asm	Start the program execution and switch to Asm mode	11
Go.Back	Run backwards (CTS)	13
Go.BackEntry	Run backwards until function entry (CTS)	14
Go.BackTillWarning	Run backwards until warning (CTS)	15
Go.Change	Run program until content changes	15
Go.direct	Start the program execution	16
Go.Hll	Start the program execution and switch to HLL mode	18
Go.Java	Run program until JAVA code starts	19
Go.Mix	Start the program execution and switch to 'Mix' mode	20
Go.MONitor	Switch to run mode debugging	21
Go.Next	Run program until next line	21
Go.Return	Run to function epilog	22
Go.Till	Run program until expression becomes true	25
Go.TillWarning	Run program until warning (CTS)	26
Go.Up	Run up to caller	27
GROUP		28
GROUP	Group functions, modules, or tasks	28
Features		28
GROUP.COLOR	Define color for group indicator	32
GROUP.Create	Create a new group	33

GROUP.CreateFunctions	Pool functions to group	34
GROUP.CreateLabels	Use labels to pool address ranges to group	35
GROUP.CreateModules	Pool modules to group	37
GROUP.CreatePrograms	Pool programs group	38
GROUP.CreateSources	Pool source files to group	39
GROUP.CreateTASK	Pool tasks to group	40
GROUP.Delete	Delete the specified group	43
GROUP.DeleteTASK	Delete specified task from group	43
GROUP.DISable	Disable a group	44
GROUP.ENable	Enable a group	45
GROUP.HIDE	Hide group from debugging	45
GROUP.List	List all specified groups	46
GROUP.Merge	Merge group members in statistic	46
GROUP.RESet	Clear all group specifications	47
GROUP.SEParate	Separate group members in statistic	47
GROUP.SHOW	Show group for debugging	48

General Commands Reference Guide H

General Commands Reference Guide H	(general_ref_h.pdf)	1
HAnalyzer		6
HAnalyzer	Host analyzer	6
HAnalyzer-specific Trace Command		7
HAnalyzer.Mode	Set the trace operation mode	7
HAnalyzer.PipeWRITE	Define a named pipe as trace sink	7
HAnalyzer.state	Display HAnalyzer trace configuration window	8
Generic HAnalyzer Trace Commands		9
HAnalyzer.ACCESS	Define access path to program code for trace decoding	9
HAnalyzer.Arm	Arm the trace	9
HAnalyzer.AutoArm	Arm automatically	9
HAnalyzer.AutoInit	Automatic initialization	9
HAnalyzer.BookMark	Set a bookmark in trace listing	9
HAnalyzer.BookMarkToggle	Toggles a single trace bookmark	9
HAnalyzer.Chart	Display trace contents graphically	10
HAnalyzer.CLOCK	Clock to calculate time out of cycle count information	10
HAnalyzer.ComPare	Compare trace contents	10
HAnalyzer.ComPareCODE	Compare trace with memory	10
HAnalyzer.DISable	Disable the trace	10
HAnalyzer.DRAW	Plot trace data against time	10
HAnalyzer.EXPORT	Export trace data for processing in other applications	10
HAnalyzer.FILE	Load a file into the file trace buffer	10
HAnalyzer.Find	Find specified entry in trace	11
HAnalyzer.FindAll	Find all specified entries in trace	11

HAnalyzer.FindChange	Search for changes in trace flow	11
HAnalyzer.FindProgram	Advanced trace search	11
HAnalyzer.FindReProgram	Activate advanced existing trace search program	11
HAnalyzer.FindViewProgram	State of advanced trace search programming	11
HAnalyzer.FLOWPROCESS	Process flowtrace	11
HAnalyzer.FLOWSTART	Restart flowtrace processing	12
HAnalyzer.Get	Display input level	12
HAnalyzer.GOTO	Move cursor to specified trace record	12
HAnalyzer.Init	Initialize trace	12
HAnalyzer.List	List trace contents	12
HAnalyzer.ListNesting	Analyze function nesting	12
HAnalyzer.ListVar	List variable recorded to trace	12
HAnalyzer.LOAD	Load trace file for offline processing	12
HAnalyzer.OFF	Switch off	13
HAnalyzer.PROfileChart	Profile charts	13
HAnalyzer.PROfileSTATistic	Statistical analysis in a table versus time	13
HAnalyzer.REF	Set reference point for time measurement	13
HAnalyzer.RESet	Reset command	13
HAnalyzer.SAVE	Save trace for postprocessing in TRACE32	13
HAnalyzer.SIZE	Define buffer size	13
HAnalyzer.STATistic	Statistic analysis	13
HAnalyzer.Timing	Waveform of trace buffer	14
HAnalyzer.TraceCONNECT	Select on-chip peripheral sink	14
HAnalyzer.TRACK	Set tracking record	14
HAnalyzer.View	Display single record	14
HAnalyzer.ZERO	Align timestamps of trace and timing analyzers	14
HTM - Configuration of the Trace Source		15
HTM	CoreSight HTM (AHB Trace Macrocell)	15
HTM.AsicControl	Set HTMASICCONTROL register	15
HTM.AuxTrace	Auxiliary packet control	16
HTM.AXIFifoClock	AXI FIFO clock for WPT HTM	16
HTM.AXIMaster	AXI master for WPT HTM	16
HTM.BusSelect	Set HTMBUSSELECT register	17
HTM.BusTrigger	Bus trigger definition	17
HTM.CLEAR	Clear HTM.Set settings	17
HTM.CLOCK	Core clock frequency	18
HTM.CycleAccurate	Cycle accurate tracing	18
HTM.DataTrace	Define broadcast of data accesses	19
HTM.ExtDisable	Set EXTDISABLE bit	19
HTM.FifoLevel	Define FIFO level	20
HTM.OFF	Switch HTM off	20
HTM.ON	Switch HTM on	20
HTM.PortRoute	Set up trace hardware	21

HTM.Register	Display HTM control registers	22
HTM.RESet	Reset HTM settings	22
HTM.Set	Program HTM manually	23
HTM.state	Display HTM configuration window	23
HTM.SyncPeriod	Set period of sync packet injection	24
HTM.Trace	Trace packet control	24
HTM.TraceExclude	No broadcast of data accesses within range	25
HTM.TraceID	Set trace ID manually	25
HTM.TraceInclude	Restrict broadcast of data accesses to range	26
HTM.TraceOFF	HTM stops to emit trace information on event	26
HTM.TraceON	HTM starts to emit trace information on event	27
HTM.TracePriority	Set priority for the HTM manually	28
HTM.TraceTrigger	Trace trigger definition	28
HTM<trace> - Trace Data Analysis		29
HTM<trace>	Command groups for HTM<trace>	29
Overview HTM<trace>		29
HTMAnalyzer	Analyze HTM information recorded by TRACE32 PowerTrace	30
HTMCAalyzer	Analyze HTM info. recorded by TRACE32 CombiProbe	30
HTMHAnalyzer	Analyze HTM info. recorded by TRACE32 host analyzer	31
HTMLA	HTM logic analyzer	31
HTMOnchip	Analyze HTM information captured in target onchip memory	32
HTMTrace	Method-independent analysis of HTM trace data	32
HVX		33
HVX	HVX registers (Hexagon Vector Extensions)	33
HVX.Init	Initialize HVX registers	33
HVX.OFF	Inhibit HVX accesses by the debugger	33
HVX.ON	Permit HVX accesses by the debugger	34
HVX.Set	Modify HVX registers	34
HVX.view	Open HVX register window	34

General Commands Reference Guide I

General Commands Reference Guide I	(general_ref_i.pdf)	1
History		8
I2C		9
I2C	I2C control	9
I2C.PIN	Set I2C pin to specified level	9
I2C.THreshold	Specify threshold for logical low	10
I2C.TRANSFER	Transfer bytes on I2C bus	10
I2C.TransferRAW	Transfer bytes on I2C bus	11
Integrator		13
Integrator	Integrator logic analyzer	13

Integrator-specific Trace Commands	14
Integrator.ABCDEF	Sampling configuration for probes ABCDEF 14
Integrator.Break	Stop trace 15
Integrator.CSElect	Select signal for counter 15
Integrator.DisConfig.LOAD	Load DLL for protocol analysis 15
Integrator.JKLMNO	Sampling configuration for probes JKLMNO 16
Integrator.Mode	Set the trace operation mode 17
Integrator.Program	Program trigger unit 18
Integrator.ReProgram	Program trigger unit 18
Integrator.TOut	Enable trigger output line 18
Integrator.TPreDelay	Pre-trigger delay 19
Integrator.TSYNC	Select trigger line and mode 20
Integrator.TWidth	Set trigger filter 21
Generic Integrator Trace Commands	22
Integrator.ACCESS	Define access path to program code for trace decoding 22
Integrator.Arm	Arm the trace 22
Integrator.AutoArm	Arm automatically 22
Integrator.AutoFocus	Calibrate AUTOFOCUS preprocessor 22
Integrator.AutoInit	Automatic initialization 22
Integrator.BookMark	Set a bookmark in trace listing 22
Integrator.Chart	Display trace contents graphically 23
Integrator.ComPare	Compare trace contents 23
Integrator.DISable	Disable the trace 23
Integrator.DisConfig	Trace disassembler configuration 23
Integrator.DisConfig.CYcle	Trace disassemble setting 23
Integrator.DisConfig.FlowMode	Enable FlowTrace analysis 23
Integrator.DisConfig.RESet	Reset trace disassemble setting 23
Integrator.DRAW	Plot trace data against time 24
Integrator.EXPORT	Export trace data for processing in other applications 24
Integrator.FILE	Load a file into the file trace buffer 24
Integrator.Find	Find specified entry in trace 24
Integrator.FindAll	Find all specified entries in trace 24
Integrator.FindChange	Search for changes in trace flow 24
Integrator.Get	Display input level 24
Integrator.GOTO	Move cursor to specified trace record 24
Integrator.Init	Initialize trace 25
Integrator.List	List trace contents 25
Integrator.ListNesting	Analyze function nesting 25
Integrator.ListVar	List variable recorded to trace 25
Integrator.LOAD	Load trace file for offline processing 25
Integrator.OFF	Switch off 25
Integrator.PROfileChart	Profile charts 25
Integrator.PROTOcol	Protocol analysis 25

Integrator.PROTOcol.Chart	Graphic display for user-defined protocol	26
Integrator.PROTOcol.Draw	Graphic display for user-defined protocol	26
Integrator.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	26
Integrator.PROTOcol.Find	Find in trace buffer for user-defined protocol	26
Integrator.PROTOcol.list	Display trace buffer for user-defined protocol	26
Integrator.PROTOcol.PROfileChart	Profile chart for user-defined protocol	26
Integrator.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	26
Integrator.PROTOcol.STATistic	Display statistics for user-defined protocol	27
Integrator.REF	Set reference point for time measurement	27
Integrator.RESet	Reset command	27
Integrator.SAVE	Save trace for postprocessing in TRACE32	27
Integrator.SelfArm	Automatic restart of trace recording	27
Integrator.ShowFocus	Display data eye for AUTOFOCUS preprocessor	27
Integrator.SIZE	Define buffer size	27
Integrator.SnapShot	Restart trace capturing once	28
Integrator.SPY	Adaptive stream and analysis	28
Integrator.state	Display trace configuration window	28
Integrator.STATistic	Statistic analysis	28
Integrator.STREAMCompression	Select compression mode for streaming	28
Integrator.STREAMFILE	Specify temporary streaming file path	28
Integrator.STREAMFileLimit	Set size limit for streaming file	28
Integrator.STREAMLOAD	Load streaming file from disk	29
Integrator.STREAMSAVE	Save streaming file to disk	29
Integrator.TCount	Set trigger counter	29
Integrator.TDelay	Trigger delay	29
Integrator.TestFocus	Test trace port recording	29
Integrator.Timing	Waveform of trace buffer	29
Integrator.TRACK	Set tracking record	29
Integrator.TRIGGER	Trigger the trace	30
Integrator.TSElect	Select trigger source	30
Integrator.View	Display single record	30
Integrator.ZERO	Align timestamps of trace and timing analyzers	30
IProbe		31
IProbe	IProbe logic analyzer	31
IProbe-specific Trace Commands		33
IProbe.ALOWerLiMit	Set lower trigger/filter comparator value	33
IProbe.ATrigEN	Enable/disable trigger contribution of a channel	33
IProbe.ATrigMODE	Set trigger/filter condition	34
IProbe.AUPPerLiMit	Set upper trigger/filter comparator value	35
IProbe.Break	Manual IProbe break	35
IProbe.CSElect	Source select for system counter	35
IProbe.EXPORT	Export trace data	36
IProbe.Mode	Set trace operation mode	36

IProbe.SELFTEST	Iprobe self-test	36
IProbe.SIZE	Define the trace buffer size	37
IProbe.state	Display the IProbe configuration window	38
IProbe.TOut	Activates/deactivates the trigger output signal (BUSA)	39
IProbe.TPreDelay	Define the trigger pre-delay counter	39
IProbe.TRIGGER	Ineffective command	39
IProbe.TSElect	Select trigger input line	40
IProbe.TSYNC	Select trigger line and mode	40
IProbe.TSYNC.SELect	Select trigger input pin and edge or state	40
IProbe.TSYNC.SIMPLE	Select simple trigger	40
IProbe.TWidth	Define trigger pulse width	41
Generic IProbe Trace Commands		42
IProbe.Arm	Arm the trace	42
IProbe.AutoArm	Arm automatically	42
IProbe.AutoInit	Automatic initialization	42
IProbe.BookMark	Set a bookmark in trace listing	42
IProbe.Chart	Display trace contents graphically	42
IProbe.ComPare	Compare trace contents	42
IProbe.DISable	Disable the trace	43
IProbe.DisConfig	Trace disassembler configuration	43
IProbe.DRAW	Plot trace data against time	43
IProbe.FILE	Load a file into the file trace buffer	43
IProbe.Find	Find specified entry in trace	43
IProbe.FindAll	Find all specified entries in trace	43
IProbe.FindChange	Search for changes in trace flow	43
IProbe.Get	Display input level	43
IProbe.GOTO	Move cursor to specified trace record	44
IProbe.Init	Initialize trace	44
IProbe.List	List trace contents	44
IProbe.ListNesting	Analyze function nesting	44
IProbe.ListVar	List variable recorded to trace	44
IProbe.LOAD	Load trace file for offline processing	44
IProbe.OFF	Switch off	44
IProbe.PROfile	Rolling live plots of trace data	44
IProbe.PROfile.channel	Display profile of signal probe channels	45
IProbe.PROfileChart	Profile charts	45
IProbe.PROTOcol	Protocol analysis	45
IProbe.PROTOcol.Chart	Graphic display for user-defined protocol	45
IProbe.PROTOcol.Draw	Graphic display for user-defined protocol	45
IProbe.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	45
IProbe.PROTOcol.Find	Find in trace buffer for user-defined protocol	45
IProbe.PROTOcol.list	Display trace buffer for user-defined protocol	46
IProbe.PROTOcol.PROfileChart	Profile chart for user-defined protocol	46

IProbe.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	46
IProbe.PROTOcol.STATistic	Display statistics for user-defined protocol	46
IProbe.REF	Set reference point for time measurement	46
IProbe.RESet	Reset command	46
IProbe.SAVE	Save trace for postprocessing in TRACE32	46
IProbe.SelfArm	Automatic restart of trace recording	47
IProbe.SnapShot	Restart trace capturing once	47
IProbe.SPY	Adaptive stream and analysis	47
IProbe.STATistic	Statistic analysis	47
IProbe.STREAMCompression	Select compression mode for streaming	47
IProbe.STREAMFILE	Specify temporary streaming file path	47
IProbe.STREAMFileLimit	Set size limit for streaming file	47
IProbe.TCount	Set trigger counter	48
IProbe.TDelay	Trigger delay	48
IProbe.Timing	Waveform of trace buffer	48
IProbe.TRACK	Set tracking record	48
IProbe.View	Display single record	48
IProbe.ZERO	Align timestamps of trace and timing analyzers	48
ISTATistic		49
ISTATistic	Instruction statistics	49
Overview ISTATistic		49
ISTATistic.ACCESS	Define access path to program code for ISTAT	52
ISTATistic.ADD	Add trace contents to ISTAT database	53
ISTATistic.Delete	Delete selected code coverage information	57
ISTATistic.EXPORT	Export instruction statistics to a file	58
ISTATistic.EXPORT.CSV	Export instruction statistics in CSV format	59
ISTATistic.EXPORT.ListFunc	Export the HLL functions	59
ISTATistic.EXPORT.ListLine	Export the HLL lines	60
ISTATistic.EXPORT.ListModule	Export the modules	61
ISTATistic.Init	Initialize ISTAT database	61
ISTATistic.List	Run-time analysis overview	62
ISTATistic.ListFunc	List run-time analysis of functions	63
ISTATistic.ListLine	List run-time analysis of HLL lines	64
ISTATistic.ListModule	List module tree of ISTAT database	64
ISTATistic.ListsYmbol	List run-time analysis of symbol regions	65
ISTATistic.LOAD	Load ISTAT database from file	65
ISTATistic.METHOD	Recording method for instruction statistics	66
ISTATistic.OFF	Deactivate the selected instruction statistics method	66
ISTATistic.ON	Activate the selected instruction statistics method	67
ISTATistic.RESet	Delete ISTAT database	67
ISTATistic.SAVE	Save ISTAT database to file	67
ISTATistic.Set	Mark specified addresses as executed	68
ISTATistic.state	Display ISTAT configuration window	69

ITM - Configuration of the Trace Source	70
ITM	CoreSight ITM (Instrumentation Trace Macrocell) 70
ITM.CLEAR	Reset ITM control register 71
ITM.CLOCK	Core clock frequency 71
ITM.CycleAccurate	Cycle accurate tracing 72
ITM.CycleMode	Timestamp source 72
ITM.CyclePrescaler	Set timestamp clock prescaler 73
ITM.DataTrace	Define broadcast of data accesses 74
ITM.DataTraceCorrelateDistance	ETM/ITM data trace correlation 76
ITM.DataTraceCorrelatePusher	Push data cycles 76
ITM.DWTADDRESS	Supply comparator values 76
ITM.InterruptTrace	Emit interrupt event information 77
ITM.OFF	Switch ITM off 78
ITM.ON	Switch ITM on 78
ITM.PCSampler	Emit PC at regular intervals 78
ITM.PortFilter	Filter by channel 79
ITM.PortRoute	Selects the trace port 79
ITM.PortSize	Trace export size 80
ITM.ProfilingTrace	Provide DWT counter information 80
ITM.Register	Display ITM control registers 81
ITM.RESet	Reset ITM settings 81
ITM.STALL	Stall processor to prevent FIFO overflow 81
ITM.state	Display ITM configuration window 82
ITM.SyncPeriod	Set period of sync packet injection 83
ITM.TlmeMode	Type of timestamp 84
ITM.TimeStamp	Emit global timestamp packets 85
ITM.TimeStampCLOCK	External clock frequency 85
ITM.TimeStampMode	Clock source for local timestamp 86
ITM.TraceID	Set trace ID manually 86
ITM.TracePriority	Set priority for the ITM manually 86
ITM<trace> - Trace Data Analysis	87
ITM<trace>	Command groups for ITM<trace> 87
Overview ITM<trace>	87
ITMAnalyzer	Analyze ITM information recorded by TRACE32 PowerTrace 88
ITMCAalyzer	Analyze ITM information recorded by TRACE32 CombiProbe 88
ITMHAlyzer	Analyze ITM information captured by the host analyzer 89
ITMLA	Analyze ITM information from binary source 89
ITMOnchip	Analyze ITM information captured in target onchip memory 90
ITMTrace	Method-independent analysis of ITM trace data 90

General Commands Reference Guide J

General Commands Reference Guide J	(general_ref_j.pdf) 1

History		5
Java		6
Java	Java debugging subsystem	6
Java.CONFIG	Configure VM type for debugging	7
Java.LOAD	Load all Java symbols	8
Java.LOADCLASS	Load Java class information	8
Java.MAP	Java VM specific mappings	9
Java.MAP.ByteCode	Define byte code area	9
Java.MAP.CB	Configure Java VM class block pointer	10
Java.MAP.CP	Configure Java VM class pointer	10
Java.MAP.FP	Configure Java VM frame pointer	10
Java.MAP.IP	Configure Java VM instruction pointer	11
Java.MAP.IPBASE	Configure Java VM IPBASE pointer	12
Java.MAP.List	List Java VM specific mappings	12
Java.MAP.LOADATTR	Load attribute information from Java class files	12
Java.MAP.LP	Configure Java VM LP pointer	13
Java.MAP.MB	Configure Java VM method block pointer	13
Java.MAP.NoByteCode	Remove byte code mapping	14
Java.MAP.NoVM	Remove VM interpreter flag	14
Java.MAP.NoVMStop	Remove breakpoint in VM interpreter	15
Java.MAP.RESet	Reset Java VM mappings	15
Java.MAP.SP	Configure Java VM stack pointer	16
Java.MAP.VM	Configure Java VM interpreter routine area	17
Java.MAP.VMStop	Configure breakpoint in VM interpreter	17
Java.OFF	Disable Java VM debugging subsystem	18
Java.ON	Activate Java debugging subsystem	18
Java.state	Display Java VM subsystem state	19
JTAG		20
JTAG	Low-level JTAG control	20
JTAG.CLIENTINDEX	Select data set for commands	21
JTAG.LOADBIT	Configure a Xilinx FPGA with a BIT file	22
JTAG.LOCK	Grab the JTAG port for manual control	22
JTAG.MIPI34	Manually control MIPI34 connector pins	24
JTAG.PARKSTATE	Define the hand over TAP state	25
JTAG.PIN	Set JTAG signals manually	26
JTAG.PROGRAM	Run programming file	28
JTAG.PROGRAM.Altera	Program Altera FPGAs	29
JTAG.PROGRAM.auto	Detect and run programming file	30
JTAG.PROGRAM.JAM	Run programming file in JAM/STAPL format	31
JTAG.PROGRAM.JBC	Run programming file in binary JAM/STAPL format	32
JTAG.PROGRAM.SVF	Run programming file in SVF format	32
JTAG.PROGRAM.Xilinx	Program Xilinx FPGAs	34
JTAG.RESet	Reset JTAG settings	35

JTAG.SEquence	Special JTAG sequences for certain events	36
JTAG.SEquence.ADD	Add new action to JTAG sequence	37
JTAG.SEquence.Append	Append one sequence to another sequence	46
JTAG.SEquence.Create	Create new JTAG sequence	46
JTAG.SEquence.Delete	Delete JTAG sequence	47
JTAG.SEquence.Execute	Run JTAG sequence	48
JTAG.SEquence.List	Show list of all sequences	49
JTAG.SEquence.MemAccess.ADD	Register sequence for memory access	50
JTAG.SEquence.MemAccess.List	View registered memory accesses	54
JTAG.SEquence.MemAccess.ReMove	Delete registered memory accesses	54
JTAG.SEquence.MemAccess.Replace	Replace registered memory access	54
JTAG.SEquence.ReMove	Remove action from sequence	55
JTAG.SEquence.Replace	Replace action inside sequence	55
JTAG.SEquence.View	Display JTAG sequence	56
JTAG.SHIFTREG	Send a TDI pattern on the JTAG port	57
JTAG.SHIFTTDI	Send a TDI pattern on the JTAG port	58
JTAG.SHIFTTMS	Send a TMS pattern on the JTAG port	59
JTAG.SWD.Init	Initialize the debug port	60
JTAG.SWD.ReadDapBus	Read register from DAP	60
JTAG.SWD.ReadScan	Read register from DAP	60
JTAG.SWD.Select	Configure SWD multi drop target selection	60
JTAG.SWD.SHIFT	Shift data by using the SWIO pin	61
JTAG.SWD.WriteDapBus	Write register to DAP	61
JTAG.SWD.WriteScan	Write register to DAP	62
JTAG.UNLOCK	Hand the JTAG port control back to the debugger	62
JTAG.USECLOCK	Observe shift commands	63
JTAG.X7EFUSE	Program Xilinx 7-Series eFuses	64
JTAG.XUSEFUSE	Program Xilinx UltraScale eFUSES	70
JTAG.CJTAG	Low-level CJTAG control	76
JTAG.CJTAG.COMMAND	Send command to the chip	76
JTAG.CJTAG.START	Access the target via CJTAG	76

General Commands Reference Guide K

General Commands Reference Guide K	(general_ref_k.pdf)	1

General Commands Reference Guide L

General Commands Reference Guide L	(general_ref_l.pdf)	1
History		6
LA		7
LA	Logic analyzer	7
LA-specific Trace Commands		8

LA.IMPORT	Import trace information	8
LA.IMPORT.CoreByteStream	Import pure single core trace data	10
LA.IMPORT.cycles	Import bus trace data	11
LA.IMPORT.ELA	Import ELA trace data	11
LA.IMPORT.ETB	Import on-chip trace data	11
LA.IMPORT.GUESSWRAP	Guess wrap pointer	13
LA.IMPORT.StartInvalid	Set start of trace as invalid	13
LA.IMPORT.StartValid	Set start of trace as valid	14
LA.IMPORT.STP	Import STP recording from file (nibble)	14
LA.IMPORT.STPByteStream	Import STP recording from file (byte)	15
LA.IMPORT.TARMAC	Import TARMAC trace file	15
LA.IMPORT.TraceFile	Import trace data where processing has failed	16
LA.IMPORT.TracePort	Import off-chip trace data	16
LA.IMPORT.UltraSOC	Import raw UltraSOC flow trace data	17
LA.IMPORT.VCD	Import recorded signals in VCD file format	17
LA.IMPORT.WRAP	Define wrap pointer	18
LA.Mode	Set the trace operation mode	19
Generic LA Trace Commands		20
LA.ACCESS	Define access path to program code for trace decoding	20
LA.Arm	Arm the trace	20
LA.AutoArm	Arm automatically	20
LA.AutoInit	Automatic initialization	20
LA.BookMark	Set a bookmark in trace listing	20
LA.Chart	Display trace contents graphically	20
LA.CLOCK	Clock to calculate time out of cycle count information	21
LA.ComPare	Compare trace contents	21
LA.ComPareCODE	Compare trace with memory	21
LA.DISable	Disable the trace	21
LA.DRAW	Plot trace data against time	21
LA.EXPORT	Export trace data for processing in other applications	21
LA.FILE	Load a file into the file trace buffer	21
LA.Find	Find specified entry in trace	21
LA.FindAll	Find all specified entries in trace	22
LA.FindChange	Search for changes in trace flow	22
LA.FLOWPROCESS	Process flowtrace	22
LA.FLOWSTART	Restart flowtrace processing	22
LA.GOTO	Move cursor to specified trace record	22
LA.Init	Initialize trace	22
LA.List	List trace contents	22
LA.ListNesting	Analyze function nesting	23
LA.ListVar	List variable recorded to trace	23
LA.LOAD	Load trace file for offline processing	23
LA.OFF	Switch off	23

LA.PROfileChart	Profile charts	23
LA.PROfileSTATistic	Statistical analysis in a table versus time	23
LA.PROTOcol	Protocol analysis	23
LA.PROTOcol.Chart	Graphic display for user-defined protocol	24
LA.PROTOcol.Draw	Graphic display for user-defined protocol	24
LA.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	24
LA.PROTOcol.Find	Find in trace buffer for user-defined protocol	24
LA.PROTOcol.list	Display trace buffer for user-defined protocol	24
LA.PROTOcol.PROfileChart	Profile chart for user-defined protocol	24
LA.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	24
LA.PROTOcol.STATistic	Display statistics for user-defined protocol	25
LA.REF	Set reference point for time measurement	25
LA.RESet	Reset command	25
LA.SAVE	Save trace for postprocessing in TRACE32	25
LA.SelfArm	Automatic restart of trace recording	25
LA.SIZE	Define buffer size	25
LA.SnapShot	Restart trace capturing once	25
LA.state	Display trace configuration window	25
LA.STATistic	Statistic analysis	26
LA.Timing	Waveform of trace buffer	26
LA.TRACK	Set tracking record	26
LA.View	Display single record	26
LA.ZERO	Align timestamps of trace and timing analyzers	26
List		27
List	Display modes for programs	27
List.auto	Display program listing	28
List.Asm	Display disassembler	36
List.EXPORT	Export a listing to an XML file	37
List.EXPORT.Asm	Export disassembler listing	37
List.EXPORT.auto	Export source and disassembler listing	38
List.EXPORT.Hll	Export source listing	38
List.EXPORT.Mix	Export source and disassembler listing	39
List.Hll	Display source	41
List.Java	Display Java byte code	42
List.Mix	Disassembler and source	44
LOGGER		45
LOGGER	Trace method LOGGER, recording and analysis commands	45
LOGGER-specific Trace Commands		46
LOGGER.ADDRESS	Software trace address	46
LOGGER.Mode	Set LOGGER operation mode	46
LOGGER.TimeStamp	Configure timestamp usage of LOGGER trace	47
Generic LOGGER Trace Commands		48

LOGGER.ACCESS	Define access path to program code for trace decoding	48
LOGGER.Arm	Arm the trace	48
LOGGER.AutoArm	Arm automatically	48
LOGGER.AutoInit	Automatic initialization	48
LOGGER.BookMark	Set a bookmark in trace listing	48
LOGGER.BookMarkToggle	Toggles a single trace bookmark	48
LOGGER.Chart	Display trace contents graphically	49
LOGGER.ComPare	Compare trace contents	49
LOGGER.DISable	Disable the trace	49
LOGGER.DRAW	Plot trace data against time	49
LOGGER.EXPORT	Export trace data for processing in other applications	49
LOGGER.FILE	Load a file into the file trace buffer	49
LOGGER.Find	Find specified entry in trace	49
LOGGER.FindAll	Find all specified entries in trace	49
LOGGER.FindChange	Search for changes in trace flow	50
LOGGER.FLOWPROCESS	Process flowtrace	50
LOGGER.FLOWSTART	Restart flowtrace processing	50
LOGGER.GOTO	Move cursor to specified trace record	50
LOGGER.Init	Initialize trace	50
LOGGER.List	List trace contents	50
LOGGER.ListNesting	Analyze function nesting	50
LOGGER.ListVar	List variable recorded to trace	51
LOGGER.LOAD	Load trace file for offline processing	51
LOGGER.OFF	Switch off	51
LOGGER.PROfileChart	Profile charts	51
LOGGER.PROfileSTATistic	Statistical analysis in a table versus time	51
LOGGER.PROTOcol	Protocol analysis	51
LOGGER.PROTOcol.Chart	Graphic display for user-defined protocol	51
LOGGER.PROTOcol.Draw	Graphic display for user-defined protocol	52
LOGGER.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	52
LOGGER.PROTOcol.Find	Find in trace buffer for user-defined protocol	52
LOGGER.PROTOcol.list	Display trace buffer for user-defined protocol	52
LOGGER.PROTOcol.PROfileChart	Profile chart for user-defined protocol	52
LOGGER.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	52
LOGGER.PROTOcol.STATistic	Display statistics for user-defined protocol	52
LOGGER.REF	Set reference point for time measurement	53
LOGGER.RESet	Reset command	53
LOGGER.SAVE	Save trace for postprocessing in TRACE32	53
LOGGER.SelfArm	Automatic restart of trace recording	53
LOGGER.SIZE	Define buffer size	53
LOGGER.SnapShot	Restart trace capturing once	53
LOGGER.state	Display trace configuration window	53
LOGGER.STATistic	Statistic analysis	53

LOGGER.Timing	Waveform of trace buffer	54
LOGGER.TRACK	Set tracking record	54
LOGGER.View	Display single record	54
LOGGER.ZERO	Align timestamps of trace and timing analyzers	54
LUA		55
LUA	Support for the Lua script language	55
LUA.Data.Loadinput	Load content from a file into the input buffer	55
LUA.Data.Saveoutput	Save output buffer into a binary file	56
LUA.Data.SET	Modify the Lua input buffer	56
LUA.Data.ShowInput	Show current content of the input buffer	57
LUA.Data.ShowOutput	Show current content of the output buffer	57
LUA.Program.List	List the current Lua scripts	57
LUA.Program.LOAD	Load a Lua script to debugger	58
LUA.Program.RESet	Reset the Lua context	58
LUA.Program.RUN	Execute a Lua script	58
LUA.Program.UNLOAD	Remove a Lua script from the debugger	59

General Commands Reference Guide M

General Commands Reference Guide M	(general_ref_m.pdf)	1
History		8
MACHINE		9
MACHINE.select	Display context of specified virtual machine	9
MAP		10
MAP	Mapping memory attributes	10
Overview MAP		10
MAP.ADelay	Set analyzer delay	12
MAP.BE	Define big endian area	12
MAP.BOnchip	Use on-chip breakpoints	13
MAP.BUS<x>	Read/write data in specified access width	14
MAP.BUS8	Bus width mapping	15
MAP.BUS16	Bus width mapping	15
MAP.BUS24	Bus width mapping	15
MAP.BUS32	Bus width mapping	16
MAP.BUS3264	Bus width mapping	16
MAP.BUS64	Bus width mapping	16
MAP.BYTE	Set EPROM width	17
MAP.CacheInhibit	CTS cache simulation	17
MAP.COMSTART	Offset for ROM monitor	17
MAP.CONST	Mapped address range contains constants	18
MAP.DenyAccess	Deny memory access by TRACE32	19
MAP.DenyBurst	Deny burst access to memory by TRACE32	20

MAP.DMUX	Define DRAM area	20
MAP.FRAG	Form fragment	20
MAP.GAP	Define gap	21
MAP.InitVar	CTS initial variable mapping	21
MAP.LE	Define little endian area	22
MAP.List	List allocation	22
MAP.MONITOR	MONITOR address range	23
MAP.NoBE	Switch off big endian	24
MAP.NoBOnchip	Use on-chip breakpoints	24
MAP.NoCacheInhibit	CTS cache simulation	24
MAP.NoCONST	Undo MAP.CONST settings	25
MAP.NoDenyAccess	Switch off deny access for TRACE32	25
MAP.NoDenyBurst	Undo MAP.DENYBURST settings	25
MAP.NoDMUX	Undo MAP.DMUX settings	26
MAP.NOFRAG	Switch off fragmentation	26
MAP.NOGAP	Switch off gap	26
MAP.NoInitVar	CTS initial variable mapping	27
MAP.NoLE	Switch off little endian	27
MAP.NoOPFetch	Switch off opfetch area mapping	27
MAP.NOPAGE	Undefine pages	28
MAP.NOROM	Unmap ESI	28
MAP.NOSWAP	Keep byte order	28
MAP.NoUpdateOnce	Undo MAP.UpdateOnce settings	29
MAP.NoVMREAD	Undo MAP.VMREAD settings	29
MAP.NoVOLATILE	Undo MAP.VOLATILE settings	29
MAP.OPFetch	Opfetch area mapping	30
MAP.PAGE	Define pages	30
MAP.RESet	Reset	31
MAP.ROM	Map ESI	31
MAP.state	State	32
MAP.SWAP	Change byte order	32
MAP.UpdateOnce	Read memory only once each time CPU stops	33
MAP.VMREAD	Redirect memory reads to TRACE32 virtual memory	34
MAP.VOLATILE	Mapped address range is volatile	34
MAP.WORD	Set EPROM width	34
MCDS		35
MCDS	Multicore debug solution	35
MCDS.BusTrace.Agents	Set bus trace agents	37
MCDS.BusTrace.Mode	Set bus trace mode	37
MCDS.CLEAR	Clear programming and initialize MCDS registers	38
MCDS.CLOCK	Configure MCDS clock system	39
MCDS.CLOCK.DEPRECATED	Deprecated MCDS clock programming	39
MCDS.CLOCK.EXTern	Set the external clock frequency	40

MCDS.CLOCK.Frequency	Specify MCDS-related frequencies by commands	41
MCDS.CLOCK.Frequency.McDsClock	Specify the MCDS clock	41
MCDS.CLOCK.Frequency.ReferenceClock	Specify the reference clock	41
MCDS.CLOCK.MCDSDIV	Set divider for generating the MCDS clock	42
MCDS.CLOCK.REFDIV	Set divider for generating the reference clock	42
MCDS.CLOCK.REFerence	Select the reference clock source	43
MCDS.CLOCK.SYStem	Set the system clock frequency	43
MCDS.CLOCK.TIMER	Setup timer for periodic trigger event	44
MCDS.CLOCK.TimeStamp	Force decoding of timestamp messages	45
MCDS.DataTrace.Agents	Set data trace agents	46
MCDS.DataTrace.Mode	Set data trace mode	47
MCDS.INFO	Information on MCDS and usage	47
MCDS.Init	Initialize MCDS registers	48
MCDS.OFF	Disable MCDS programming	48
MCDS.ON	Enable MCDS programming	48
MCDS.Option	Control MCDS feature behavior	49
MCDS.Option.CoreBreak	Break when BREANK_OUT becomes active	49
MCDS.Option.DataAssign	Data assignment in trace listing	49
MCDS.Option.eXception	Exception identification in trace decoder	50
MCDS.Option.FlowControl	Configure AGBT fifo overflow control	51
MCDS.Option.QuickOFF	Disable trace recording by hardware	52
MCDS.Option.RESetBehavior	Configure Onchip behavior after chip reset	52
MCDS.PERipheralTrace	Control peripheral trace	53
MCDS.PortSIZE	Set number of used Aurora lanes	53
MCDS.PortSPEED	Set Aurora lane speed	54
MCDS.ProgramTrace.Agents	Set program trace agents	55
MCDS.ProgramTrace.Mode	Set program trace mode	56
MCDS.Register	Open window with MCDS registers	57
MCDS.RESet	Reset the MCDS unit in the debug tool	57
MCDS.RM	MCDS resource management commands	58
MCDS.RM.ReSTore	Restore MCDS registers	58
MCDS.RM.WriteTarget	Flush MCDS register cache	58
MCDS.SessionKEY	Provide MCDS session key	59
MCDS.Set	Program MCDS on hardware level	59
MCDS.SOURCE	Set MCDS trace sources	61
MCDS.SOURCE.ALL	Enable all MCDS trace sources	61
MCDS.SOURCE.DEFaulT	Set default MCDS trace sources	61
MCDS.SOURCE.NONE	Disable all MCDS trace sources	62
MCDS.SOURCE.Set	Set individual MCDS trace sources	63
MCDS.state	Display MCDS configuration window	69
MCDS.TlmeMode	Configure MCDS timestamp creation and processing	71
MCDS.TraceAgents.CLEAR	Clear all trace agents	72
MCDS.TraceBuffer	Configure MCDS trace buffer	73

MCDS.TraceBuffer.ARRAY	Select MCDS trace buffer array	73
MCDS.TraceBuffer.DETECT	Auto-detect MCDS trace buffer configuration	74
MCDS.TraceBuffer.LowerGAP	Set MCDS trace buffer lower gap	75
MCDS.TraceBuffer.NoStealing	Prevent conflicts with third-party tools	76
MCDS.TraceBuffer.SIZE	Set MCDS trace buffer size	76
MCDS.TraceBuffer.state	Show trace buffer state window	77
MCDS.TraceBuffer.UpperGAP	Set MCDS trace buffer upper gap	77
MCDSBase<trace>	Non-optimized MCDS trace processing	78
MCSDSDCA<trace>	MCDS trace processing with data cycle assignment	78
MCSDSDTU<trace>	MCDS trace processing with DDTU reordering	79
MIPS		80
MIPS	Number of instructions per second	80
Overview MIPS		80
MIPS.List	List the MIPS trace data	83
MIPS.ListNesting	Show program nesting	85
MIPS.PROfileChart	Profile charts for MIPS	86
MIPS.PROfileChart.AddressGROUP	MIPS profile chart for address groups	86
MIPS.PROfileChart.ALL	MIPS profile chart for program run	87
MIPS.PROfileChart.DatasYmbol	MIPS profile chart for pointer	87
MIPS.PROfileChart.DistriB	MIPS profile chart for distributions	88
MIPS.PROfileChart.GROUP	MIPS profile chart for groups	89
MIPS.PROfileChart.Line	MIPS per high-level language line graphically	90
MIPS.PROfileChart.MODULE	MIPS profile chart for modules	91
MIPS.PROfileChart.PROGRAM	MIPS profile chart for programs	92
MIPS.PROfileChart.RWINST	MIPS per cycle type graphically	92
MIPS.PROfileChart.sYmbol	MIPS for all program symbols graphically	93
MIPS.PROfileChart.TASK	MIPS per task graphically	94
MIPS.PROfileChart.TASKINFO	MIPS for data trace via context ID	94
MIPS.PROfileChart.TASKINTR	MIPS profile chart for ISR2 (ORTI)	95
MIPS.PROfileChart.TASKKernel	MIPS profile chart with kernel marker	95
MIPS.PROfileChart.TASKORINTERRUPT	MIPS graph per task/interrupt	96
MIPS.PROfileChart.TASKSRV	MIPS profile chart for OS service routines	96
MIPS.PROfileChart.TASKVSINTR	MIPS chart for task-related interrupts	97
MIPS.PROfileSTATistic	Profile statistics for MIPS	98
MIPS.PROfileSTATistic.Address	MIPS per address as profile statistic	98
MIPS.PROfileSTATistic.AddressGROUP	MIPS per address group	99
MIPS.PROfileSTATistic.ALL	MIPS profile statistic for program run	99
MIPS.PROfileSTATistic.DatasYmbol	MIPS profile statistic for pointer	100
MIPS.PROfileSTATistic.DistriB	Distribution statistical analysis	100
MIPS.PROfileSTATistic.GROUP	MIPS per GROUP as profile statistic	101
MIPS.PROfileSTATistic.INTERRUPT	MIPS per interrupt as table	101
MIPS.PROfileSTATistic.Line	MIPS per high-level language line as table	102
MIPS.PROfileSTATistic.MODULE	MIPS per module as profile statistic	102

MIPS.PROfileSTATistic.PROGRAM	MIPS per program as profile statistic	103
MIPS.PROfileSTATistic.RUNNABLE	MIPS per runnable as table	103
MIPS.PROfileSTATistic.RWINST	MIPS per cycle type as table	104
MIPS.PROfileSTATistic.sYmbol	MIPS for all program symbols as table	104
MIPS.PROfileSTATistic.TASK	MIPS per task as table	105
MIPS.PROfileSTATistic.TASKINFO	MIPS for data trace via context ID	105
MIPS.PROfileSTATistic.TASKINTR	MIPS per ISR2 (ORTI) as table	106
MIPS.PROfileSTATistic.TASKKernel	MIPS per task as table	106
MIPS.PROfileSTATistic.TASKORINTERRUPT	MIPS per task as table	107
MIPS.PROfileSTATistic.TASKSRV	MIPS per OS service routine as table	107
MIPS.STATistic	Statistical analysis for MIPS	108
MIPS.STATistic.ALL	MIPS for the program run	108
MIPS.STATistic.ChildTREE	MIPS for the callee context of a function	108
MIPS.STATistic.DistriB	MIPS distribution analysis	109
MIPS.STATistic.Func	MIPS for functions numerically	109
MIPS.STATistic.GROUP	MIPS statistic for groups	109
MIPS.STATistic.LINKage	Per caller MIPS statistic of function	110
MIPS.STATistic.MODULE	MIPS for modules numerically	110
MIPS.STATistic.ParentTREE	MIPS statistic for call context of a function	111
MIPS.STATistic.PROGRAM	MIPS for programs numerically	111
MIPS.STATistic.RWINST	MIPS per cycle type numerically	112
MIPS.STATistic.sYmbol	MIPS for all program symbols numerically	112
MIPS.STATistic.TASK	MIPS per task numerically	112
MIPS.STATistic.TASKINFO	MIPS for data trace via context ID	113
MIPS.STATistic.TASKINTR	MIPS per ISR2 numerically	113
MIPS.STATistic.TASKKernel	MIPS task analysis with kernel markers	113
MIPS.STATistic.TASKSRV	MIPS per OS service routine numerically	114
MIPS.STATistic.TREE	Tree display of nesting functions with MIPS	114
MMU		115
MMU	Memory management unit	115
Overview MMU		115
MMU.DUMP	Dump MMU tables	116
MMU.FORMAT	Define MMU table structure	118
MMU.INFO	Translation information related to an address	121
MMU.INFO.TaskPageTable	Translation information related to an address	122
MMU.List	Compact display of MMU translation table	123
MMU.MemAnalysis	Analyze page tables	124
MMU.SCAN	Scan MMU tables (static snapshot)	128
MMU.Set	Set MMU registers or tables	129
MMU.TDUMP	Dump task page table	130
MMU.TSCAN	Scan task page table	130
MMU.view	View MMU registers	130
MMX		131

MMX	MMX registers (MultiMedia eXtension)	131
MMX.Init	Initialize MMX registers	131
MMX.Set	Modify MMX registers	131
MMX.view	Open MMX register window	132
Mode		133
Mode	Set up the debug mode	133
Appendix - <format> Options of MMU.FORMAT		134

General Commands Reference Guide N

General Commands Reference Guide N	(general_ref_n.pdf)	1
NAME		4
NAME	Logical names for physical connections	4
NAME.Combi	Create virtual signal for trace events	4
NAME.Delete	Delete names	6
NAME.Group	Groups signals	7
NAME.list	Display name definitions	8
NAME.RESet	Delete all names	8
NAME.SELect	Define names	9
NAME.Set	Define names	10
NAME.User	Create new user channel	12
NAME.Word	Group signals as word	14
NEXUS		15
NEXUS	NEXUS trace	15

General Commands Reference Guide O

General Commands Reference Guide O	(general_ref_o.pdf)	1
History		5
OCP		6
OCP	OpenCoreProtocol WatchPoint	6
ON		7
ON	Event-controlled PRACTICE script execution	7
Onchip		12
Onchip	Trace method Onchip, recording, and analysis commands	12
Onchip-specific Trace Commands		13
Onchip.ATTACH	Attach to the onchip trace	13
Onchip.Mode	Set the trace operation mode	13
Onchip Trace Commands		17
Onchip.ACCESS	Define access path to program code for trace decoding	17
Onchip.Arm	Arm the trace	17

Onchip.AutoArm	Arm automatically	17
Onchip.AutoInit	Automatic initialization	17
Onchip.BookMark	Set a bookmark in trace listing	17
Onchip.Chart	Display trace contents graphically	17
Onchip.CLOCK	Clock to calculate time out of cycle count information	18
Onchip.ComPare	Compare trace contents	18
Onchip.ComPareCODE	Compare trace with memory	18
Onchip.DISable	Disable the trace	18
Onchip.DisConfig	Trace disassembler configuration	18
Onchip.DRAW	Plot trace data against time	18
Onchip.EXPORT	Export trace data for processing in other applications	18
Onchip.FILE	Load a file into the file trace buffer	19
Onchip.Find	Find specified entry in trace	19
Onchip.FindAll	Find all specified entries in trace	19
Onchip.FindChange	Search for changes in trace flow	19
Onchip.FindProgram	Advanced trace search	19
Onchip.FindReProgram	Activate advanced existing trace search program	19
Onchip.FindViewProgram	State of advanced trace search programming	19
Onchip.FLOWPROCESS	Process flowtrace	20
Onchip.FLOWSTART	Restart flowtrace processing	20
Onchip.GOTO	Move cursor to specified trace record	20
Onchip.Init	Initialize trace	20
Onchip.JOINFILE	Concatenate several trace recordings	20
Onchip.List	List trace contents	20
Onchip.ListNesting	Analyze function nesting	20
Onchip.ListVar	List variable recorded to trace	21
Onchip.LOAD	Load trace file for offline processing	21
Onchip.MERGEFILE	Combine two trace files into one	21
Onchip.OFF	Switch off	21
Onchip.PlatformCLOCK	Set clock for platform traces	21
Onchip.PROfileChart	Profile charts	21
Onchip.PROfileSTATistic	Statistical analysis in a table versus time	21
Onchip.PROTOcol	Protocol analysis	22
Onchip.PROTOcol.Chart	Graphic display for user-defined protocol	22
Onchip.PROTOcol.Draw	Graphic display for user-defined protocol	22
Onchip.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	22
Onchip.PROTOcol.Find	Find in trace buffer for user-defined protocol	22
Onchip.PROTOcol.list	Display trace buffer for user-defined protocol	22
Onchip.PROTOcol.PROfileChart	Profile chart for user-defined protocol	22
Onchip.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	23
Onchip.PROTOcol.STATistic	Display statistics for user-defined protocol	23
Onchip.REF	Set reference point for time measurement	23
Onchip.RESet	Reset command	23

Onchip.SAVE	Save trace for postprocessing in TRACE32	23
Onchip.SelfArm	Automatic restart of trace recording	23
Onchip.SIZE	Define buffer size	23
Onchip.SnapShot	Restart trace capturing once	24
Onchip.state	Display trace configuration window	24
Onchip.STATistic	Statistic analysis	24
Onchip.STREAMCompression	Select compression mode for streaming	24
Onchip.STREAMFILE	Specify temporary streaming file path	24
Onchip.STREAMFileLimit	Set size limit for streaming file	24
Onchip.STREAMLOAD	Load streaming file from disk	24
Onchip.STREAMSAVE	Save streaming file to disk	25
Onchip.TDelay	Trigger delay	25
Onchip.TestFocus	Test trace port recording	25
Onchip.TestUtilization	Tests trace port utilization	25
Onchip.Timing	Waveform of trace buffer	25
Onchip.TraceCONNECT	Select on-chip peripheral sink	25
Onchip.TRACK	Set tracking record	25
Onchip.TRIGGER	Trigger the trace	26
Onchip.View	Display single record	26
Onchip.ZERO	Align timestamps of trace and timing analyzers	26
Onchip2		27
Onchip2	Second onchip trace buffer	27

General Commands Reference Guide P

General Commands Reference Guide P	(general_ref_p.pdf)	1
History		6
PCI		7
PCI	Legacy PCI configuration	7
PCI.Dump	Display PCI device data	8
PCI.Option.DOMAIN	Set PCI domain	8
PCI.Read	Read a PCI register	9
PCI.Scan	List PCI devices	10
PCI.Write	Write a PCI register	11
PCPOnchip		12
PER		13
PER	Peripheral files	13
Overview PER		13
PER.IMPORT	Import of alternative peripheral file formats	14
PER.<format>.ReProgram	Set default peripheral file	15
PER.<format>.Save	Save to file	15
PER.<format>.TestProgram	Test mode	15

PER.<format>.view	Display peripherals	15
PER.IMPORT.AccessClass	TRACE32 access class	16
PER.IMPORT.EnumDelimiter	Delimiter for BITFLD items.	16
PER.IMPORT.FieldsFromDescription	Generate BITFLDs from description	16
PER.IMPORT.ForMaT	Input file format	17
PER.IMPORT.INDent	Indent trees, registers and fields	17
PER.IMPORT.InputFile	Input files for conversion	18
PER.IMPORT.Load	Load external converter project	18
PER.IMPORT.LOGfile	Create logfile of conversion	18
PER.IMPORT.MaximumChoiceLength	Maximum choice item length	19
PER.IMPORT.MaximumDescriptionLength	Maximum tooltip length	19
PER.IMPORT.MergeGroups	Minimize number of GROUPs	19
PER.IMPORT.ModuleFiles	Split .per file into separate files	20
PER.IMPORT.ModulePath	Output directory for module files	20
PER.IMPORT.MSBfirst	Order of bits in BITFLD command	20
PER.IMPORT.NumberOfColumns	Number of output columns	21
PER.IMPORT.OutputFile	Name of generated peripheral file	21
PER.IMPORT.REPeat	Generate REPEAT commands	22
PER.IMPORT.RESet	Reset import settings	22
PER.IMPORT.RULES	Apply rules file	22
PER.IMPORT.SortSubTrees	Sort TREEs alphabetically	23
PER.IMPORT.SortTopTrees	Sort TREEs alphabetically	23
PER.IMPORT.STOre	Store current project	23
PER.IMPORT.WithValue	Precede bitfield items with value	24
PER.In	Read port	25
PER.Program	Interactive programming	25
PER.ReProgram	Set default peripheral file	26
PER.ReProgramDECRYPT	Load default program (encrypted)	27
PER.Set	Modify memory	28
PER.Set.ByName	Modify memory by name	28
PER.Set.CONDItions	Workaround for PER functions	30
PER.Set.Field	Modify a bit field in memory	30
PER.Set.Index	Modify indirect (indexed) register	32
PER.Set.IndexField	Set fields at indexed register	33
PER.Set.Out	Write data stream to memory	33
PER.Set.SaveIndex	Modify indirect (indexed) register	34
PER.Set.SaveIndexField	Set fields at indexed register	35
PER.Set.SaveTIndex	Set fields at indexed registers	35
PER.Set.SaveTIndexField	Set fields at indexed registers	35
PER.Set.SEQuence	Set SGROUP members	36
PER.Set.SEQuenceField	Set SGROUP members	36
PER.Set.SHADOW	Modify data based on shadow RAM	36
PER.Set.simple	Modify registers/peripherals	37

PER.Set.TIndex	Set fields at indexed registers	37
PER.Set.TIndexField	Set fields at indexed registers	38
PER.STOre	Generate PRACTICE script from PER settings	39
PER.TestProgram	Test mode	41
PER.view	Display peripherals	41
PER.viewDECRYPT	View decrypted PER file in a PER window	44
Programming Commands		45
PERF		46
PERF	Sample-based profiling	46
Overview PERF		46
PERF.ADDRESS	Restrict evaluation to specified address area	53
PERF.Arm	Activate the performance analyzer manually	54
PERF.AutoArm	Couple performance analyzer to program execution	54
PERF.AutoInit	Automatic initialization	54
PERF.ContextID	Enable sampling the context ID register	55
PERF.DISable	Disable the performance analyzer	55
PERF.Init	Reset current measurement	55
PERF.List	Default profiling	56
PERF.ListDistriB	Memory contents profiling	62
PERF.ListFunc	Function profiling	63
PERF.ListFuncMod	HLL function profiling (restricted)	65
PERF.ListLABEL	Label-based profiling	67
PERF.ListLine	Profiling by HLL lines	69
PERF.ListModule	Profiling by modules	70
PERF.ListProgram	Profiling based on performance analyzer program	71
PERF.ListRange	Profiling by ranges	71
PERF.ListS10	Profiling in n-byte segments	72
PERF.ListTASK	Profiling by tasks/threads	73
PERF.ListTREE	Profiling by module/function tree	75
PERF.ListVarState	Variable state profiling	76
PERF.LOAD	Load previously stored PERF results	77
PERF.METHOD	Specify acquisition method	77
PERF.MMUSPACES	Include space IDs for addresses in the sampling	88
PERF.Mode	Specify sampling object	88
PERF.OFF	Stop the performance analyzer manually	90
PERF.PROfile	Graphic profiling display	90
PERF.Program	Write a performance analyzer program	92
PERF.ReProgram	Load an existing performance analyzer program	93
PERF.RESet	Reset analyzer	94
PERF.RunTime	Retain time for program run	94
PERF.SAVE	Save the PERF results for postprocessing	95
PERF.SnoopAddress	Address for memory sample	95
PERF.SnoopMASK	Mask for memory sample	95

PERF.SnoopSize	Size for memory sample	96
PERF.Sort	Specify sorting of evaluation results	96
PERF.state	Display state	97
PERF.STREAM	PERF stream mode	98
PERF.ToProgram	Automatic generation of performance analyzer program	98
PERF.View	Detailed view	99
PERSVD		102
PERSVD	Built-in converter for peripheral files in CMSIS-SVD format	102
PERSVD.Save	Save converted file	102
PERSVD.view	Display peripherals	102
PMI		104
PMI	Power management interface	104
POD		105
POD	Configure input behavior of digital and analog probe	105
POD.ADC	Probe configuration	105
POD.Level	Input state	108
POD.RESet	Input level reset	109
POD.state	Input state	109
POD.USB	Set up USB probe	111
PORT		112
PORT.Arm	Arm the trace	112
PORT.AutoArm	Arm automatically	112
PORT.BookMark	Set a bookmark in trace listing	112
PORT.Chart	Display trace contents graphically	112
PORT.DRAW	Plot trace data against time	112
PORT.FindAll	Find all specified entries in trace	112
PORT.FindChange	Search for changes in trace flow	113
PORT.GOTO	Move cursor to specified trace record	113
PORT.Init	Initialize trace	113
PORT.OFF	Switch off	113
PORT.PROfileChart	Profile charts	113
PORT.PROTOcol	Protocol analysis	113
PORT.PROTOcol.Chart	Graphic display for user-defined protocol	113
PORT.PROTOcol.Draw	Graphic display for user-defined protocol	114
PORT.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	114
PORT.PROTOcol.Find	Find in trace buffer for user-defined protocol	114
PORT.PROTOcol.list	Display trace buffer for user-defined protocol	114
PORT.PROTOcol.PROfileChart	Profile chart for user-defined protocol	114
PORT.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	114
PORT.PROTOcol.STATistic	Display statistics for user-defined protocol	114
PORT.REF	Set reference point for time measurement	115
PORT.RESet	Reset command	115

PORT.SAVE	Save trace for postprocessing in TRACE32	115
PORT.SelfArm	Automatic restart of trace recording	115
PORT.SnapShot	Restart trace capturing once	115
PORT.STATistic	Statistic analysis	115
PORT.Timing	Waveform of trace buffer	115
PORT.TRACK	Set tracking record	115
PORT.ZERO	Align timestamps of trace and timing analyzers	116
Probe		117
Probe	Probe logic analyzer	117

General Commands Reference Guide Q

General Commands Reference Guide Q	(general_ref_q.pdf)	1

General Commands Reference Guide R

General Commands Reference Guide R	(general_ref_r.pdf)	1
History		5
Register		6
Register	Processor registers	6
Register.Init	Initialize the processor registers	6
Register.LOG	Log registers	11
Register.REFRESH	Refresh register window	12
Register.RELOAD	Reload the compiler register settings	12
Register.Set	Modify register contents	13
Register.StackTop	Define stack top address	14
Register.view	Display registers	15
RESet		18
RESet	Reset all commands	18
RTP		19
RTP.CLEAR	Clear tracebuffer	19
RTP.DirectDataMode	Simple trace mode	19
RTP.DirectDataMode.Mode	Direct data mode read/write	19
RTP.HaltOnOverflow	Halt system on RTP FIFO overflow	19
RTP.Mode	Select the trace mode	19
RTP.OFF	Disables the RTP module	19
RTP.ON	Activates the RTP module	19
RTP.PortClock	Configure RTPCLK	20
RTP.PortSize	Size of RTP data port	20
RTP.RESet	Resets RTP settings	20
RTP.state	Display RTP setup	20
RTP.TraceMode	Complex trace mode	20

RTP.TraceMode.RAM<x>.SECTIon<y>	Configures a trace region	20
RTP.TraceMode.TraceExclude	Invert all trace regions	20
RTS		21
RTS	Real-time profiling (RTS)	21
Overview RTS		21
RTS.COMMAND	Issue command to RTS API model	24
RTS.Init	Initialize RTS	24
RTS.LOAD	Load RTS API module	24
RTS.OFF	Deactivate real-time profiling	24
RTS.ON	Activate real-time profiling	25
RTS.PROfile	Display performance characteristics charts	25
RTS.RESet	Restore default settings and initialize RTS	27
RTS.state	Open status and control window	27
RTS.StopOnBadaddress	Stop RTS on VM errors	28
RTS.StopOnError	Stop RTS on flow errors	29
RTS.StopOnFifofull	Stop RTS on FIFOFULL	29
RTS.StopOnNoaccesstocode	Stop RTS on no access to code	30
RTS.StopOnUnknowntask	Stop RTS on unknown task	30
RTS.TlmeMode	Enable RTS processing with time information	31
RTS.TrackData	Enable RTS data tracking	31
RTS.TRIGGERACK	Acknowledge RTS trigger	32
RTS.TriggerConnect	Propagate RTS triggers to RTS trigger slaves	32
RTS.TriggerOnExecute	Generate RTS trigger on execution	32
RTS.TriggerOnRead	Generate RTS trigger on read event	33
RTS.TriggerOnWrite	Generate RTS trigger on write event	33
RTS.TriggerOnWTM	Generate RTS trigger on watchpoint event	33
RTS.TriggerSlave	Receive RTS triggers	34
RTS.TriggerWaitForAck	Stall RTS processing until trigger acknowledged	34
RTS.UnknownData	HTM unknown data	34
RTS.UNLOAD	Unload RTS API module	35
RunTime		36
RunTime	Runtime measurement	36
Overview RunTime		36
RunTime-specific Trace Commands		39
RunTime.List	List runtime logs	39
RunTime.Mode	Mode selection	40
RunTime.refA	Set reference	41
RunTime.refB	Set reference	41
RunTime.SHOW	Display results	42
RunTime.state	Display RunTime configuration and results	44
RunTime.WAIT	Wait until a condition is true or a period has elapsed	45
Generic RunTime Trace Commands		46

RunTime.Arm	Arm the trace	46
RunTime.AutoArm	Arm automatically	46
RunTime.AutoInit	Automatic initialization	46
RunTime.BookMark	Set a bookmark in trace listing	46
RunTime.Chart	Display trace contents graphically	46
RunTime.CLOCK	Clock to calculate time out of cycle count information	46
RunTime.ComPare	Compare trace contents	47
RunTime.DISable	Disable the trace	47
RunTime.EXPORT	Export trace data for processing in other applications	47
RunTime.FILE	Load a file into the file trace buffer	47
RunTime.Find	Find specified entry in trace	47
RunTime.FindAll	Find all specified entries in trace	47
RunTime.FindChange	Search for changes in trace flow	47
RunTime.GOTO	Move cursor to specified trace record	47
RunTime.Init	Initialize trace	48
RunTime.LOAD	Load trace file for offline processing	48
RunTime.OFF	Switch off	48
RunTime.PROfileChart	Profile charts	48
RunTime.REF	Set reference point for time measurement	48
RunTime.RESet	Reset command	48
RunTime.SAVE	Save trace for postprocessing in TRACE32	48
RunTime.SIZE	Define buffer size	48
RunTime.STATistic	Statistic analysis	49
RunTime.Timing	Waveform of trace buffer	49
RunTime.TRACK	Set tracking record	49
RunTime.View	Display single record	49
RunTime.ZERO	Align timestamps of trace and timing analyzers	49

General Commands Reference Guide S

General Commands Reference Guide S	(general_ref_s.pdf)	1
History		14
SELFTEST		15
SELFTEST	Execute selftest operation	15
SETUP		16
SETUP	Setup commands	16
SETUP.ALIST	Default analyzer display	17
SETUP.ALIST.RESet	Reset analyzer display	17
SETUP.ALIST.set	Default analyzer display	17
SETUP.BreakPointTableWalk	Set up MMU translation for breakpoints	17
SETUP.BreakTransfer	Breakpoint synchronization	18
SETUP.COLORCORE	Enable coloring for core-specific info in SMP systems	18
SETUP.DIS	Disassembler configuration	19

SETUP.DUMP	Defaults for hex-dumps	20
SETUP.EMUPATH	Emulation softkeys configuration	21
SETUP.GoOnPaused	Route go to paused core	21
SETUP.IMASKASM	Mask interrupts during assembler step	22
SETUP.IMASKHLL	Mask interrupts during HLL step	22
SETUP.LISTCLICK	Double-click source line symbol to run this action	23
SETUP.PROCESS	Processing percentage in statistics window	24
SETUP.SIMULINK	Deprecated command	24
SETUP.StepAllCores	Force single stepping on all cores	25
SETUP.StepAtBreakPoint	Single step to skip breakpoint	26
SETUP.StepAutoAsm	HLL steps stops at assembler code	26
SETUP.StepBeforeGo	Single step before go	27
SETUP.StepByStep	Single step HLL lines	27
SETUP.StepNoBreak	Stepping HLL lines with disabled breakpoints	27
SETUP.StepOnPaused	Route step to selected core	28
SETUP.StepTrace	Show stepping trail in list window	28
SETUP.StepWithinBreakpoints	Multi-core step on SMP systems	28
SETUP.StepWithinTask	Task selective stepping	29
SETUP.sYmbol	Length of symbols	29
SETUP.TIMEOUT	Define emulation monitor time-out	30
SETUP.Var	Defaults for the Var commands	31
SETUP.VarCall	Define call dummy routine	35
SETUP.VarPtr	Limit pointer access	36
SETUP.VerifyBreakSet	Additional verification for software breakpoints	36
SIM		37
SIM	TRACE32 Instruction Set Simulators	37
SIM.AREA	Selects area for simulation output	37
SIM.CACHE	Cache/MMU simulation and more	38
SIM.CACHE.Allocation	Define the cache allocation technique	39
SIM.CACHE.Mode	Define memory coherency strategy	40
SIM.CACHE.MPURegions	Specify MPU regions	40
SIM.CACHE.OFF	Disable cache and MMU simulation	41
SIM.CACHE.ON	Enable cache and MMU simulation	41
SIM.CACHE.Replacement	Define the replacement strategy	41
SIM.CACHE.SETS	Define the number of cache/TLB sets	43
SIM.CACHE.state	Display cache and MMU settings	44
SIM.CACHE.Tags	Define address mode for cache lines	45
SIM.CACHE.TRACE	Select simulator trace method	45
SIM.CACHE.View	Analysis of memory accesses for cache simulation	46
SIM.CACHE.ViewTLB	Analysis of TLB accesses for MMU simulation	46
SIM.CACHE.WAYS	Define number of cache ways	47
SIM.CACHE.Width	Define width of cache line	47
SIM.command	Issue command to simulation model	48

SIM.INTerrupt	Trigger interrupt	48
SIM.List	List loaded simulator models	49
SIM.LOAD	Load simulator module	49
SIM.RESet	Reset TRACE32 Instruction Set Simulator	49
SIM.UNLOAD	Unload simulator module	50
SLTrace		51
SLTrace	Trace sink for SYStem.LOG events	51
SLTrace.state	Display configuration window	52
SNOOPer		53
SNOOPer	Sample-based trace	53
SNOOPer-specific Trace Commands		54
SNOOPer.<specific_cmds>	Overview of SNOOPer-specific commands	54
SNOOPer.CORE	Select cores for PC snooping	54
SNOOPer.ERRORSTOP	Set behavior on sampling errors	55
SNOOPer.Mode	Set operation mode of SNOOPer trace	56
SNOOPer.PC	Enable PC snooping	60
SNOOPer.Rate	Select sampling rate	61
SNOOPer.SELect	Define address for monitoring	61
SNOOPer.SIZE	Define trace buffer size	63
SNOOPer.TDelay	Define trigger delay	63
SNOOPer.TOut	Define the trigger destination	64
SNOOPer.TValue	Define data value for trigger	65
Generic SNOOPer Trace Commands		66
SNOOPer.ACCESS	Define access path to program code for trace decoding	66
SNOOPer.Arm	Arm the trace	66
SNOOPer.AutoArm	Arm automatically	66
SNOOPer.AutoInit	Automatic initialization	66
SNOOPer.BookMark	Set a bookmark in trace listing	66
SNOOPer.BookMarkToggle	Toggles a single trace bookmark	66
SNOOPer.Chart	Display trace contents graphically	67
SNOOPer.Chart.DistriB	Distribution display graphically	67
SNOOPer.Chart.sYmbol	Symbol analysis	67
SNOOPer.Chart.VarState	Variable activity chart	67
SNOOPer.ComPare	Compare trace contents	67
SNOOPer.DISable	Disable the trace	67
SNOOPer.DRAW	Plot trace data against time	67
SNOOPer.DRAW.channel	Plot no-data values against time	68
SNOOPer.DRAW.Var	Plot variable values against time	68
SNOOPer.EXPORT	Export trace data for processing in other applications	68
SNOOPer.FILE	Load a file into the file trace buffer	68
SNOOPer.Find	Find specified entry in trace	68
SNOOPer.FindAll	Find all specified entries in trace	68

SNOOPPer.FindChange	Search for changes in trace flow	68
SNOOPPer.Get	Display input level	69
SNOOPPer.GOTO	Move cursor to specified trace record	69
SNOOPPer.Init	Initialize trace	69
SNOOPPer.List	List trace contents	69
SNOOPPer.ListVar	List variable recorded to trace	69
SNOOPPer.LOAD	Load trace file for offline processing	69
SNOOPPer.OFF	Switch off	69
SNOOPPer.PROfileChart	Profile charts	69
SNOOPPer.PROfileChart.COUNTER	Display a profile chart	70
SNOOPPer.PROfileSTATistic	Statistical analysis in a table versus time	70
SNOOPPer.PROTOcol	Protocol analysis	70
SNOOPPer.PROTOcol.Chart	Graphic display for user-defined protocol	70
SNOOPPer.PROTOcol.Draw	Graphic display for user-defined protocol	70
SNOOPPer.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	70
SNOOPPer.PROTOcol.Find	Find in trace buffer for user-defined protocol	70
SNOOPPer.PROTOcol.list	Display trace buffer for user-defined protocol	71
SNOOPPer.PROTOcol.PROfileChart	Profile chart for user-defined protocol	71
SNOOPPer.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	71
SNOOPPer.PROTOcol.STATistic	Display statistics for user-defined protocol	71
SNOOPPer.REF	Set reference point for time measurement	71
SNOOPPer.RESet	Reset command	71
SNOOPPer.SAVE	Save trace for postprocessing in TRACE32	71
SNOOPPer.SelfArm	Automatic restart of trace recording	72
SNOOPPer.SnapShot	Restart trace capturing once	72
SNOOPPer.state	Display trace configuration window	72
SNOOPPer.STATistic	Statistic analysis	72
SNOOPPer.STATistic.DistriB	Distribution analysis	72
SNOOPPer.Timing	Waveform of trace buffer	72
SNOOPPer.TRACK	Set tracking record	72
SNOOPPer.View	Display single record	73
SNOOPPer.ZERO	Align timestamps of trace and timing analyzers	73
SPE		74
SPE	Signal Processing eXtension (SPE)	74
SPE.Init	Initialize SPE registers	74
SPE.Set	Modify SPE registers	74
SPE.view	Display SPE register window	75
SSE		76
SSE	SSE registers (Streaming SIMD Extension)	76
SSE.Init	Initialize SSE registers	76
SSE.Set	Modify SSE registers	76
SSE.view	Display SSE registers	77

StatCol		78
StatCol	Statistics collector	78
Step		79
Step	Single-step	79
Step.Asm	Assembler single-step	79
Step.Back	Step backwards	79
Step.BackChange	Step back until expression changes	80
Step.BackOver	Step back over call	80
Step.BackTill	Step back until expression true	80
Step.Change	Step until expression changes	81
Step.Diverge	Step to next unreached line	82
Step.Hll	Step in HLL-mode	84
Step.Mix	Step in mixed-mode	84
Step.Over	Step over call	85
Step.single	Single-step	85
Step.Till	Step until expression true	86
STM		87
STM	System trace configuration	87
STOre		88
STOre	Store settings as PRACTICE script	88
SVE		92
SVE	Access the scalable vector extension SVE	92
SVE.Init	Initialize SVE registers	92
SVE.RESet	Reset SVE settings	92
SVE.Set	Modify SVE registers	92
SVE.view	Display SVE registers	93
sYmbol		94
sYmbol	Debug symbols	94
Overview sYmbol		94
sYmbol.AddInfo	Provide additional symbolic information	96
sYmbol.AddInfo.Address	Add symbol information to fixed address	98
sYmbol.AddInfo.Delete	Delete information	99
sYmbol.AddInfo.LINK	Define information for 'sYmbol.AddInfo' commands	100
sYmbol.AddInfo.List	List additional information	101
sYmbol.AddInfo.LOADASAP2	Load scaling information from ASAP2 file	101
sYmbol.AddInfo.Member	Add information to member of struct	102
sYmbol.AddInfo.RESet	Remove all additional information	104
sYmbol.AddInfo.Type	Add information to a data type	104
sYmbol.AddInfo.Var	Add information to a variable	105
sYmbol.AutoLOAD	Automated loading of symbols	106
sYmbol.AutoLOAD.CHECK	Update autoloader table	107
sYmbol.AutoLOAD.CHECKCoMmanD	Configure dynamic autoloader	108

sYmbol.AutoLOAD.CHECKDLL	Configure automatic DLL file loader	109
sYmbol.AutoLOAD.CHECKEPOC	Dynamic autoloader for Symbian	110
sYmbol.AutoLOAD.CHECKLINUX	Configure autoloader for Linux debugging	110
sYmbol.AutoLOAD.CHECKQNX	Configure autoloader for QNX debugging	111
sYmbol.AutoLOAD.CHECKUEFI	Configure autoloader for UEFI debugging	111
sYmbol.AutoLOAD.CHECKWIN	Configure autoloader	112
sYmbol.AutoLOAD.CHECKWINCE	Configure autoloader	112
sYmbol.AutoLOAD.CLEAR	Remove symbol information	113
sYmbol.AutoLOAD.config	Configure symbol autoloader	113
sYmbol.AutoLOAD.Create	Create entry for autoloader table	114
sYmbol.AutoLOAD.Delete	Delete autoloader entries	114
sYmbol.AutoLOAD.List	List autoloader table	115
sYmbol.AutoLOAD.LOADEPOC	Definition for static autoloader for Symbian	116
sYmbol.AutoLOAD.RESet	Reset autoloader	117
sYmbol.AutoLOAD.SET	Mark symbol information manually as loaded	117
sYmbol.AutoLOAD.TOUCH	Initiate automatic loading by command	118
sYmbol.Browse	Browse symbols	119
sYmbol.Browse.Class	Browse classes	119
sYmbol.Browse.Enum	Browse enumeration types	119
sYmbol.Browse.Function	Browse functions	120
sYmbol.Browse.Module	Browse modules	121
sYmbol.Browse.MVar	Browse module variables	122
sYmbol.Browse.name	Browse symbols (flat)	122
sYmbol.Browse.SFunction	Browse functions	123
sYmbol.Browse.SModule	Browse modules	125
sYmbol.Browse.SOURCE	Browse source	126
sYmbol.Browse.Struct	Browse containers for different variable types	127
sYmbol.Browse.sYmbol	Browse symbols	128
sYmbol.Browse.Type	Browse HLL types	129
sYmbol.Browse.TypeDef	Browse type definitions	130
sYmbol.Browse.Union	Browse unions	130
sYmbol.Browse.Var	Browse variables	131
sYmbol.CASE	Set symbol search mode	132
sYmbol.CHECK	Check database	132
sYmbol.Class	View class hierarchy	133
sYmbol.CLEANUP	Workarounds for redundant symbol information	134
sYmbol.CLEANUP.DOUBLES	Make ambiguous symbols unique	135
sYmbol.ColorCode	Enable color coding	135
sYmbol.ColorDef	Specify keyword colors	136
sYmbol.CREATE	Create and modify user-defined symbols	136
sYmbol.CREATE.ATTRibute	Create user-defined attribute	137
sYmbol.CREATE.Done	Finish symbol creation	137
sYmbol.CREATE.Function	Create user-defined function	138

sYmbol.CREATE.Label	Create user-defined symbol	139
sYmbol.CREATE.LocalVar	Create user-defined local variable	139
sYmbol.CREATE.MACRO	Create user-defined macro	140
sYmbol.CREATE.Module	Create user-defined module	140
sYmbol.CREATE.RESet	Erase all user-defined symbols	141
sYmbol.CREATE.Var	Create user-defined variable	141
sYmbol.CUTLINE	Limit size of text blocks	142
sYmbol.Delete	Delete symbols of one program	142
sYmbol.DeleteMACRO	Delete macro information	143
sYmbol.DeletePATtern	Delete labels from symbol database using wildcards	143
sYmbol.DEMangle	C++ demangler	143
sYmbol.DEOBFUSCATE	Deobfuscate global and static symbol	144
sYmbol.DONE	Finish load of symbols	144
sYmbol.ECA	ECA file management	145
sYmbol.ECA.BINary	Static preprocessing for code coverage	146
sYmbol.ECA.BINary.CollapseAll	Collapse all trees	146
sYmbol.ECA.BINary.ControlFlowMode.INSTR	Consider instrumentation	146
sYmbol.ECA.BINary.ControlFlowMode.Trace	Consider trace events	147
sYmbol.ECA.BINary.EditDecision	Modify start address of decision	147
sYmbol.ECA.BINary.ExpandAll	Expand all trees	148
sYmbol.ECA.BINary.EXPORT.AdJoinGAPS	Split up observability gaps	149
sYmbol.ECA.BINary.EXPORT.Decisions	Export decision details as CSV	149
sYmbol.ECA.BINary.EXPORT.GAPS	Export observability gaps to JSON	149
sYmbol.ECA.BINary.FilterMapped	Filter display by the mapping state	150
sYmbol.ECA.BINary.FilterType	Filter display by decision type	151
sYmbol.ECA.BINary.PROCESS	Static preprocessing for code coverage	151
sYmbol.ECA.BINary.SetCONDitionOffset	Set condition offset	153
sYmbol.ECA.BINary.SetDecisionState	Disable/Enable decision evaluation	154
sYmbol.ECA.BINary.view	Result of static preprocessing for code coverage	155
sYmbol.ECA.Delete	Delete loaded ECA data	157
sYmbol.ECA.Init	Clear gathered ECA data	157
sYmbol.ECA.List	List ECA file overview	157
sYmbol.ECA.LOAD	Load a single ECA file	160
sYmbol.ECA.LOADALL	Load all ECA files	161
sYmbol.FILTER.ADD.SOURCE	Add source files to filter	162
sYmbol.FILTER.ADD.sYmbol	Add symbols to filter	162
sYmbol.FILTER.Delete	Delete filter	163
sYmbol.ForEach	Symbol wildcard command	164
sYmbol.INFO	Display detailed information about debug symbol	165
sYmbol.LANGUAGE	Select language	168
sYmbol.List	Display list of all symbols	169
sYmbol.List.ATTRibute	Display memory attributes	169
sYmbol.List.BUILTIN	List built-in data types	169

sYmbol.List.ColorDef	List the keyword color definitions	170
sYmbol.List.Enum	List of enumeration constants	171
sYmbol.List.FRAME	Display frames	172
sYmbol.List.Function	Display functions	173
sYmbol.List.IMPORT	List imported symbols	173
sYmbol.List.InlineBlock	List inlined code blocks	174
sYmbol.List.InlineFunction	List inlined functions	174
sYmbol.List.LINE	Display source lines	175
sYmbol.List.Local	Display local symbols	176
sYmbol.List.MACRO	List all C macros	176
sYmbol.List.MAP	Display memory load map	177
sYmbol.List.Module	Display modules	177
sYmbol.List.PATCH	Display STF-symbol information	178
sYmbol.List.Program	Display programs	178
sYmbol.List.REFerence	Display reference information	179
sYmbol.List.SECtion	Display physical sections	180
sYmbol.List.SOURCE	Display source file names	181
sYmbol.List.SourceFunction	Display source to function relations	183
sYmbol.List.SOURCETREE	Display source files hierarchy	184
sYmbol.List.STACK	Display virtual stack	184
sYmbol.List.Static	Display static symbols	185
sYmbol.List.TREE	Display symbols in tree form	185
sYmbol.List.Type	Display data types	186
sYmbol.LSTLOAD	Load assembler source file	187
sYmbol.LSTLOAD.GHILLS	Load GHILLS assembler source file	187
sYmbol.LSTLOAD.HPASM	Load HP assembler source file	187
sYmbol.LSTLOAD.IAR	Load IAR assembler source file	189
sYmbol.LSTLOAD.INT68K	Load Intermetrics assembler source file	190
sYmbol.LSTLOAD.INTEL	Load INTEL assembler source file	190
sYmbol.LSTLOAD.INTEL2	Load INTEL assembler source file	191
sYmbol.LSTLOAD.KEIL	Load Keil assembler source file	191
sYmbol.LSTLOAD.MicroWare	Load MICROWARE assembler source file	191
sYmbol.LSTLOAD.MRI68K	Load MICROTEC assembler source file	193
sYmbol.LSTLOAD.OAK	Load OAK assembler source file	193
sYmbol.MARKER	Fine-tune the nested function run-time analysis	194
sYmbol.MARKER.Create	Marker for nesting function run-time analysis	195
sYmbol.MARKER.Delete	Delete a marker	199
sYmbol.MARKER.List	Displays the marker list	199
sYmbol.MARKER.RESet	Erase all markers	199
sYmbol.MARKER.TOUCH	Marker post-processing	200
sYmbol.MATCH	Symbol search mode	200
sYmbol.MEMory	Display memory usage	201
sYmbol.Modify	Modify symbols	202

sYmbol.Modify.Access	Modify access of symbols	202
sYmbol.Modify.ADDRess	Modify address of symbols	203
sYmbol.Modify.AddressToRange	Modify address of symbols	203
sYmbol.Modify.AlienFunction	Disable frame info for a function	204
sYmbol.Modify.ATTRibute	Modify memory attribute	204
sYmbol.Modify.CutFunction	Reduce function address information	204
sYmbol.Modify.NAME	Rename symbol	205
sYmbol.Modify.NAMES	Rename symbols	205
sYmbol.Modify.RangeToAddress	Modify address of symbols	206
sYmbol.Modify.RangeToFunction	Modify address range into function	206
sYmbol.Modify.SOURCE	Define source file	206
sYmbol.Modify.SplitFunction	Split function	207
sYmbol.Modify.StaticCOPY	Create static copy of local stack variables	207
sYmbol.Modify.StaticToStack	Change static variables	208
sYmbol.Modify.TYPE	Modify type of symbols	208
sYmbol.name	Display symbols	209
sYmbol.NAMESPACES	Search symbol in C++ namespace	211
sYmbol.NEW	Create new symbol	212
sYmbol.NEW.ATTRibute	Create user-defined memory attribute	212
sYmbol.NEW.Function	Create user-defined function	214
sYmbol.NEW.Label	Create user-defined symbol	215
sYmbol.NEW.LocalVar	Create user-defined local variable	216
sYmbol.NEW.MACRO	Create user-defined macro	216
sYmbol.NEW.Module	Create user-defined module	216
sYmbol.NEW.Var	Create user-defined variable	217
sYmbol.OVERLAY	Code overlay	218
sYmbol.OVERLAY.AutoID	Automatically determine overlay IDs	218
sYmbol.OVERLAY.Create	Declare code overlay section	220
sYmbol.OVERLAY.DETECT	Detect the current overlay status	224
sYmbol.OVERLAY.FRIEND	Declare a friend overlay segment	224
sYmbol.OVERLAY.List	Show declared code overlay sections	226
sYmbol.OVERLAY.RESet	Reset overlay declarations	226
sYmbol.PATCH	STF-symbol information	227
sYmbol.PATCH.DISable	Disable instrumentation code	227
sYmbol.PATCH.ENable	Enable instrumentation code	227
sYmbol.PATCH.List	Display STF-symbol information	228
sYmbol.POINTER	Define special register	230
sYmbol.POSTFIX	Set symbol postfix	230
sYmbol.PREFIX	Set symbol prefix	230
sYmbol.RELOCate	Relocate symbols	231
sYmbol.RELOCate.Auto	Control automatic relocation	231
sYmbol.RELOCate.Base	Define base address	232
sYmbol.RELOCate.List	List relocation info	232

sYmbol.RELOCate.Magic	Define program magic number	232
sYmbol.RELOCate.Passive	Define passive base address	233
sYmbol.RELOCate.shift	Relocate symbols	233
sYmbol.RESet	Clear symbol table	234
sYmbol.SourceBeautify	Beautify HLL lines on loading	235
sYmbol.SourceCONVert	Conversion for Japanese font	236
sYmbol.SourceLOAD	Initiate the loading of an HLL source file	237
sYmbol.SourcePATH	Source search path	238
sYmbol.SourcePATH.Delete	Delete path from search list	238
sYmbol.SourcePATH.DOWN	Make directory last in search order	239
sYmbol.SourcePATH.List	List source search paths	239
sYmbol.SourcePATH.RESet	Reset search path configuration	241
sYmbol.SourcePATH.Set	Define search path	242
sYmbol.SourcePATH.SetBaseDir	Define directory as base for relative paths	243
sYmbol.SourcePATH.SetCache	Internal use only	244
sYmbol.SourcePATH.SetCachedDir	Cache direct search path directory	244
sYmbol.SourcePATH.SetCachedDirCache	Internal use only	245
sYmbol.SourcePATH.SetCachedDirIgnoreCache	Cache direct search path	245
sYmbol.SourcePATH.SetDir	Define directory as direct search path	246
sYmbol.SourcePATH.SetDynamicDir	Adjust search order at hit	247
sYmbol.SourcePATH.SetMasterDir	Store cached files only relative	248
sYmbol.SourcePATH.SetRecurseDir	Define recursive direct search path	249
sYmbol.SourcePATH.SetRecurseDirCache	Internal use only	249
sYmbol.SourcePATH.SetRecurseDirIgnoreCase	Recursive search path	250
sYmbol.SourcePATH.Translate	Replace part of the source path	250
sYmbol.SourcePATH.TranslateSUBpath	Replace sub-path	252
sYmbol.SourcePATH.UP	Move path up in the search order	252
sYmbol.SourcePATH.Verbose	Display search details in message AREA	253
sYmbol.SourceRELOAD	Reload source files	254
sYmbol.STATE	Display statistic	254
sYmbol.STRIP	Set max. symbol length	255
sYmbol.TYPEINFO	Display information about a specific data type	255
sYmbol.View	Show symbol info	256
SYnch		257
SYnch	Synchronization mechanisms between different TRACE32 systems	257
Overview SYnch		257
SYnch.Connect	Connect to other TRACE32 PowerView instances	258
SYnch.MasterBreak	Invite other TRACE32 to stop synchronously	260
SYnch.MasterGo	Invite other TRACE32 to start synchronously	261
SYnch.MasterStep	Invite other TRACE32 to Asm step synchronously	261
SYnch.MasterSystemMode	Invite other TRACE32 to follow mode change	262
SYnch.OFF	Disable connection mechanism	262
SYnch.ON	Enable connection mechanism	262

SYnch.RESet	Reset SYnch mechanism	263
SYnch.SlaveBreak	Synchronize with stop in connected TRACE32	263
SYnch.SlaveGo	Synchronize with start in connected TRACE32	264
SYnch.SlaveStep	Synchronize with asm step in connected TRACE32	264
SYnch.SlaveSystemMode	Synch. with mode changes in other TRACE32	265
SYnch.state	Display current SYnch settings	265
SYnch.XTrack	Establish time synchronization to another TRACE32 instance	266
SYStem		268
SYStem	System configuration	268
SYStem.BdmClock	Select BDM clock	268
SYStem.BREAKTIMEOUT	Define the used timeout for break	269
SYStem.CADICommand	Send a command to target	270
SYStem.CADIconfig	CADI-specific setups	271
SYStem.CADIconfig.ExecSwOnly	Filter on executing software capability	271
SYStem.CADIconfig.RemoteServer	Define connection to CADI server	271
SYStem.CADIconfig.SpecRegDefine	Define special register set	273
SYStem.CADIconfig.SpecRegsOnly	Use only special defined register set	273
SYStem.CADIconfig.Traceconfig	Define network settings to CADI trace	274
SYStem.CADIconfig.TraceCore	Define core for CADI trace	274
SYStem.CONFIG	Configure debugger according to target topology	275
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	276
SYStem.CONFIG.CoreNumber	Set up number of hardware threads	282
SYStem.CONFIG.DEBUGPORT	Specify debugport	283
SYStem.CONFIG.DEBUGTIMESCALE	Extend debug driver timeouts	286
SYStem.CONFIG.ELA	Configure Embedded Logic Analyzer (ELA)	287
SYStem.CONFIG.ListCORE	Display the cores of a virtual target	287
SYStem.CONFIG.ListSIMulation	Display the simulations of a virtual target	288
SYStem.CONFIG.MULTITAP	Select type of JTAG multi-TAP network	289
SYStem.CONFIG.MULTITAP.JtagSEQUence	JTAG seq. on SYStem.Up	290
SYStem.CONFIG.state	Display target configuration	292
SYStem.CONFIG.TRACEPORT	Declare trace source and trace port type	293
SYStem.CONFIG.TRANSACTORPIPENAME	Set up pipe name	294
SYStem.CONFIG.USB	USB configuration	294
SYStem.CONFIG.XCP	XCP specific settings	294
SYStem.CPU	Select CPU	296
SYStem.CpuAccess	Run-time memory access (intrusive)	297
SYStem.CpuBreak	Master control to deny stopping the target (long stop)	298
SYStem.CpuSpot	Master control to deny spotting the target (short stop)	299
SYStem.DCI	DCI configuration	299
SYStem.DETECT	Detect target system resources	300
SYStem.DLLCommand	Custom DLL connection to target	304
SYStem.InfineonDAS	Configure the InfineonDAS debug port	305
SYStem.IRISconfig	IRIS-specific setups	306

SYStem.IRISconfig.RemoteServer	Define connection to IRIS server	306
SYStem.JtagClock	Define JTAG frequency	307
SYStem.LOG	Log read and write accesses to the target	308
SYStem.LOG.CLEAR	Clear the 'SYStem.LOG.List' window	309
SYStem.LOG.CLOSE	Close the system log file	310
SYStem.LOG.Init	Clear the 'SYStem.LOG.List' window	310
SYStem.LOG.List	Log the accesses made by TRACE32	311
SYStem.LOG.Mode	Set logging mode	312
SYStem.LOG.OFF	Pause logging	313
SYStem.LOG.ON	Resume logging	313
SYStem.LOG.OPEN	Open a system log file	314
SYStem.LOG.RESet	Reset configuration of system log to defaults	314
SYStem.LOG.Set	Select the TRACE32 accesses to be logged	315
SYStem.LOG.SIZE	Define number of lines in the 'SYStem.LOG.List' window	316
SYStem.LOG.state	Open configuration window of system log	317
SYStem.LOG.StopOnError	Stop logging on error	318
SYStem.MCDCommand	Send command to MCD server	318
SYStem.MCDconfig	Send configuration to MCD server	319
SYStem.MemAccess	Select run-time memory access method	320
SYStem.Mode	Select mode	321
SYStem.Option	Special setup	321
SYStem.Option.IMASKASM	Disable interrupts while single stepping	322
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	322
SYStem.Option.MACHINESPACES	Address extension for guest OSes	322
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	323
SYStem.Option.ZoneSPACES	Enable symbol management for zones	324
SYStem.PAUSE	Pause the execution of operations	325
SYStem.POLLING	Polling mode of CPU	326
SYStem.PORT	Configure external communication interface	327
SYStem.RESet	Reset configuration	328
SYStem.RESetOut	Reset peripherals	328
SYStem.RESetTarget	Release target reset	328
SYStem.state	Display SYStem.state window	329
SYStem.TARGET	Set target IP name or address	330
SYStem.VirtualTiming	Modify timing constraints	331
SYStem.VirtualTiming.HardwareTimeout	Disable/enable hardware timeout	332
SYStem.VirtualTiming.HardwareTimeoutScale	Multiply hardware timeout	332
SYStem.VirtualTiming.InternalClock	Base for artificial time calculation	333
SYStem.VirtualTiming.MaxPause	Limit pause	334
SYStem.VirtualTiming.MaxTimeout	Override time-outs	334
SYStem.VirtualTiming.OperationPause	Insert a pause after each operation	335
SYStem.VirtualTiming.PauseinTargetTime	Set up pause time-base	335
SYStem.VirtualTiming.PauseScale	Multiply pause with a factor	336

SYStem.VirtualTiming.PollingPause	Advance emulation time when polling	336
SYStem.VirtualTiming.TimeinTargetTime	Set up general time-base	337
SYStem.VirtualTiming.TimeScale	Multiply time-base with a factor	338
SystemTrace		339
SystemTrace	MIPI STP and CoreSight ITM	339
SystemTrace.state	Open system-trace configuration window	341

General Commands Reference Guide T

General Commands Reference Guide T	(general_ref_t.pdf)	1
History		15
TargetSystem		16
TargetSystem	TRACE32 PowerView instances	16
TargetSystem.NewInstance	Start new TRACE32 PowerView instance	17
TargetSystem.state	Show overview of multicore system	21
TASK		26
TASK	OS Awareness for TRACE32	26
Overview TASK		26
TASK.ACCESS	Control memory access	31
TASK.ATTACH	Attach to a running process	31
TASK.Break	Stop the execution of a single task or thread	31
TASK.CACHEFLUSH	Reread task list	32
TASK.CONFIG	Configure OS Awareness	32
TASK.COPYDOWN	Copy file from host into target	33
TASK.COPYUP	Copy file from target into host	34
TASK.Create	Create task	35
TASK.Create.MACHINE	Define a manual machine	35
TASK.Create.RUNNABLE	Define an AUTOSAR runnable	36
TASK.Create.SPACE	Define a manual MMU space	37
TASK.Create.task	Define a manual task	39
TASK.CreateExtraID	Create a virtual task	40
TASK.CreateID	Create virtual task	40
TASK.DELete	Delete file from target	40
TASK.DeletelD	Delete virtual task	41
TASK.DETACH	Detach from task	41
TASK.Go	Start the execution of a single task or thread	41
TASK.INSTALL	Deprecated	42
TASK.KILL	End task	42
TASK.List	Information about tasks	43
TASK.List.MACHINES	List machines	43
TASK.List.RUNNABLES	List AUTOSAR runnables	44
TASK.List.SPACES	List MMU spaces	44

TASK.List.tasks	List all running tasks	45
TASK.List.TREE	Display tasks in a tree structure	46
TASK.ListID	List virtual tasks	47
TASK.NAME	Translation of task magic number to task name	48
TASK.NAME.DELeTe	Delete a task name table entry	48
TASK.NAME.RESet	Reset task name table	48
TASK.NAME.Set	Set a task name table entry	49
TASK.NAME.view	Show task name translation table	49
TASK.ORTI	AUTOSAR/OSEK support	50
TASK.ORTI.CPU	Set OSEK SMP CPU number	50
TASK.ORTI.load	Configure OS Awareness for OSEK/ORTI	50
TASK.ORTI.NOSTACK	Exclude an ORTI task from stack evaluation	51
TASK.ORTI.SPLITSTACK	Split stack analysis of idle ORTI task to cores	52
TASK.RELOAD	Reread task list	53
TASK.RESet	Reset OS Awareness	53
TASK.RUN	Load task	54
TASK.select	Display context of specified task	55
TASK.SETDIR	Set the awareness directory	56
TASK.STacK	Stack usage coverage	57
TASK.STacK.ADD	Add stack space coverage	57
TASK.STacK.DIRection	Define stack growth direction	59
TASK.STacK.Init	Initialize unused stack space	59
TASK.STacK.PATtern	Define stack check pattern	60
TASK.STacK.PATternGAP	Define check pattern gap	61
TASK.STacK.ReMove	Remove stack space coverage	61
TASK.STacK.RESet	Reset stack coverage	62
TASK.STacK.view	Open stack space coverage	63
TCB		65
TCB	Trace control block	65
TCB.AllBranches	Broadcast all branches	66
TCB.CPU	Broadcast information for specified CPU only	66
TCB.CycleAccurate	Cycle accurate tracing	67
TCB.DataTrace	Broadcast specified address and data information	68
TCB.EX	Broadcast exception level information	69
TCB.FCR	Broadcast function call-return information	69
TCB.IM	Broadcast instruction cache miss information	69
TCB.InstructionCompletionSizeBits	Specify size of completion message	70
TCB.KE	Broadcast kernel mode information	70
TCB.LSM	Broadcast load store data cache information	71
TCB.OFF	Switch TCB off	71
TCB.ON	Switch TCB on	71
TCB.PCTrace	Broadcast program counter trace	72
TCB.PortMode	Specify trace clock ratio	73

TCB.PortWidth	Specify trace port width	73
TCB.Register	Display TCB control register	74
TCB.RESet	Reset TCB setup to default	75
TCB.SourceSizeBits	Specify number of bit for core information in trace	75
TCB.SRC	Control selective trace	75
TCB.STALL	Stall CPU for complete trace	76
TCB.state	Display TCB setup	76
TCB.SV	Broadcast supervisor mode information	77
TCB.SyncPeriod	Specify TCB sync period	77
TCB.TC	Broadcast information for specified HW thread	78
TCB.ThreadSizeBits	Specify number of bit for thread information in trace	78
TCB.Type	Specify TCB type	79
TCB.UM	Broadcast user mode information	79
TCB.Version	Specify trace cell version	80
TERM		81
TERM	Terminal emulation	81
Overview TERM		81
TERM.CLEAR	Clear terminal window	85
TERM.CLOSE	Close files	85
TERM.CMDLINE	Specify a command line	85
TERM.GATE	Terminal with virtual hosting	86
TERM.HARDCOPY	Print terminal window contents	86
TERM.HEAPINFO	Define memory heap parameters	87
TERM.LocalEcho	Enables/disables local echo for new terminal windows	87
TERM.METHOD	Select terminal protocol	88
TERM.METHOD2	Select additional terminal protocol	91
TERM.Mode	Define terminal type	93
TERM.Out	Send data to virtual terminal	94
TERM.OutBREAK	Send serial break	94
TERM.PIPE	Connect terminal to named pipe	95
TERM.PipeREAD	Connect terminal input to named pipe	95
TERM.PipeWRITE	Connect terminal output to named pipe	95
TERM.PULSE	Enable pulse generator for transfers	96
TERM.READ	Get terminal input from file	97
TERM.RESet	Reset terminal parameters	97
TERM.SCROLL	Enable automatic scrolling for terminal window	97
TERM.SIZE	Define size of terminal window	98
TERM.STDIN	Get terminal input from file	98
TERM.TCP	Route terminal input/output to TCP port	99
TERM.TELNET	Open TELNET terminal window	99
TERM.TRIGGER	Trigger on string in terminal window	100
TERM.view	Terminal display	102
TERM.WRITE	Write terminal output to file	103

TPIU		104
TPIU	Trace Port Interface Unit (TPIU)	104
Overview TPIU		104
TPIU.CLEAR	Re-write the TPIU registers	105
TPIU.IGNOREZEROS	Workaround for a special chip	105
TPIU.NOFLUSH	Workaround for a chip bug affecting TPIU flush	105
TPIU.PortClock	Inform debugger about HSSTP trace frequency	106
TPIU.PortMode	Select the operation mode of the TPIU	107
TPIU.PortSize	Select interface type and port size of the TPIU	109
TPIU.RefClock	Set up reference clock for HSSTP	110
TPIU.Register	Display TPIU registers	111
TPIU.RESet	Reset TPIU settings	111
TPIU.state	Display TPIU configuration window	112
TPIU.SWVPrescaler	Set up SWV prescaler	112
TPIU.SWVZEROS	Workaround for a chip bug	113
TPIU.SyncPeriod	Set period of sync packet injection	114
TPU		115
TPU.BASE	Base address	115
TPU.Break	Break TPU	115
TPU.Dump	Memory display	115
TPU.Go	Start TPU	115
TPU.List	View microcode	115
TPU.ListEntry	Table display	115
TPU.Register.ALL	Register operation mode	115
TPU.Register.NEWSTEP	New debugging mode	116
TPU.Register.Set	Register modification	116
TPU.Register.view	Register display	116
TPU.RESet	Disable TPU debugger	116
TPU.SCAN	Scannig TPU	116
TPU.SELect	Select TPU for debugging	116
TPU.Step	Single step TPU	116
TPU.view	View TPU channels	116
Trace		117
Trace	Trace configuration and display	117
Overview Trace		118
<trace>.ACCESS	Define access path to program code for trace decoding	131
<trace>.Arm	Arm the trace	134
<trace>.AutoArm	Arm automatically	135
<trace>.AutoFocus	Calibrate AUTOFOCUS preprocessor	135
<trace>.AutoInit	Automatic initialization	140
<trace>.AutoStart	Automatic start	140
<trace>.BookMark	Set a bookmark in trace listing	140
<trace>.BookMarkToggle	Toggles a single trace bookmark	143

<trace>.Chart	Display trace contents graphically	144
<trace>.Chart.Address	Time between program events as a chart	153
<trace>.Chart.AddressGROUP	Address group time chart	155
<trace>.Chart.ChildTREE	Display callee context of a function as chart	156
<trace>.Chart.DatasYmbol	Analyze pointer contents graphically	157
<trace>.Chart.DistriB	Distribution display graphically	159
<trace>.Chart.Func	Function activity chart	161
<trace>.Chart.GROUP	Group activity chart	162
<trace>.Chart.INTERRUPT	Display interrupt chart	163
<trace>.Chart.INTERRUPTTREE	Display interrupt nesting	164
<trace>.Chart.Line	Graphical HLL lines analysis	165
<trace>.Chart.MODULE	Code execution broken down by module as chart	166
<trace>.Chart.Nesting	Show function nesting at cursor position	167
<trace>.Chart.PAddress	Which instructions accessed data address	168
<trace>.Chart.PROGRAM	Code execution broken down by program	169
<trace>.Chart.PsYmbol	Shows which functions accessed data address	170
<trace>.Chart.RUNNABLE	Runnable activity chart	172
<trace>.Chart.sYmbol	Symbol analysis	173
<trace>.Chart.TASK	Task activity chart	176
<trace>.Chart.TASKFunc	Task related function run-time analysis (legacy)	177
<trace>.Chart.TASKINFO	Context ID special messages	177
<trace>.Chart.TASKINTR	Display ISR2 time chart (ORTI)	178
<trace>.Chart.TASKKernel	Task run-time chart with kernel markers (flat)	179
<trace>.Chart.TASKORINTERRUPT	Task and interrupt activity chart	180
<trace>.Chart.TASKORINTRState	Task and ISR2 state analysis	181
<trace>.Chart.TASKSRV	Service routine run-time analysis	182
<trace>.Chart.TASKState	Task state analysis	183
<trace>.Chart.TASKVSINTERRUPT	Time chart of interrupted tasks	185
<trace>.Chart.TASKVSINTR	Time chart of task-related interrupts	186
<trace>.Chart.TREE	Display function chart as tree view	187
<trace>.Chart.Var	Variable chart	188
<trace>.Chart.VarState	Variable activity chart	189
<trace>.CLOCK	Clock to calculate time out of cycle count information	191
<trace>.ComPare	Compare trace contents	192
<trace>.ComPareCODE	Compare trace with memory	194
<trace>.CustomTrace	Custom trace	195
<trace>.CustomTrace.<label>.COMMAND	Send command to specific DLL	195
<trace>.CustomTrace.<label>.ListString	Display ASCII strings	195
<trace>.CustomTrace.<label>.UNLOAD	Unload a single DLL	196
<trace>.CustomTraceLoad	Load a DLL for trace analysis/Unload all DLLs	196
<trace>.DISable	Disable the trace	197
<trace>.DisConfig	Trace disassembler configuration	198
<trace>.DisConfig.CYcle	Trace disassemble setting	198

<trace>.DisConfig.FlowMode	Enable FlowTrace analysis	200
<trace>.DisConfig.RESet	Reset trace disassemble setting	200
<trace>.DRAW	Plot trace data against time	201
<trace>.DRAW.channel	Plot no-data values against time	204
<trace>.DRAW.Data	Plot data values against time	206
<trace>.DRAW.Var	Plot variable values against time	210
<trace>.EXPORT	Export trace data for processing in other applications	212
<trace>.EXPORT.ARTI	Export trace data as ARTI for CP	213
<trace>.EXPORT.ARTIAP	Export trace data as ARTI for AP	214
<trace>.EXPORT.Ascii	Export trace data as ASCII	215
<trace>.EXPORT.Bin	Export trace data as binary file	216
<trace>.EXPORT.BRANCHFLOW	Export branch events from trace data	218
<trace>.EXPORT.CSVFunc	Export the function nesting to a CSV file	219
<trace>.EXPORT.cycles	Export trace data	220
<trace>.EXPORT.Func	Export function nesting	223
<trace>.EXPORT.MDF	Export trace data as MDF	224
<trace>.EXPORT.MTV	Export in MCDS Trace Viewer format	225
<trace>.EXPORT.TASK	Export task switches	226
<trace>.EXPORT.TASKEVENTS	Export task event to CSV	227
<trace>.EXPORT.TracePort	Export trace packets as recorded at trace port	228
<trace>.EXPORT.VCD	Export trace data in VCD format	230
<trace>.EXPORT.VERILOG	Export trace data in VERILOG format	231
<trace>.EXPORT.VHDL	Export trace data in VHDL format	232
<trace>.ExtractCODE	Extract code from trace	232
<trace>.FILE	Load a file into the file trace buffer	233
<trace>.Find	Find specified entry in trace	235
<trace>.FindAll	Find all specified entries in trace	237
<trace>.FindChange	Search for changes in trace flow	238
<trace>.FindProgram	Advanced trace search	239
<trace>.FindReProgram	Activate advanced existing trace search program	240
<trace>.FindViewProgram	State of advanced trace search programming	240
<trace>.FLOWPROCESS	Process flowtrace	241
<trace>.FLOWSTART	Restart flowtrace processing	241
<trace>.Get	Display input level	242
<trace>.GOTO	Move cursor to specified trace record	244
<trace>.Init	Initialize trace	246
<trace>.JOINFILE	Concatenate several trace recordings	246
<trace>.List	List trace contents	248
<trace>.ListNesting	Analyze function nesting	263
<trace>.ListVar	List variable recorded to trace	266
<trace>.LOAD	Load trace file for offline processing	270
<trace>.MERGEFILE	Combine two trace files into one	272
Trace.METHOD	Select trace method	273

<trace>.Mode	Set the trace operation mode	276
<trace>.OFF	Switch off	278
<trace>.PipeWRITE	Connect to a named pipe to stream trace data	278
<trace>.PlatformCLOCK	Set clock for platform traces	278
<trace>.PortFilter	Specify utilization of trace memory	279
<trace>.PortSize	Set external port size	280
<trace>.PortType	Specify trace interface	280
<trace>.PROfile	Rolling live plots of trace data	282
<trace>.PROfile.channel	Display profile of signal probe channels	282
<trace>.PROfile.CTU	Display complex trigger unit counter profile	282
<trace>.PROfileChart	Profile charts	283
<trace>.PROfileChart.Address	Address profile chart	289
<trace>.PROfileChart.AddressGROUP	Address group time chart	290
<trace>.PROfileChart.AddressRate	Address rate profile chart	292
<trace>.PROfileChart.COUNTER	Display a profile chart	293
<trace>.PROfileChart.DatasYmbol	Analyze pointer contents graphically	295
<trace>.PROfileChart.DIStance	Time interval for a single event	296
<trace>.PROfileChart.DistriB	Distribution display in time slices	297
<trace>.PROfileChart.DURation	Time between two events	298
<trace>.PROfileChart.GROUP	Group profile chart	301
<trace>.PROfileChart.INTERRUPT	Display interrupt profile chart	302
<trace>.PROfileChart.Line	HLL-line profile chart	303
<trace>.PROfileChart.MODULE	Module profile chart	304
<trace>.PROfileChart.PAddress	Which instructions accessed data address	305
<trace>.PROfileChart.PROGRAM	Program profile chart	306
<trace>.PROfileChart.PsYmbol	Which functions accessed data address	307
<trace>.PROfileChart.Rate	Event frequency	309
<trace>.PROfileChart.RUNNABLE	Runnable profile chart	311
<trace>.PROfileChart.sYmbol	Dynamic program behavior graphically (flat)	312
<trace>.PROfileChart.TASK	Dynamic task behavior graphically (flat)	313
<trace>.PROfileChart.TASKINFO	Context ID special messages	314
<trace>.PROfileChart.TASKINTR	ISR2 profile chart (ORTI)	315
<trace>.PROfileChart.TASKKernel	Task profile chart with kernel markers	316
<trace>.PROfileChart.TASKORINTERRUPT	Task and interrupt profile chart	317
<trace>.PROfileChart.TASKSRV	Profile chart of OS service routines	318
<trace>.PROfileChart.TASKVSINTERRUPT	Interrupted tasks	319
<trace>.PROfileChart.TASKVSINTR	Profile chart for task-related interrupts	320
<trace>.PROfileChart.Var	Variable profile chart	321
<trace>.PROfileSTATistic	Statistical analysis in a table versus time	322
<trace>.PROfileSTATistic.Address	Statistical analysis for addresses	325
<trace>.PROfileSTATistic.AddressGROUP	Stat. for address groups	325
<trace>.PROfileSTATistic.COUNTER	Statistical analysis for counter	326
<trace>.PROfileSTATistic.DatasYmbol	Statistic analysis for pointer content	326

<trace>.PROfileSTATistic.DistriB	Distribution statistical analysis	327
<trace>.PROfileSTATistic.GROUP	Statistical analysis for groups	328
<trace>.PROfileSTATistic.INTERRUPT	Statistical analysis for interrupts	329
<trace>.PROfileSTATistic.Line	Statistical analysis for HLL lines	330
<trace>.PROfileSTATistic.MODULE	Statistical analysis for modules	331
<trace>.PROfileSTATistic.PAddress	Which instr. accessed data address	332
<trace>.PROfileSTATistic.PROGRAM	Statistical analysis for programs	332
<trace>.PROfileSTATistic.PsYmbol	Which functions accessed data address	333
<trace>.PROfileSTATistic.RUNNABLE	Statistical analysis for runnables	333
<trace>.PROfileSTATistic.sYmbol	Statistical analysis for symbols	334
<trace>.PROfileSTATistic.TASK	Statistical analysis for tasks	335
<trace>.PROfileSTATistic.TASKINFO	Context ID special messages	335
<trace>.PROfileSTATistic.TASKINTR	Statistical analysis for ISR2 (ORTI)	336
<trace>.PROfileSTATistic.TASKKernel	Stat. analysis with kernel markers	337
<trace>.PROfileSTATistic.TASKORINTERRUPT	Interrupts and tasks	337
<trace>.PROfileSTATistic.TASKSRV	Analysis of OS service routines	338
<trace>.PROfileSTATistic.TASKVSINTERRUPT	Interrupted tasks	338
<trace>.PROTOcol	Protocol analysis	339
<trace>.PROTOcol.Chart	Graphic display for user-defined protocol	339
<trace>.PROTOcol.Draw	Graphic display for user-defined protocol	341
<trace>.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	342
<trace>.PROTOcol.Find	Find in trace buffer for user-defined protocol	343
<trace>.PROTOcol.list	Display trace buffer for user-defined protocol	344
<trace>.PROTOcol.PROfileChart	Profile chart for user-defined protocol	347
<trace>.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	348
<trace>.PROTOcol.STATistic	Display statistics for user-defined protocol	350
Protocol specific Options		351
<trace>.REF	Set reference point for time measurement	357
<trace>.RESet	Reset command	357
<trace>.SAVE	Save trace for postprocessing in TRACE32	358
<trace>.SelfArm	Automatic restart of trace recording	362
<trace>.ShowFocus	Display data eye for AUTOFOCUS preprocessor	365
<trace>.ShowFocusClockEye	Display clock eye	368
<trace>.ShowFocusEye	Display data eye	370
<trace>.SIZE	Define buffer size	373
<trace>.SnapShot	Restart trace capturing once	373
<trace>.SPY	Adaptive stream and analysis	374
<trace>.state	Display trace configuration window	376
<trace>.STATistic	Statistic analysis	378
<trace>.STATistic.Address	Time between up to 8 program events	387
<trace>.STATistic.AddressDIStance	Time interval for single program event	388
<trace>.STATistic.AddressDURation	Time between two program events	389
<trace>.STATistic.AddressGROUP	Address group run-time analysis	391

<trace>.STATistic.ChildTREE	Show callee context of a function	393
<trace>.STATistic.COLOR	Assign colors to function for colored graphics	394
<trace>.STATistic.CYcle	Analyze cycle types	395
<trace>.STATistic.DatasYmbol	Analyze pointer contents numerically	398
<trace>.STATistic.DISTance	Time interval for a single event	400
<trace>.STATistic.DistriB	Distribution analysis	401
<trace>.STATistic.DURATION	Time between two events	402
<trace>.STATistic.FIRST	Start point for statistic analysis	404
<trace>.STATistic.Func	Nesting function runtime analysis	406
<trace>.STATistic.FuncDURATION	Statistic analysis of single function	422
<trace>.STATistic.FuncDURATIONInternal	Statistic analysis of single func.	423
<trace>.STATistic.GROUP	Group run-time analysis	424
<trace>.STATistic.Ignore	Ignore false records in statistic	426
<trace>.STATistic.INTERRUPT	Interrupt statistic	427
<trace>.STATistic.InterruptIsFunction	Statistics interrupt processing	428
<trace>.STATistic.InterruptIsKernel	Statistics interrupt processing	430
<trace>.STATistic.InterruptIsKernelFunction	Statistics interrupt processing	430
<trace>.STATistic.InterruptIsTaskswitch	Statistics interrupt processing	430
<trace>.STATistic.INTERRUPTTREE	Display interrupt nesting	431
<trace>.STATistic.LAST	End point for statistic analysis	433
<trace>.STATistic.Line	High-level source code line analysis	435
<trace>.STATistic.LINKage	Per caller statistic of function	436
<trace>.STATistic.Measure	Analyze the performance of a single signal	438
<trace>.STATistic.MODULE	Code execution broken down by module	440
<trace>.STATistic.PAddress	Which instructions accessed data address	441
<trace>.STATistic.ParentTREE	Show the call context of a function	442
<trace>.STATistic.PROCESS	Re-process statistics	444
<trace>.STATistic.PROGRAM	Code execution broken down by program	445
<trace>.STATistic.PsYmbol	Shows which functions accessed data address	446
<trace>.STATistic.RUNNABLE	Runnable runtime analysis	448
<trace>.STATistic.RUNNABLEDURATION	Runnable duration analysis	449
<trace>.STATistic.Sort	Specify sorting criteria for statistic commands	450
<trace>.STATistic.sYmbol	Flat run-time analysis	458
<trace>.STATistic.TASK	Task activity statistic	461
<trace>.STATistic.TASKFunc	Task related function run-time analysis	464
<trace>.STATistic.TASKINFO	Context ID special messages	464
<trace>.STATistic.TASKINTR	ISR2 statistic (ORTI)	465
<trace>.STATistic.TASKKernel	Task analysis with kernel markers (flat)	466
<trace>.STATistic.TASKLOCK	Analyze lock accesses from tasks	469
<trace>.STATistic.TASKORINTERRUPT	Statistic of interrupts and tasks	470
<trace>.STATistic.TASKORINTRState	Task and ISR2 statistic analysis	471
<trace>.STATistic.TASKSRV	Analysis of time in OS service routines	472
<trace>.STATistic.TASKState	Performance analysis	474

<trace>.STATistic.TASKStateDURation	Task state runtime analysis	478
<trace>.STATistic.TASKTREE	Tree display of task specific functions	479
<trace>.STATistic.TASKVSINTERRUPT	Statistic of interrupts, task-related	480
<trace>.STATistic.TASKVSINTR	ISR2 statistic (ORTI), task related	481
<trace>.STATistic.TREE	Tree display of nesting function run-time analysis	482
<trace>.STATistic.Use	Use records	483
<trace>.STATistic.Var	Statistic of variable accesses	484
<trace>.STREAMCompression	Select compression mode for streaming	485
<trace>.STREAMFILE	Specify temporary streaming file path	486
<trace>.STREAMFileLimit	Set size limit for streaming file	487
<trace>.STREAMLOAD	Load streaming file from disk	488
<trace>.STREAMSAVE	Save streaming file to disk	490
<trace>.TCount	Set trigger counter	490
<trace>.TDelay	Trigger delay	491
<trace>.TERMination	Use trace line termination of preprocessor	493
<trace>.TestFocus	Test trace port recording	494
<trace>.TestFocusClockEye	Scan clock eye	496
<trace>.TestFocusEye	Check signal integrity	497
<trace>.TestUtilization	Tests trace port utilization	497
<trace>.THreshold	Optimize threshold for trace lines	498
<trace>.Timing	Waveform of trace buffer	499
<trace>.TMode	Select trigger mode	501
<trace>.TraceCONNECT	Select on-chip peripheral sink	501
<trace>.TRACK	Set tracking record	502
<trace>.TRIGGER	Trigger the trace	502
<trace>.TSElect	Select trigger source	503
<trace>.View	Display single record	504
<trace>.ZERO	Align timestamps of trace and timing analyzers	505
TRACEPORT		506
TRACEPORT	Configure trace hardware	506
TRACEPORT.EndsKiP	Define number of bytes skipped at the end of frame	507
TRACEPORT.LaneCount	Select port size of the trace port	508
TRACEPORT.LanePolarity	Set polarity for each lane of the trace port	508
TRACEPORT.LaneSpeed	Inform debugger about trace port rate	509
TRACEPORT.MsgBlitEndian	Change bit-order within each byte	510
TRACEPORT.MsgBYteEndian	Change byte-order within each word	511
TRACEPORT.MsgLongEndian	Change dword-order within each qword	511
TRACEPORT.MsgWOrdEndian	Change word-order within each dword	512
TRACEPORT.OSCFrequency	Set OSC clock frequency	512
TRACEPORT.PinReMap	Adapt the lane order of the trace port	513
TRACEPORT.RefCLOCK	Set up reference clock for trace port	514
TRACEPORT.RESet	Reset trace port configuration	514
TRACEPORT.StartsKiP	Define number of bytes skipped at the start of frame	515

TRACEPORT.state	Display trace port configuration window	516
TRANSLation		517
TRANSLation	Debugger address translation	517
Overview TRANSLation		517
TRANSLation.AutoEnable	Auto-enable debugger MMU translation	520
TRANSLation.AutoSCAN	Autoscan feature for debugger MMU	520
TRANSLation.CacheFlush	Flush TRACE32 address translation cache	521
TRANSLation.CLEANUP	Clean up MMU table	521
TRANSLation.COMMON	Common address ranges for kernel and tasks	522
TRANSLation.COMMON.ADD	Add another common address range	524
TRANSLation.COMMON.CLEAR	Clear all common logical address ranges	524
TRANSLation.Create	Create translation	525
TRANSLation.CreateID	Add entry to MMU space ID table	526
TRANSLation.CreateTab	Create multiple translations	526
TRANSLation.Delete	Delete translation	527
TRANSLation.DeleteID	Remove entry from MMU space ID table	527
TRANSLation.List	List MMU translation table	528
TRANSLation.ListID	List MMU space ID table	529
TRANSLation.NoProtect	Unprotect memory	529
TRANSLation.OFF	Deactivate debugger address translation	530
TRANSLation.ON	Activate debugger address translation	530
TRANSLation.PAGER	Allow paged breakpoints for Linux	531
TRANSLation.Protect	Protect memory	532
TRANSLation.Protect.ADD	Add range to protected memory ranges	532
TRANSLation.Protect.OFF	Switch protection of target memory off	533
TRANSLation.Protect.ON	Protect entire target memory	534
TRANSLation.RESet	Reset MMU configuration	535
TRANSLation.SCANall	Scan MMU tables	535
TRANSLation.ScanID	Scan MMU address space tables from kernel	536
TRANSLation.SHADOW	Enable shadow access to target memory	536
TRANSLation.state	Overview of translation settings	537
TRANSLation.TableWalk	Automatic MMU page table walk	538
TRANSLation.TlbAutoScan	Allow automatic TLB scans during table walk	539
TRANSLation.TRANSPARENT	Transparent banking area	541
TrBus		542
TrBus	Trigger bus	542
Overview TrBus		542
TrBus.Arm	Arm the trigger bus	547
TrBus.Connect	Configure TRIGGER as input or output	548
TrBus.Mode	Define polarity/edge for the trigger signal	548
TrBus.OFF	Switch trigger bus off	548
TrBus.Out	Define source for the external trigger pulse	549
TrBus.RESet	Reset setting for trigger bus	549

TrBus.Set	Define the target for the incoming trigger	550
TrBus.state	Display settings for the trigger bus	550
TrBus.Trigger	Stimulate a trigger on the trigger bus	550
TrOnchip		551
TrOnchip	Onchip triggers	551
TrOnchip.RESet	Reset settings to defaults	551
TrOnchip.state	Display onchip trigger window	551
TrPOD		553
TrPOD	Trigger probe	553
TrPOD.Clock	Defines data mask	553
TrPOD.ClockPOL	Defines data polarity	553
TrPOD.Data	Defines data mask	554
TrPOD.DataPOL	Defines data polarity	554
TrPOD.Mode	Defines data polarity	555
TrPOD.OFF	Switch off	556
TrPOD.ON	Switch on	556
TrPOD.RESet	Reset command	556
TrPOD.state	State display	557
TrPOD.Time	Defines the time for the pulse width trigger	557

General Commands Reference Guide U

General Commands Reference Guide U	(general_ref_u.pdf)	1

General Commands Reference Guide V

General Commands Reference Guide V	(general_ref_v.pdf)	1
History		5
Var		6
Var	HLL variables and expressions	6
Overview Var		6
Var.AddSticker	Add variable sticker to source listing window	23
Var.AddWatch	Add variable to Var.Watch window	24
Var.AddWatchPATtern	Add variables to Var.Watch window using wildcards	24
Var.Assign	Assignment to a variable	25
Var.Break	Breakpoint on variable	26
Var.Break.Delete	Delete breakpoint on variable	26
Var.Break.direct	Set temporary breakpoint on HLL expression	28
Var.Break.Pass	Define pass condition for breakpoint	29
Var.Break.Set	Set breakpoint to HLL expression	30
Var.Call	Call a new procedure	31
Var.CHAIN	Display linked list	32
Var.DelWatch	Delete variable from watch	33

Var.DRAW	Graphical variable display	33
Var.DRAWXY	Graphical variable display	37
Var.DUMP	Memory dump	38
Var.Eval	Evaluate high-level expression	39
Var.EXPORT	Export variables in CSV format to file	39
Var.FixedCHAIN	Display linked list	41
Var.FixedTABLE	Display table	41
Var.Go	Real-time emulation	43
Var.Go.Back	Re-run program backwards until variable access (CTS)	43
Var.Go.Change	Real-time emulation till expression changes	44
Var.Go.direct	Real-time emulation with breakpoint	45
Var.Go.Till	Real-time emulation till expression true	46
Var.IF	PRACTICE conditional branching	47
Var.INFO	View information about HLL variable or HLL expression	48
Var.Local	Local variables	49
Var.LOG	Log variables	50
Var.NEW	Creates a TRACE32-internal variable	52
Var.NEWGLOBAL	Creates a global TRACE32-internal variable	53
Var.NEWLOCAL	Creates a local TRACE32-internal variable	54
Var.OBJECT	Pretty printing for C++ objects	56
Var.PATtern	Display variables allowing wildcards for symbol name and type	58
Var.PRINT	Display variables	59
Var.PROfile	Graphical display of variable	60
Var.Ref	Referenced variables	61
Var.set	Modify variable	62
Var.Step	Step	65
Var.Step.BackChange	Step back till expression changes	65
Var.Step.BackTill	Step back till expression true	65
Var.Step.Change	Step till expression changes	66
Var.Step.Till	Step till expression true	66
Var.TABLE	Display table	67
Var.TREE	Display variables in the form of a tree structure	68
Var.TYPE	Display variable types	69
Var.View	Display variables	70
Var.Watch	Open Var.Watch window	72
Var.WHILE	PRACTICE loop construction	73
Var.WRITE	Write variables to file	74
VCO		75
VCO	Clock generator	75
VCO.BusFrequency	Control bus clock	75
VCO.Down	Frequency down	75
VCO.Frequency	Control VCO clock	76
VCO.Rate	VCO rate	76

VCO.RESet	VCO reset	77
VCO.state	State display	77
VCO.TimeBaseFrequency	Set the time base clock	77
VCO.Up	Frequency up	78
VCU		79
VCU	VCU registers (Vector Computational Unit)	79
VCU.Init	Initialize VCU registers	79
VCU.RESet	Reset VCU registers	79
VCU.Set	Set VCU register	80
VCU.view	Display VCU registers	80
VE		81
VE	Virtual execution mode	81
VE.OFF	Turn off virtual execution mode	81
VE.ON	Turn on virtual execution mode	81
VPU		82
VPU	Vector Processing Unit (VPU)	82
VPU.Init	Initialize VPU registers	82
VPU.Set	Modify VPU registers	83
VPU.view	Display ALTIVEC register window	84

General Commands Reference Guide W

General Commands Reference Guide W	(general_ref_w.pdf)	1

General Commands Reference Guide X

General Commands Reference Guide X	(general_ref_x.pdf)	1

General Commands Reference Guide Y

General Commands Reference Guide Y	(general_ref_y.pdf)	1
sYmbol		3

General Commands Reference Guide Z

General Commands Reference Guide Z	(general_ref_z.pdf)	1

Source Level Debugging

Application Note C++ Debugging

Application Note C++ Debugging	(app_cpp_debugging.pdf)	1

Sample Code used by This Application Note	3
Gathering Information of Objects	4
Display Options Dedicated to Objects	4
Most Derived Class	6
Lifetime of the “this” Pointer	6
Class Conversions	7
Gathering Information of Classes	8
C++ Overloading and Symbol Mangling	9
Demangling	9
Ambiguous Symbols	9
Other Command Line Hints	10
ELF Loader Options	11
Debugging in C++ Code	12
Target Order / Source Order	12
Breakpoints	13

Multicore Debugging

Application Note for iAMP Debugging

Application Note for iAMP Debugging	(app_iamp.pdf)	1
SMP, iAMP or AMP?		3
iAMP Setup		7
Example iAMP Setup		10

FLASH Programming

Onchip/NOR FLASH Programming User's Guide

Onchip/NOR FLASH Programming User's Guide	(norflash.pdf)	1
Introduction		5
Standard Approach		6
On-chip FLASH		6
Off-chip FLASH Devices Supporting CFI		11
Programming Commands		21
FLASH.ReProgram Command (Target-controlled)		21
FLASH.ReProgram Command (TRACE32 Tool-based)		27

FLASH.Erase / FLASH.Program Command	28
The FLASH.AUTO Command	31
Unlocking Command	40
DualPort FLASH Programming	42
Benefits	42
Preconditions	42
Usage	43
Special Features for Onchip FLASHs	45
OTP Sector Programming	45
Mirrored FLASH Addresses	47
FLASH.Create Command	49
FLASH.TARGET Command	54
FLASH.CLock Command	55
FLASH.CHANGETYPE Command	56
FLASH.UNSECUREerase Command	57
FLASH Declaration in Detail	58
Further Applications for FLASH Declarations Using CFI	58
Declarations for not CFI-conform FLASH Devices	69
TRACE32 Tool-based vs. Target-controlled FLASH Programming	75
Maintaining the Declared FLASH Devices	87
List of Supported FLASH Devices	87
FLASH Programming via Boundary Scan	88
Boundary scan chain configuration	88
FLASH interface definition	89
FLASH Programming	90
Full Example	92
FAQ	94
Further Information	94

NAND FLASH Programming User's Guide

NAND FLASH Programming User's Guide(nandflash.pdf)	1
Introduction	5
How This Manual is Organized	5
Related Documents	6
Contacting Support	6
List of Abbreviations	8
Background Information	8
What is a NAND Flash Device?	8
About Blocks, Pages, Main Area, and Spare Area	9
About Bad Block Markers	10

About NAND Flash Controllers	11
Standard Approach	12
Identifying and Running Scripts for NAND Flash Programming	12
If There Is No Script	14
Scripts for NAND Flash Programming	15
Establishing Communication between Debugger and Target CPU	17
Configuring the NAND Flash Controller	18
Resetting Default Values	20
Identifying the Type of NAND Flash Controller	21
Informing TRACE32 about the NAND Flash Register Addresses	23
Informing TRACE32 about the NAND Flash Programming Algorithm	25
Checking the Identification from the NAND Flash Device	34
Erasing the NAND Flash Device	35
Programming the NAND Flash Device	36
Other Useful Commands (NAND)	39
Full Examples: Generic NAND Flash Programming	51
Full Example: CPU-Specific NAND Flash Programming	55
About OneNAND Flash Devices	56
Scripts for OneNAND Flash Devices	57
Establishing Communication between Debugger and Target CPU	59
Configuring the OneNAND Flash Bus	59
Resetting Default Values	60
Informing TRACE32 about the OneNAND Flash Address	60
Informing TRACE32 about the OneNAND Flash Programming Algorithm	61
Checking the Identification from the OneNAND Flash Device	65
Erasing the OneNAND Flash Device	66
Programming the OneNAND Flash Device	67
Other Useful Commands (OneNAND)	69
Full Example	79
Appendix A: ECC (Error Correction Code)	80
How to Generate ECC and to Detect Error	80
Appendix B: Spare Area Schemes	84
Linux MTD NAND Driver Default Spare Area Schemes	84
SAMSUNG Standard Spare Area Schemes	86

Serial FLASH Programming User's Guide

Serial FLASH Programming User's Guide	(serialflash.pdf) 1
Introduction	4
How This Manual is Organized	4
Related Documents	4

Contacting Support	5
List of Abbreviations	7
Background Knowledge	8
What is a Serial Flash Device?	8
About SPI Interface Controllers in Serial Flash Memories	8
About Blocks and Pages	9
File Name Convention for Serial Flash Drivers	10
Standard Approach	11
Identifying and Running Scripts for Serial Flash Programming	11
If There Is No Script	13
Scripts for SPI Controllers	14
Establishing Communication between Debugger and Target CPU	16
Configuring the SPI Controller	17
Resetting Default Values	18
Informing TRACE32 about the Serial Flash Register Addresses (SPI)	18
Informing TRACE32 about the Serial Flash Programming Algorithm	19
FLASHFILE Declaration Examples	24
Checking the Identification from the Serial Flash Device	26
Erasing the Serial Flash Device	27
Programming the Serial Flash Device	28
Other Useful Commands	32
Full Examples	34
FLASH Programming via Boundary Scan	37
Example 1 for the SPI Protocol	37
Example 2 for the I2C Protocol	39

eMMC FLASH Programming User's Guide

eMMC FLASH Programming User's Guide	(emmcflash.pdf)	1
Introduction		4
How This Manual is Organized		4
Related Documents		4
Contacting Support		5
List of Abbreviations		7
Background Knowledge		8
What is an eMMC Flash Device?		8
About Blocks and Pages		8
About eMMC Interface Controllers in eMMC Flash Memories		9
Standard Approach		10
Identifying and Running Scripts for eMMC Flash Programming		10
If There Is No Script		12

Scripts for eMMC Controllers	13
Establishing Communication between Debugger and Target CPU	14
Configuring the eMMC Controller	15
Resetting Default Values	16
Informing TRACE32 about the eMMC Controller Address	16
Informing TRACE32 about the eMMC Flash Programming Algorithm	16
FLASHFILE Declaration Examples	20
Checking the Identification from the eMMC Flash Device	22
Erasing the eMMC Flash Device	23
Programming the eMMC Flash Device	23
Other Useful Commands	27
Full Examples	29
FLASH Programming via Boundary Scan	32

Application Notes for FLASH

How to Write your own FLASH Algorithm

How to Write your own FLASH Algorithm (flash_app_own_algorithm.pdf)	1
FLASH Programming	3
Target Controlled Flash Programming	3

Tips to Solve NOR FLASH Programming Problems

Tips to Solve NOR FLASH Programming Problems(flash_diagnosis.pdf)	1
Introduction	4
General Recommendation	6
TRACE32 Error Messages	7
TRACE32 Tool-based Programming	8
Sources of Errors	9
General Course of Action in the Case of Problems (Tool-based)	10
Target-controlled Programming	34
Sources of Errors	35
General Course of Action in the Case of Problems (Target-controlled)	36
Errors Caused by Wrong Usage of the TRACE32 Commands	47
Unintentional Erasing of the Complete FLASH Device	47
Multiple FLASH Devices on one Target	48
FAQ	48
Appendix A	49
Conversion of the Tool-based to Target-controlled FLASH Programming	49

Checking the Bus Configuration	50
Main Difference between Intel and AMD/Spansion FLASH Devices	51
Intel FLASH Devices	52
AMD/Spansion FLASH Devices	56
Appendix B	68
FLASH Width BYTE	68
FLASH Width WORD	69
FLASH Width LONG	70
FLASH Width QUAD	72

How to Write your own FLASHFILE Algorithm

How to Write your own FLASHFILE Algorithm (flashfile_app_own_algorithm.pdf)	1
FLASHFILE Programming	3
Target Controlled Flashfile Programming	3

JTAG

Application Note JTAG Interface

Application Note JTAG Interface (app_jtag_interface.pdf)	1
History	3
Introduction	4
Related Documents	4
Debugging a JTAG Session	5
JTAG Basics	7
Main Concept	8
JTAG Implementation	15
Single TAP Controller	15
Multiple TAP Controllers	16
Custom JTAG Access	28
Overview	29
Remote API	31
Command Line Control	41

Boundary Scan User's Guide

Boundary Scan User's Guide (boundary_scan.pdf) 1

Introduction 4

 Intended Audience 4

 How This Manual is Organized 5

 Related Documents 5

List of Abbreviations 5

What to know about Boundary Scan 6

Configuration of the Boundary Scan Chain 8

 Configure Boundary Scan Engine 8

 Loading the BSDL Files 10

 Initialization of the Boundary Scan Chain 11

 Check the Configuration 12

General Operation 13

 Basic Mode of Operation 13

 Preparation of the Boundary Scan Chain 13

 Execution of the Boundary Scan Commands 17

 Working with the GUI 18

Interactive Board Test 26

 Configure Run Mode 27

 Execute Level and Connection Tests 30

Automated Board Test 33

 Prepare Boundary Scan Chain 34

 Run Tests 35

 Full Example 36

Test of Non-Boundary Scan Devices 38

Special Tests 42

 Boundary Scan Oscilloscope 42

 Other Instructions and Data Registers 43

Tips and Tricks 45

TRACE32 Lua Library

TRACE32 Lua Library	(lua_library.pdf)	1
TRACE32 Lua Library		3
Functions for JTAG Access		4
Input Output Functions		6
Example		7

Intel® DCI [Direct Connect Interface]

Debugging via Intel® DCI User´s Guide

Debugging via Intel® DCI User´s Guide	(dci_intel_user.pdf)	1
Introduction		4
4-wire DCI OOB		4
DCI OOB Hardware		6
DCI DbC		7
Target System Requirements		8
Related Documents		8
Start a TRACE32 Session using Intel® DCI		9
Prepare Your Target		9
Connecting to an Intel® SoC using DCI OOB		9
Connecting to an Intel® Client or Server System using DCI OOB		10
Connecting to an Intel® SoC using DCI DbC		11
Connecting to an Intel® Client or Server System using DCI DbC		12
Troubleshooting		13
DCI error: no response to connect pattern		13
Could not stop the target		13
Target Power Fail		13
Intel® DCI Specific Commands		14
DCI	Commands to configure the Intel® DCI trace handler	14
DCI.DESTination	Set trace destination	14
DCI.ON	Enable trace handler	14
DCI.OFF	Disable trace handler	15
SYStem.DCI	Intel® DCI specific SYStem commands	16
SYStem.DCI.Bridge	Select DCI bridge	16
SYStem.DCI.BssbClock	Configure DCI OOB clock rate	16
SYStem.DCI.CKDIrouting	Routing of the CK and DI signals	17

SYStem.DCI.DisCONnect	Force DCI disconnect	17
SYStem.DCI.DORouting	Routing of the DO signals	18
SYStem.DCI.PortPower	Configure VBUS	19
SYStem.DCI.TimeOut	Configure timeouts of internal operations	20
Intel® DCI Specific Functions		21
In This Section		21
SYStem.DCI.Bridge()	Currently selected DCI bridge	21
SYStem.DCI.BssbClock()	Currently selected DCI OOB clock	21
SYStem.DCI.TIMEOUT()	Timeouts of internal operations	22

Peripheral Files

Peripheral Files Programming

Peripheral Files Programming	(per_prog.pdf)	1
History		7
Introduction		7
Passing Arguments		8
Memory Classes		11
Comma-Separated-Values (CSV) File Format for *.per Files		12
Manual Peripheral File Generation		17
GROUP Commands		20
SGROUP Commands		25
Other Top Level Commands		34
Commands within GROUPs		58
Automated Peripheral File Generation		81
Graphical User Interface		81
Rules file		81
Error Messages		145
Functions		155

Converter SPIRIT XML to PER Commands

Converter SPIRIT XML to PER Commands	(converter_spiritxml.pdf)	1
Introduction		3
Main view		4
Component properties		7
Start processing		9

Application Note for Complex Trigger Language

Application Note for Complex Trigger Language	(app_ctl.pdf)	1
History		5
Introduction		6
Basic Structure of CTL Programs		7
Complex Statements		8
Agents		9
State Machines		12
TRACE32 Commands Using CTL Programs		14
CTL Onchip Triggers Logic		14
CTL for Trace Find		14
CTL Streaming Trace Trigger		15
CTL for Onchip Triggers Logic		16
CTL for TriCore MCDS		17
CTL for Arm ETM		31
Examples for CTL Trace Find		37
Use case 1: Checking Variable Access		37
Use case 2: Checking Timing Constraints - Address Duration		39
Use case 3: Checking Timing Constraints - Address Distance		41
Keyword Reference: CTL Conditions/Triggers		43
BREAKPOINT	ABCDE breakpoint	43
BusTrigger	Incoming trigger signal	43
BMC	Benchmark counter event	43
COUNT	Trigger on event counter	43
CLOCKS	Trigger on clock cycles counter	44
CTM	Cross trigger	44
EXTIN	External input	44
FALSE	Never condition	45
FLAG	Flag status	45
MACHINE	Machine comparator	45
Program	Program access comparator	46
ProgramFail	Conditional instruction execution	46
ProgramPass	Conditional instruction execution	47
Read	Read access	47
ReadWrite	Read or write access	48
SingleShot	Single shot comparators	49
SingleShot.Program	Single shot program execution	49
SingleShot.ProgramFail	Single shot conditional execution	49

SingleShot.ProgramPass	Single shot conditional execution	50
SingleShot.Read	Single shot read access	50
SingleShot.ReadWrite	Single shot read or write access	51
SingleShot.Write	Single shot write access	51
NoSingleShot	Non single shot comparators	52
NoSingleShot.Program	Non single shot program execution	52
NoSingleShot.ProgramFail	Non single shot conditional execution	52
NoSingleShot.ProgramPass	Non single shot conditional execution	53
NoSingleShot.Read	Non single shot read access	53
NoSingleShot.ReadWrite	Non single shot read or write access	53
NoSingleShot.Write	Non single shot write access	54
STATE.LEAVE	Leave the state transition (edge sensitive)	55
STATE.ENTER	Enter the state transition (edge sensitive)	55
STATE.TRACEON	Active state of a TraceON action	56
TASK	Task comparator	56
TIME	Time counter comparator	56
TRUE	Always condition	57
Var	Specify HLL expressions	58
Var.Program	Flat function execution	58
Var.Read	Variable read access	58
Var.ReadWrite	Variable read or write access	59
Var.status	tbd.	59
Var.Write	Variable write access	59
Write	Write access	60
ZONE	Zone comparator	60
Keyword Reference: CTL Actions		61
Break	Stop the program execution	61
BusCLOCKS	tbd.	61
BusCount	tbd.	61
BusTIME	tbd.	61
BusTrigger	tbd.	62
CLEAR	Clear flag	62
CTM	Cross trigger	62
ENABLE	Enable counter	63
EVENT	Trace event	63
EXTOUT	External output	64
FOUND	Add the trace sample to the search items result	64
GOTO	Change active state	64
INCRement	Increment counter	65
RELOAD	Reload counter	65
SET	Set flag	66
Spot	Shortly stop the program execution	66
TraceData	Sample specified data event	67

TraceEnable	Enable the trace on the specified event	68
TraceOFF	Switch OFF the trace sampling	69
TraceON	Switch ON the trace sampling	70
TraceTIME	tbd.	70
TraceTrigger	Stop sampling to the trace buffer on specified event	70
CTL Programming Errors		71

Trace Application Notes

Trace Analysis

Application Note for Trace.DRAW

Application Note for Trace.DRAW	(app_trace_draw.pdf)	1
Introduction		3
Intended Audience		3
Prerequisites		3
Related Documents		3
Restrictions		4
Trace Licenses		4
Data Capture		5
Full Trace		5
Filtered Trace		8
Trace Searching		9
Locate all Writes to a Variable		10
Locate Dedicated Reads to a Variable		11
Monitoring a Variable Over Time		12
Static Analysis		12
Variable Graphs		15
Advanced Navigation in Trace.DRAW		18
Mouse Support		18
Synchronizing Windows		20
Synchronizing Between TRACE32 Instances		23
Exporting Data		24
Appendix: Trace.DRAW Draw Options		26

Application Note for Trace.Find

Application Note for Trace.Find	(app_trace_find.pdf)	1

The Trace Find Dialog	4
Push Buttons	6
Cycle Tab (default)	9
Data Field	12
Expert Tab	14
Group Tab	21
Changes Tab	23
Signal Tab	31
Trace Find Commands	33
Overview	33
Combining Search Items	33
Record Numbers and Record Ranges	34
Trace Item with Specified Value	35
Format the Result	43
Related TRACE32 Functions	44
Convert Setting in Trace Find Dialog to a TRACE32 Command	45
Use Cases	46
Find Task Switches	46
Trace.Find Keyword Reference	49
APPEAR	Match when condition becomes a match 50
AT	Combine with condition at other record 50
CHANGE	Match when selected item changed 50
NOT	Negate condition 51
OR	OR condition 51
WITHIN	Time restriction for condition 51

Application Note for Trace.List

Application Note for Trace.List	(app_trace_list.pdf)	1
Overview		4
List items		4
Record items		5
Pre-defined Item Groups		5
Item reference: List items		6
List.Asm	Show disassembled mnemonics	6
List.ADDRESS	Show program address	7
List.CODE	Show program opcode	7
List.Comment	Show comments to disassembled mnemonics	7
List.DIAG	Disassembler-related diagnostic information	8
List.Dummy	Show dummy cycles	8
List.EXEC	Execution information about conditional branches	9

List.Hll	Show source code	9
List.Label	Show associated label	9
List.Mix	Show disassembled mnemonics and HLL code	10
List.NoDummy	Suppress the display of dummy cycles	10
List.NoFetch	Suppress the display of program fetches	10
List.SOURCE	Display associated source file name	11
List.SOURCEFILE	Display associated source file path and name	11
List.sYmbol	Show debug info for every linear program block	11
List.TASK	Display OS Awareness information	12
List.Tlme	Display time information for HLL lines	12
Item reference: Record items		13
Address	Start address contained in trace record	13
CPU	Set of items Run, Address, CYcle, Data and sYmbol	13
CYcle	Show bus cycle type of trace record	13
Run	Core and execution information	14
SIGNALS	Trace-port signals	15
sYmbol	Debug symbol of start address in trace record	15
Item reference: CLOCKS items		16
CLOCKS.Back	Number of clocks relative time to previous record	16
CLOCKS.Fore	Number of clocks relative time to next record	16
CLOCKS.REF	Number of clocks relative to reference point	17
CLOCKS.Trigger	Number of clocks relative to trigger point	17
CLOCKS.Zero	Number of clocks relative to global zero point	17
Item reference: Data items		18
Data.any	Data value stored in trace record (auto width)	18
Data.B<x>	Data value stored in trace record (byte lanes)	18
Data.L<x>	Data value stored in trace record (long lanes)	18
Data.Oct	Data value stored in trace record (oct-word)	18
Data.Q<x>	Data value stored in trace record (oct-word)	19
Data.T<x>	Data value stored in trace record (tbyte)	19
Data.W<x>	Data value stored in trace record (word)	19
Data.sYmbol	Debug symbol associated to data value	19
Item reference: ENERGY items		20
ENERGY.Abs	Energy consumption since start of trace data	20
ENERGY.Back	Energy consumption since previous record	20
ENERGY.Fore	Energy consumption until next record	20
ENERGY.REF	Energy consumption since reference record	20
ENERGY.Trigger	Energy consumption since trigger point	20
ENERGY.Zero	Energy consumption since global zero point	20
Item reference: Tlme items		22
Tlme.AddressBack	Time relative to previous occurrence of address	22
Tlme.AddressFore	Time relative to next address	23

Tlme.Back	Time relative to previous record	23
Tlme.Fore	Time relative to next record	23
Tlme.FUNC	Time spent to execute a function	23
Tlme.FUNCEX	Time spent outside the current function	23
Tlme.FUNCIN	Time spent in code of function	24
Tlme.MARK<x>BACK	Time relative back to the last marker	24
Tlme.MARK<x>FORW	Time relative forward to the next marker	24
Tlme.REF	Time relative to reference point	24
Tlme.Trigger	Time relative to trigger point	24
Tlme.Zero	Time relative to global reference	25

Application Note for eMMC Analysis

Application Note for eMMC Analysis	(app_analyze_emmc.pdf)	1
History		3
Introduction		4
TRACE32-based eMMC Access Log Solution		5
Implementation Example for Linux OS		8
Comparison with the Software Method ftrace		11
Conclusion		13
References		14
Appendix A: Source Code Example		15
Appendix B: Time Details		18

Application Note Profiling on AUTOSAR CP with ARTI

Application Note Profiling on AUTOSAR CP with ARTI	(app_autosar_cp_arti.pdf)	1
History		4
About this manual		5
Introduction		5
Related Documentation		6
Using ARTI Hooks		7
Hook Macros		7
Instrumentation		8
Object Detection		13
Timing Parameters		14
Overview of TRACE32 Command Structure		16
TASK.ARTI		16

TASK.ORTI	16
TASK.List	17
Trace.List	18
Trace.Chart	18
Trace.ProfileChart	19
Trace.STATistic	19
Trace.PROfileSTATistic	20
DURation Analysis	20
DIStance Analysis	21
SMP Options	21
GROUP	22
BMC	23
Trace.EXPORT	23
Task Runtime Analysis	24
Trace.Chart.TASKState	25
Trace.STATistic.TASKState	26
Trace.STATistic.TASKStateDURation	27
Runnable Runtime Analysis	28
Trace.Chart.Runnable	28
Trace.STATistic.RUNNABLE	29
Trace.STATistic.RUNNABLEDURation <runnablestart>	29
ISR2 Runtime Analysis	30
Trace.Chart.TASKINTR	30
Trace.STATistic.TASKINTR	31
Trace.Chart.TASKORINTRState	31
Trace.STATistic.TASKORINTRState	31
Interrupt Runtime Analysis	32
Spinlock Analysis	33
CPU Load Measurement	34
Grouping the Idle Tasks	34
CPU Load Overview	35
CPU Load in Time Slots	35
CPU Load by Benchmark Counters	36
Jitter Measurement	37
Jitter on Tasks	37
Jitter on Runnables	38
Export	40
CSV Export	40
Trace.EXPORT.TASKEVENTS (deprecated)	40
Trace.EXPORT.ARTI	40
Trace.EXPORT.MDF	41

Software Traces

Application Note for FDX

Application Note for FDX	(app_fdx.pdf)	1
General Function		3
Restrictions		3
Related Tutorials		4
Contacting Support		4
General Data Transfer		6
Target Application Interface for General Data Transfer		6
Host Application Interface for General Data Transfer		8
Configuration of TRACE32 for General Data Transfer		10
FDX Software Trace		12
Target Application Interface for FDX Trace		12
Configuration of TRACE32 for FDX Trace		14

Application Note for the SNOOPer Trace

Application Note for the SNOOPer Trace	(app_snooper.pdf)	1
History		3
Introduction		4
SNOOPer Trace Configuration		6
Sampling the Memory		8
Logging a Single Variable		10
Logging only Data Changes		12
Logging Multiple Variables		13
Display the SNOOPer Trace Results		14
SNOOPer Trace Trigger		19
Sampling the Program Counter		20
Setup		21
Display Options		21
Sampling the Program Counter and the Current Task		24
Data Sampling via Debug Communication Channel		27
Sampling Benchmark Counters		29
Sampling ETM Counters		33
Save and Load		37

Application Note for the LOGGER Trace	(app_logger.pdf)	1
History		3
Introduction		3
Related Tutorials		3
The LOGGER Trace Format		4
LOGGER Description Block		4
LOGGER Trace Records		5
LOGGER Target Application		7
LOGGER Functions		7
LOGGER Macros		11
LOGGER Trace Configuration		12
Display of LOGGER Trace Contents		15
List of Recorded Samples		16
Graphical Display of LOGGER Trace Results		18
Using the LOGGER for Task Switch Trace		19
LOGGER Trace Trigger		20

Code Coverage

Application Note for t32cast

Application Note for t32cast	(app_t32cast.pdf)	1
History		3
Introduction		4
Intended Audience		4
Prerequisites		5
Related Documents		5
Restrictions		5
Installation		6
System Requirements		6
License Requirements		6
Installing t32cast		6
Command Line Parameters of t32cast		7
Common Options for All Subcommands		8
Additional Options for ECA Subcommands		8
Additional Options for INSTRUMENT Subcommand		9

Application Note for Trace-Based Code Coverage

Application Note for Trace-Based Code Coverage	1
History	6
Intended Audience	7
Introduction	8
Supported Code Coverage Metrics	8
Code Coverage and Certification	9
Trace-Based Code Coverage	10
Test Variants	13
Merge Results and Generate Report	13
MC/DC, Condition and Decision Coverage	14
Multiple Code Coverage Modes	14
Evaluation of Switch Case Statements	19
Code Coverage Workflow	20
General Workflow	20
Workflow for the Individual Code Coverage Metrics	21
Build Process	29
Recommendations for the Build Toolchain	29
Build Process Statement Coverage	29
Build Process Function and ocb Decision Coverage	30
Build Process Call Coverage	31
Build Process MC/DC, Condition and Decision Coverage	32
Trace Data Collection Overview	42
TRACE32 Tool Configurations	42
Choose the Appropriate Trace Data Collection Variant	43
Preconditions	45
SMP Multicore Systems	47
Steps in Preparation for Trace Data Collection	48
Notes on the Individual Test Variants	48
Preparation for Function, Object Code, ocb Decision Coverage	49
Preparation for Statement Coverage	50
Preparation for Call Coverage	51
Preparation for MC/DC, Condition and Decision Coverage	52
Trace Data Collection	56
Incremental Code Coverage	56
Incremental Code Coverage in STREAM Mode	59
RTS Mode Code Coverage	63
SPY Mode Code Coverage	69

Code Coverage with Virtual Targets	75
ART Mode Code Coverage	77
Code Coverage Analysis	80
Code Coverage Tags	80
Object Code Coverage Evaluation	81
Statement Coverage Evaluation	86
Full Decision Coverage Evaluation	90
Object Code Based (ocb) Decision Coverage Evaluation	96
Condition Coverage Evaluation	103
Modified Condition/Decision Coverage (MC/DC) Evaluation	109
Function Coverage Evaluation	115
Call Coverage Evaluation	119
Comment Your Results	126
TRACE32 Merge and Report Tool	128
Appendix A: TRACE32 Coverage Report Utility	130
Appendix B: Assemble Multiple Test Runs at Address Level	132
Save and Restore Code Coverage Measurement	132
Save and Restore Trace Recording	134
Appendix C: Assembler-Only Functions and Code Coverage	136
Object Code Coverage	136
Source Code Metrics	137
Appendix D: Data Coverage	139
Trace Data Collection	139
Evaluation	140
Appendix E: Trace Decoding in Detail	143
Trace Decoding for Static Applications	143
Trace Decoding for Applications Using a Rich OS	145
Appendix F: Coding Guidelines	147
Appendix G: Object Code Coverage Tags in Detail	150
Standard Tags	150
Tags for Arm-ETMv1/v3/v4 for Arm/Cortex Architecture	151
Appendix E: Data Coverage in Detail	153

AutoFocus User's Guide

AutoFocus User's Guide	(autofocus_user.pdf)	1
Introduction		4
Intended Audience		4

Prerequisites	4
Contacting Support	5
Installation	7
Hardware Installation	7
Software Installation	15
Recommendation for the Software Start	15
Recommendation for Power Down	15
Utilization	16
Preprocessor	16
Trace Port	20
FAQ	24
Diagnosis	25
Displaying Error Messages	25
Searching for Errors	26
Types of Trace Decoder Errors	28
Trace Test Failure Messages	29
Access the Diagnosis Tool	30
Diagnosis Check List	31
Choose the Best Termination PCB Type	51
How to understand A.ShowFocusEye and A.ShowFocusClockEye	52
Recommendations for Target Board Design	55
Pin Remapping	57
Functional Description - Pin Remapping	57
Pinouts	59
Technical Data	61
AutoFocus Preprocessor Hardware Versions	61
Electrical Specification for AutoFocus II Preprocessors	64
Dimensions	67
Connector Layout for Arm	76

PowerTrace Serial User's Guide

PowerTrace Serial User's Guide	(serialtrace_user.pdf)	1
Introduction		5
Intended Audience		5
Prerequisites		5
Contacting Support		6
Installation		8
Hardware Installation		8
Software Installation		32

Recommendation for the Software Start	32
Recommendation for Power Down	32
Trace Port Utilization	33
HSSTP-based Trace Port	33
AGBT-based Trace Port	38
Serial NEXUS-based Trace Port	38
PCIe-based Trace Port	38
FAQ	40
Diagnosis	41
Device LED Codes	41
Displaying Error Messages	42
Searching for Errors	43
Types of Trace Decoder Errors	45
Trace Test Failure Messages	46
Diagnosis Check List	47
Recommendations for Target Board Design	52
Technical Data	53
PowerTrace Serial Accessory Sets	53
Electrical Specification for PowerTrace Serial	56
Dimensions	58
Connector Layout	59

System Trace User's Guide

System Trace User's Guide	(trace_stm.pdf)	1
History		7
Introduction		8
Installation		10
Software Installation		10
Hardware Installation		10
Utilization of the STM		12
Start-up Script		12
STM Component - Configuration of TRACE32		15
SYStem.CONFIG.STM	Inform TRACE32 about STM component	15
SYStem.CONFIG.STM.Mode	STPv2 timestamp format	17
SYStem.CONFIG.STM.Type	Configure STM type in TRACE32	18
STM Component - General Target Configuration		19
STM	Configure STM component on target	19
STM.FilterMasters	Display specified masters only	20

STM.FilterChannels	Display specified channels only	20
STM.Init	Initialize trace hardware	21
STM.OFF	Switch STM off	21
STM.ON	Switch STM on	21
STM.PortEndianness	Select port endianness	22
STM.PortMode	Select STM port modes	23
STM.PortRoute	Select output of STP data	24
STM.PortSize	Port size in bits	25
STM.PrintfTraceFormat	Define format of printftrace style messages	26
STM.Register	Display STM register	26
STM.RESet	Reset STM settings	27
STM.SetMaster	Set master ID manually	27
STM.state	Display STM settings	28
STM.SyncTime	Trace decoder resync time	29
STM.SyncPeriod	Add synchronization packets	29
STM.TimeStamps	Enables timestamps	29
STM.TimeStampCLOCK	Configure debugger for STM timestamp clock	30
STM Component - TI specific Target Configuration		31
STM.HWMasters	Enable hardware masters for tracing	31
STM.IdleCount	Maximum idle packets	31
STM.IgnoreHeader	Ignore leading dword in printftrace message	32
STM.SWMasters	Enable software masters for tracing	32
STM.OCPAutoidle	Set OCP idle clock behavior	33
STM.PATTERN	Enable test pattern generator	33
STM.ChannelRepeat	Period of channel packet insertion	34
STM.MasterRepeat	Period of master packet insertion	34
STM Component - CoreSight specific Target Configuration		35
STM.DMArequests	DMA requests enable	35
STM.COMPression	Data compression enable	35
STM.EventMASK	Mask hardware event inputs	36
STM.PortMASK	Mask stimulus ports	36
STM.TraceID	Sets trace ID	37
STM.TracePriority	Set priority for STM manually	37
STM<trace> - Trace Data Analysis		38
STM<trace>	Command groups for STM<trace> recording and analysis	38
Overview STM<trace>		38
STMAAnalyzer	Analyze STM data recorded by TRACE32 PowerTrace	39
STMCAAnalyzer	Analyze STM data recorded by TRACE32 CombiProbe	40
STMHAnalyzer	Display and analyze STM data recorded by the host	41
STMLA	Display and analyze STM data from binary file	41
STMONchip	Display and analyze STM data stored on target memory	42
STMTrace	Method-independent display and analysis of STM trace data	43

PrintfTrace		45
PrintfTrace	Decoder for STP-based software messages	45
CMI Component		48
SYStem.CONFIG.CMI	Inform TRACE32 about CMI component	48
CMI	Configure CMI component on target	49
CMI.EnableMessage	Enables event or activity message generation	50
CMI.Mode	Set event or activity mode	51
CMI.OFF	Switch CMI off	51
CMI.ON	Switch CMI on	51
CMI.Register	Display the CMI register	52
CMI.RESet	Resets CMI settings to their defaults	52
CMI.SamplingWindow	Sampling window	53
CMI.SamplingWindow.CLock	Set sampling window ratio	53
CMI.SamplingWindow.Size	Set sampling window size	53
CMI.state	Display CMI settings	54
CMI Example		55
CMITrace	Display and analyze CMI trace data	56
CMN Component		58
SYStem.CONFIG.CMN	Inform TRACE32 about CMN component	58
CMN	Configure CMN component on target	60
CMN.CycleAccurate	Enable cycle counter information	61
CMN.EnhancedFilter	Set an individual filter on a CMN XP	62
CMN.Init	Initialize CMN on target	63
CMN.MPAM	MPAM format of CMN tracking messages	64
CMN.NodeID	Set global filter for xp target or source	65
CMN.OFF	Switch CMN trace off	66
CMN.ON	Switch CMN trace on	66
CMN Opcode	Set global filter for opcode	66
CMN.PortRoute	Select output of CMN data	67
CMN.Register	Display CMN register	67
CMN.RESet	Reset CMN settings	68
CMN.state	Display CMN settings	68
CMN.SyncPeriod	Set period of synchronisation packet	69
CMN.TimeStampCLOCK	Configure debugger for CMN timestamp clock	69
CMN.TimeStampPeriod	Set period of timestamp packet	69
CMN.TraceChannel	Set global filter for CMN channel	70
CMN.TraceID	Sets trace ID	70
CMN.TracePriority	Set priority for CMN manually	70
CMN.VERsion	CMN type	71
CMN Example		72
CMNTrace	Display and analyze CMN trace data	73
CPTracer Component		76

CPTracer	Configure CPTracer component on target	76
CPTracer.RESet	Resets CPT settings to their defaults	77
CPTracer.state	Display CPT settings	77
CPTracer.TraceID	Set ATB ID	79
CPTracer.<aggregator>.ON	Switch aggregator on	79
CPTracer.<aggregator>.OFF	Switch aggregator off	79
CPTracer.<aggregator>.SYNC	Sync period of aggregator	80
CPTracer.<aggregator>.<probe>.ADDRESSLOW	Lower filter address	80
CPTracer.<aggregator>.<probe>.ADDRESSHIGH	Upper filter address	81
CPTracer.<aggregator>.<probe>.CHannel	Filter by channel ID	81
CPTracer.<aggregator>.<probe>.DIRection	Filter by transfer direction	81
CPTracer.<aggregator>.<probe>.OPeration	Mode of operation	83
CPTracer.<aggregator>.<probe>.PERiod	Set period of sample window	83
CPTracer.<aggregator>.<probe>.RouteID	Filter by route ID	84
CPTracer Example		85
CPTracerTrace	Display and analyze CPT trace data	86
OCP Component		88
SYStem.CONFIG.OCP	Inform TRACE32 about OCP component	88
OCP	Configure OCP component on target	89
OCP.AutoIDLE	OCP-WP clocking strategy	90
OCP.DebugPort	Select target to be traced	90
OCP.OFF	Switch OCP off	90
OCP.ON	Switch OCP on	91
OCP.Register	Display OCP registers	91
OCP.RESet	Reset OCP settings to their defaults	91
OCP.state	Display OCP settings	92
OCP.TraceFilter	Set filter criteria	93
OCP.TraceFilter<x>.NAME	Name a filter	93
OCP.TraceFilter<x>.MCMd	Filters traffic by transaction type	93
OCP.TraceFilter<x>.Initiator	Filters traffic by transaction initiator	94
OCP.TraceFilter<x>.REQinfo	Filters traffic by transaction qualifier	94
OCP.TraceEnable	Filter OCP traffic by address range	95
OCP.TraceOFF	Stop tracing	95
OCP.TraceON	Start tracing	96
OCP.TriggerOut<x>	Generate trigger event	96
OCPTrace	Display and analyze OCP trace data	97
PMI Component		98
SYStem.CONFIG.PMI	Inform TRACE32 about PMI component	98
PMI	Configure PMI component on target	99
PMI.EnableMessage	Enables event message generation	100
PMI.OFF	Switch PMI off	100
PMI.ON	Switch PMI on	101
PMI.Register	Display the PMI registers	101

PMI.RESet	Resets PMI settings to their defaults	101
PMI.SamplingWindow	Sampling window	102
PMI.SamplingWindow.CLock	Set sampling window clock	102
PMI.SamplingWindow.Size	Set sampling window size	102
PMI.state	Display PMI settings	103
PMI Example		104
PMITrace	Display and analyze PMI trace data	105
StatCol Component (Statistics Collector)		107
SYStem.CONFIG.SC	Inform TRACE32 about StatCol component	107
StatCol	Configure StatCol component on target	108
StatCol.RESet	Resets all statistics collector settings to their default	109
StatCol.state	Display statistics collector settings	109
StatCol.<probe>.OFF	Switch probe off	110
StatCol.<probe>.ON	Switch probe on	110
StatCol.<probe>.REQuestEvenT	Select event detector	111
StatCol.<probe>.ReSPonseEvenT	Select event detector	112
StatCol.<probe>.CollectTime	Set up collection period	112
StatCol.<probe>.Counter	Counter configuration	113
StatCol.<probe>.Counter <counter> ADDRMAX	Filter max address	113
StatCol.<probe>.Counter <counter> ADDRMIN	Filter min address	113
StatCol.<probe>.Counter <counter> ADDRn	Enable address filtering	114
StatCol.<probe>.Counter <counter> EventInfo	Select 'EventInfo' to count	114
StatCol.<probe>.Counter <counter> MAX	Set max threshold for events	115
StatCol.<probe>.Counter <counter> MIN	Set min threshold for events	115
StatCol.<probe>.Counter <counter> SElect	Set counter input	116
StatCol.<probe>.Counter <counter> Filter	Set filter criteria	117
StatCol.<probe>.Counter <counter> Filter <filter> MUX	Input port	118
StatCol.<probe>.Counter <counter> Filter <filter> OFF	Switch filter off	118
StatCol.<probe>.Counter <counter> Filter <filter> ON	Switch filter on	118
StatCol.<probe>.Counter <counter> FunCTioN	Predefined settings	119
StatCol Example		124
StatColTrace	Display and analyze StatCol trace data	125
Generic Subcommands, Parameters, and Options		126
SYStem.CONFIG.<component>.<generic>		126
SYStem.CONFIG.<component>.Base	Base address of a component	127
SYStem.CONFIG.<component>.Name	Name of a component	127
SYStem.CONFIG.<component>.RESet	Reset of a component	127
SYStem.CONFIG.<component>.view	Display component settings	128
FAQ		128

Bootloader Awareness Manual coreboot

Bootloader Awareness Manual coreboot	(boot_coreboot.pdf)	1
History		3
Overview		4
Brief Overview of Documents for New Users		4
Supported Versions		5
Configuration		6
x86 32-Bit		6
x64 64-Bit		6
Features		7
Display of coreboot Resources		7
Coreboot Specific Menu		8
Coreboot Commands		9
EXTension.INFO	Display coreboot build information	9
EXTension.BuildCONF	Display coreboot build configuration	9
EXTension.LOG	Display coreboot log	10
EXTension.CBFS	Display coreboot file system	11
EXTension.TimeStamps	Display timestamps	12
EXTension.MemoryMAP	Display memory mapping	13
EXTension.CBTABLE	Display contents of coreboot tables	14

Bootloader Awareness Manual GRUB

Bootloader Awareness Manual GRUB	(boot_grub.pdf)	1
History		3
Overview		3
Brief Overview of Documents for New Users		4
Supported Architectures		4
Configuration		5
Features		6
Display of GRUB Resources		6
GRUB Specific Menu		7
Symbol Autoloader		8
GRUB Commands		10
EXTension.DEVices	Display GRUB boot devices	10
EXTension.EnvVar	Display GRUB environment variables	10

EXTension.FileSystem	Display GRUB boot file system	11
EXTension.MODule	Display GRUB loaded modules	11
EXTension.sYmbol	Symbol management	12
EXTension.sYmbol.DELeMod	Delete module debug symbols	12
EXTension.sYmbol.LOADMod	Load module debug symbols	12
EXTension.sYmbol.ModPATH	Set GRUB module directory path	13
GRUB PRACTICE Functions		14
TASK.MOD.MAGIC()	Magic of GRUB module	14
TASK.MOD.NAME()	Name of GRUB module	14
TASK.MOD.BASE()	Base address of GRUB module	14
TASK.MOD.INIT()	Init address of GRUB module	15
TASK.MOD.SIZE()	Size of GRUB module	15
TASK.Y.MODP(modpath)	Get path for GRUB modules	15

UEFI Awareness Manuals

UEFI Awareness Manual BLDK

UEFI Awareness Manual BLDK	(uefi_bldk.pdf)	1
History		4
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
x86 32-Bit		6
x64 64-Bit		6
Hooks & Internals in Intel BLDK		7
Features		8
Display of UEFI Resources		8
Symbol Autoloader		9
Intel BLDK Specific Menu		12
Debugging UEFI Phases of Intel BLDK		13
Debugging from Reset Vector		13
SEC Phase		13
PEI Phase		13
DXE Phase		14
BDS Phase		15
Intel BLDK Commands		16
EXTension.ConfigTab	Display DXE configuration table	16
EXTension.DXEDRiVer	Display loaded DXE drivers	16

EXTension.DXEModule	Display DXE modules	17
EXTension.FV	Display firmware volumes	18
EXTension.HOB	Display HOBs	19
EXTension.Option	Set awareness options	19
EXTension.PEIModule	Display PEI modules	20
EXTension.PEISvc	Display PEI services	21
EXTension.POST	Display POST code	21
EXTension.PROTOcol	Display installed protocols	22
EXTension.UCode	Display microcodes	22
Intel BLDK PRACTICE Functions		23
EXT.DXEDRV.ENTRY()	Entry address for DXE driver	23
EXT.DXEDRV.MAGIC()	DXE driver magic number	23
EXT.DXEDRV.PATH()	Build path for DXE driver	23
EXT.DXEFILE.MACHINE()	Machine type for DXE module	24
EXT.DXEFILE.PATH()	Build path for DXE module	24
EXT.PEIM.ENTRY()	Entry address for PEI module	24
EXT.PEIM.MAGIC()	Magic of PEI module	25
EXT.PEIM.PATH()	Build path for PEI module	25

UEFI Awareness Manual H2O

UEFI Awareness Manual H2O	(uefi_h2o.pdf)	1
History		4
Overview		5
Brief Overview of Documents for New Users		5
Supported Versions		6
Configuration		7
x86/Atom 32-Bit		7
x64/Atom 64-Bit		7
Hooks & Internals in InsydeH2O		8
Features		9
Display of UEFI Resources		9
Symbol Autoloader		10
InsydeH2O Specific Menu		13
Debugging UEFI Phases of InsydeH2O		15
Debugging from Reset Vector		15
SEC Phase		15
PEI Phase		16
DXE Phase		17
BDS Phase		17
InsydeH2O Commands		18

EXTension.ConfigTab	Display DXE configuration table	18
EXTension.DXEDRiVer	Display loaded DXE drivers	18
EXTension.DXEModule	Display DXE modules	19
EXTension.FV	Display firmware volumes	20
EXTension.HOB	Display HOBs	21
EXTension.Option	Set awareness options	22
EXTension.PEIModule	Display PEI modules	22
EXTension.PEISvc	Display PEI services	23
EXTension.POST	Display POST code	23
EXTension.PROTOcol	Display installed protocols	24
EXTension.UCode	Display microcodes	24
InsydeH2O PRACTICE Functions		25
EXT.DXEDRV.ENTRY()	Entry address for DXE driver	25
EXT.DXEDRV.MAGIC()	Magic of DXE driver	25
EXT.DXEDRV.PATH()	Build path for DXE driver	25
EXT.DXEFILE.PATH()	Build path for DXE module	26
EXT.PEIM.ENTR()	Entry addres for PEI module	26
EXT.PEIM.MAGIC()	Magic of PEI module	26
EXT.PEIM.PATH()	Build path for PEI module	26

UEFI Awareness Manual TianoCore

UEFI Awareness Manual TianoCore	(uefi_tiano.pdf)	1
History		4
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
ARM 32-Bit		6
ARM 64-Bit		7
Hooks & Internals in TianoCore		7
Features		8
Display of UEFI Resources		8
Symbol Autoloader		9
TianoCore Specific Menu		12
Debugging UEFI Phases of TianoCore		13
Debugging from Reset Vector		13
SEC Phase		13
PEI Phase		13
DXE Phase		13
BDS Phase		14

TianoCore Commands	15
EXTension.ConfigTab	Display DXE configuration table 15
EXTension.DXEDRiVer	Display loaded DXE drivers 15
EXTension.DXEModule	Display DXE modules 16
EXTension.FV	Display firmware volumes 17
EXTension.HOB	Display HOBs 17
EXTension.Option	Set awareness options 18
EXTension.PEIModule	Display PEI modules 19
EXTension.PEISvc	Display PEI services 19
EXTension.POST	Display POST code 20
EXTension.PROTOCOL	Display installed protocols 20
TianoCore PRACTICE Functions	21
EXT.DXEDRV.ENTRY()	Entry address for DXE driver 21
EXT.DXEDRV.MAGIC()	Magic of DXE driver 21
EXT.DXEDRV.PATH()	Build path for DXE driver 21
EXT.DXEFILE.PATH()	Build path for DXE module 22
EXT.PEIM.ENTRY()	Entry address for PEI module 22
EXT.PEIM.MAGIC()	Magic of PEI module 22
EXT.PEIM.PATH()	Build path for PEI module 22

OS Awareness Manuals

OS Awareness Manual AMX

OS Awareness Manual AMX	(rtos_amx.pdf) 1
History	4
Overview	4
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Manual Configuration	6
Automatic Configuration	7
Quick Configuration Guide	7
Hooks & Internals of AMX	8
Features	9
Display of Kernel Resources	9
Task Stack Coverage	9
Task-Related Breakpoints	10
Task Context Display	11
Dynamic Task Performance Measurement	12

Task Runtime Statistics	13
Task State Analysis	14
Function Runtime Statistics	15
AMX specific Menu	17
AMX Commands	18
TASK.DBPool	Display buffer pools 18
TASK.DEVent	Display event groups 18
TASK.DEXChange	Display message exchanges 18
TASK.DMailBoX	Display mailboxes 19
TASK.DMPool	Display memory pools 20
TASK.DSEMaphore	Display semaphores 21
TASK.DSYSstem	Display system state 21
TASK.DTask	Display tasks 22
TASK.DTImEr	Display timers 23
AMX PRACTICE Functions	24
TASK.CONFIG()	OS Awareness configuration information 24

OS Awareness Manuals for ARTI

Application Note Profiling on AUTOSAR CP with ARTI

Application Note Profiling on AUTOSAR CP with ARTI(app_autosar_cp_arti.pdf)	1
History	4
About this manual	5
Introduction	5
Related Documentation	6
Using ARTI Hooks	7
Hook Macros	7
Instrumentation	8
Object Detection	13
Timing Parameters	14
Overview of TRACE32 Command Structure	16
TASK.ARTI	16
TASK.ORTI	16
TASK.List	17
Trace.List	18
Trace.Chart	18
Trace.ProfileChart	19
Trace.STATistic	19
Trace.PROfileSTATistic	20

DURation Analysis	20
DIStance Analysis	21
SMP Options	21
GROUP	22
BMC	23
Trace.EXPORT	23
Task Runtime Analysis	24
Trace.Chart.TASKState	25
Trace.STATistic.TASKState	26
Trace.STATistic.TASKStateDURation	27
Runnable Runtime Analysis	28
Trace.Chart.Runnable	28
Trace.STATistic.RUNNABLE	29
Trace.STATistic.RUNNABLEDURation <runnablestart>	29
ISR2 Runtime Analysis	30
Trace.Chart.TASKINTR	30
Trace.STATistic.TASKINTR	31
Trace.Chart.TASKORINTRState	31
Trace.STATistic.TASKORINTRState	31
Interrupt Runtime Analysis	32
Spinlock Analysis	33
CPU Load Measurement	34
Grouping the Idle Tasks	34
CPU Load Overview	35
CPU Load in Time Slots	35
CPU Load by Benchmark Counters	36
Jitter Measurement	37
Jitter on Tasks	37
Jitter on Runnables	38
Export	40
CSV Export	40
Trace.EXPORT.TASKEVENTS (deprecated)	40
Trace.EXPORT.ARTI	40
Trace.EXPORT.MDF	41
TIMEX	43

OS Awareness Manual Atomthreads

OS Awareness Manual Atomthreads	(rtos_atomthreads.pdf)	1
Overview		3

Terminology	3
Brief Overview of Documents for New Users	3
Supported Versions	4
Restrictions	4
Configuration	5
Quick Configuration Guide	6
Hooks & Internals in Atomthreads	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Task Context Display	9
Dynamic Task Performance Measurement	10
Task Runtime Statistics	11
Function Runtime Statistics	11
Atomthreads Specific Menu	13
Atomthreads Commands	14
TASK.MuTeX	Display mutexes 14
TASK.Queue	Display message queues 15
TASK.SEMaphore	Display semaphores 16
TASK.TaskList	Display tasks 17
TASK.TIMER	Display timers 18
Atomthreads PRACTICE Functions	19
TASK.CONFIG()	OS Awareness configuration information 19
TASK.STRUCT()	OS structure names 19
Appendix A: Context ID on Cortex-A Systems	20

OS Awareness Manual ARTX-166

OS Awareness Manual ARTX-166	(rtos_rtxartx166.pdf)	1
Overview		3
Brief Overview of Documents for New Users		4
Supported Versions		4
Configuration		5
Quick Configuration Guide		5
Hooks & Internals in ARTX-166		6
Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task-Related Breakpoints		8
Dynamic Task Performance Measurement		9

Task Runtime Statistics	9
Task State Analysis	10
Function Runtime Statistics	11
ARTX-166 Specific Menu	13
ARTX-166 Commands	14
TASK.Task	Display tasks 14
ARTX-166 PRACTICE Functions	15
TASK.CONFIG()	OS Awareness configuration information 15

OS Awareness Manual ChibiOS/RT

OS Awareness Manual ChibiOS/RT	(rtos_chibios.pdf)	1
Overview		3
Terminology		3
Brief Overview of Documents for New Users		4
Supported Versions		4
Configuration		5
Quick Configuration Guide		6
Hooks & Internals in ChibiOS		6
Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task-Related Breakpoints		8
Task Context Display		9
Dynamic Task Performance Measurement		10
Task Runtime Statistics		11
Function Runtime Statistics		12
ChibiOS specific Menu		14
ChibiOS Commands		15
TASK.CONDvar	Display condition variables	15
TASK.EVEnT	Display events	15
TASK.Heap	Display heaps	16
TASK.MailBox	Display mailboxes	16
TASK.MuTeX	Display mutexes	17
TASK.Pool	Display memory pools	17
TASK.Queue	Display queues	18
TASK.SEMaphore	Display semaphores	18
TASK.Thread	Display threads	19
TASK.VTimer	Display virtual timers	19
ChibiOS PRACTICE Functions		20
TASK.CONFIG()	OS Awareness configuration information	20

OS Awareness Manual Cmicro	(rtos_cmicro.pdf)	1
Overview		3
Brief Overview of Documents for New Users		3
Supported Versions		4
Configuration		5
Manual Configuration		5
Automatic Configuration		6
Hooks & Internals of SDT-Cmicro		6
Features		7
Display of Kernel Resources		7
Process Runtime Statistics		7
Function Runtime Statistics		8
SDT-Cmicro specific Menu		9
SDT-Cmicro Commands		10
TASK.PList	Display process instances	10
TASK.PType	Display process types	10
TASK.QList	Display signal queue	11
SDT-Cmicro PRACTICE Functions		12
TASK.CONFIG()	OS Awareness configuration information	12

OS Awareness Manual CMX

OS Awareness Manual CMX	(rtos_cmx.pdf)	1
Overview		3
Brief Overview of Documents for New Users		4
Supported Versions		4
Configuration		5
Quick Configuration Guide		5
Hooks & Internals in CMX		5
Features		6
CMXBug Terminal Emulation		6
Display of Kernel Resources		6
Task Stack Coverage		6
Task-Related Breakpoints		7
Dynamic Task Performance Measurement		8
Task Runtime Statistics		8
Task State Analysis		9
Function Runtime Statistics		10
CMX specific Menu		12

CMX Commands	13
TASK.DCyclic	Display cyclic timers 13
TASK.DMailbox	Display mailboxes 13
TASK.DQueue	Display queues 14
TASK.DRes	Display resources 14
TASK.DSema	Display semaphores 15
TASK.DTask	Display tasks 15
CMX PRACTICE Functions	16
TASK.CONFIG()	OS Awareness configuration information 16
TASK.STACK()	Stack information of a task 16

OS Awareness Manual DSP/BIOS

OS Awareness Manual DSP/BIOS(rtos_bios.pdf)	1
Overview	3
Brief Overview of Documents for New Users	4
Supported Versions	4
Configuration	5
Quick Configuration Guide	6
Hooks & Internals in DSP/BIOS	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Dynamic Task Performance Measurement	9
DSP/BIOS specific Menu	10
DSP/BIOS Commands	11
TASK.KerNeL	Display kernel information 11
TASK.LOG.DISable	Disable system log events 11
TASK.LOG.ENABLE	Enable system log events 11
TASK.LOG.View	Display logs 12
TASK.MailBoX	Display mailboxes 13
TASK.MEMory	Display memory segments 13
TASK.SEMaphore	Display semaphores 14
TASK.SWI	Display SWIs 14
TASK.TaSK	Display tasks 15
DSP/BIOS PRACTICE Functions	16

OS Awareness Manual eCos

OS Awareness Manual eCos(rtos_ecos.pdf)	1
History	3

Overview	3
Terminology	3
Brief Overview of Documents for New Users	4
Supported Versions	4
Configuration	5
Quick Configuration Guide	6
Hooks & Internals in eCos	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Task Context Display	9
SMP Support	10
Dynamic Task Performance Measurement	10
Task Runtime Statistics	11
Task State Analysis	12
Function Runtime Statistics	13
eCos specific Menu	15
eCos Commands	16
TASK.SCHEDuler	Display scheduler information 16
TASK.THRead	Display threads 16
eCos PRACTICE Functions	18
TASK.CONFIG()	OS Awareness configuration information 18

OS Awareness Manual embOS

OS Awareness Manual embOS	(rtos_embos.pdf) 1
Overview	4
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Quick Configuration Guide	6
Hooks & Internals in embOS	7
Debug Features	8
Display of Kernel Resources	8
Task Stack Coverage	8
Task-Related Breakpoints	9
Task Context Display	10
Dynamic Task Performance Measurement	11
embOS specific Menu	12
Trace Features	13

Task Runtime Statistics	13
Task State Analysis	14
Function Runtime Statistics	15
embOS specific Menu for Tracing	16
embOS Commands	17
TASK.CSema	Display 'CSemaphore' 17
TASK.EVenT	Display event object 17
TASK.MailBox	Display mailbox 18
TASK.MuTeX	Display mutex 18
TASK.Queue	Display queue 19
TASK.RWLock	Display RW locks 19
TASK.RSema	Display 'RSemaphore' 20
TASK.SEMaphore	Display semaphore 20
TASK.TaskList	Display tasks 21
TASK.TIMer	Display timer 21
embOS PRACTICE Functions	22
TASK.CONFIG()	OS Awareness configuration information 22

OS Awareness Manual FAMOS

OS Awareness Manual FAMOS	(rtos_famos.pdf) 1
Overview	3
Terminology	3
Brief Overview of Documents for New Users	4
Supported Versions	4
Configuration	5
Quick Configuration Guide	5
Hooks & Internals in FAMOS	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Task Context Display	9
Dynamic Task Performance Measurement	10
Task Runtime Statistics	11
Task State Analysis	12
Function Runtime Statistics	13
FAMOS Specific Menu	15
FAMOS Commands	16
TASK.Kernel	Display kernel state 16
TASK.MailBox	Display mailboxes 16
TASK.MailQueues	Display mailqueues 17

TASK.Semaphore	Display semaphores	17
TASK.Thread	Display threads	18
TASK.TIMER	Display timers	19
FAMOS PRACTICE Functions		20
TASK.CONFIG()	OS Awareness configuration information	20

OS Awareness Manual FreeRTOS

OS Awareness Manual FreeRTOS(rtos_freertos.pdf)		1
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Manual Configuration		6
Automatic Configuration		7
Quick Configuration Guide		7
Hooks & Internals in FreeRTOS		8
Features		9
Display of Kernel Resources		9
Task Stack Coverage		9
Task-Related Breakpoints		11
Task Context Display		12
SMP Support		13
Dynamic Task Performance Measurement		13
Task Runtime Statistics		14
Task State Analysis		15
Function Runtime Statistics		16
FreeRTOS specific Menu		18
FreeRTOS Commands		19
TASK.EvtGrp	Display event groups	19
TASK.MsgBuf	Display message buffers	19
TASK.Option	Set awareness options	20
TASK.Queue	Display queues	20
TASK.Semaphore	Display semaphores	21
TASK.StrBuf	Display stream buffers	21
TASK.TaskList	Display tasks	22
TASK.Timer	Display timers	23
FreeRTOS PRACTICE Functions		24
TASK.AVAIL()	Availability of FreeRTOS objects	24
TASK.CONFIG()	OS Awareness configuration information	24
TASK.STRUCT()	Structure names	25

Run Mode Debugging Manual Linux

Run Mode Debugging Manual Linux	(rtos_linux_run.pdf)	1
Debugging Modes for Embedded Linux		3
Run Mode Debugging with TRACE32 as GDB Front-end		3
Stop Mode Debugging		3
Integrated Run & Stop Mode Debugging via JTAG		4
Basic Concepts		5
Ethernet as Communication Interface to the gdbserver		5
DCC as Communication Interface to the t32server		5
The Space ID for Run Mode Debugging		6
Process Debugging		8
Switching between Run & Stop Mode Debugging		10
Commands for Run Mode Debugging		13
Breakpoint Conventions		14

OS Awareness Manual Linux

OS Awareness Manual Linux	(rtos_linux_stop.pdf)	1
History		5
Overview		6
Terminology		6
Brief Overview of Documents for New Users		6
Supported Versions		7
Configuration		8
Quick Configuration Guide		8
Hooks & Internals in Linux		8
Features		10
Display of Kernel Resources		10
Task-Related Breakpoints		10
Task Context Display		11
MMU Support		13
Symbol Autoloader		21
SMP Support		22
Dynamic Task Performance Measurement		23
Task Runtime Statistics		23
Process / thread switch support for ARM using context ID register:		24
Task State Analysis		24

Function Runtime Statistics	25
Linux Specific Menu	27
Debugging Linux Kernel and User Processes	28
Linux Kernel	29
User Processes	31
Kernel Modules	39
Trapping Segmentation Violation	42
Linux Commands	43
TASK.CHECK	Check awareness integrity 43
TASK.DMESG	Display the kernel ring buffer 43
TASK.DTask	Display tasks 44
TASK.DTB	Display the device tree blob 45
TASK.DTS	Display the device tree source 45
TASK.NET	Display network devices 45
TASK.FS	Display file system internals 46
TASK.MAPS	Display process maps 46
TASK.MMU.SCAN	Scan process MMU space 46
TASK.MODule	Display kernel modules 47
TASK.Option	Set awareness options 47
TASK.Process	Display processes 48
TASK.PS	Display “ps” output 49
TASK.sYmbol	Process/Module symbol management 50
TASK.sYmbol.DELeTe	Unload process symbols and MMU 50
TASK.sYmbol.DELeTeLib	Unload library symbols 51
TASK.sYmbol.DELeTeMod	Unload module symbols and MMU 51
TASK.sYmbol.LOAD	Load process symbols and MMU 52
TASK.sYmbol.LOADLib	Load library symbols 53
TASK.sYmbol.LOADMod	Load module symbols and MMU 53
TASK.sYmbol.Option	Set symbol management options 54
TASK.VMAINFO	Display vmallocated areas 57
TASK.Watch	Watch processes 58
TASK.Watch.ADD	Add process to watch list 58
TASK.Watch.DELeTe	Remove process from watch list 58
TASK.Watch.DISable	Disable watch system 59
TASK.Watch.DISableBP	Disable process creation breakpoints 59
TASK.Watch.ENABLE	Enable watch system 59
TASK.Watch.ENABLEBP	Enable process creation breakpoints 60
TASK.Watch.Option	Set watch system options 60
TASK.Watch.View	Show watched processes 61
Linux PRACTICE Functions	64
TASK.ARCHITECTURE()	Target architecture 64
TASK.CONFIG()	OS awareness configuration information 64

TASK.CURRENT()	Magic or space ID of current task	64
TASK.ERROR.CODE()	Awareness error code	65
TASK.ERROR.HELP()	Awareness error help ID	65
TASK.LIB.ADDRESS()	Library load address	65
TASK.LIB.CODESIZE()	Library code size	66
TASK.LIB.PATH()	Library target path and name	66
TASK.MOD.CODEADDR()	Code start address of module	66
TASK.MOD.DATAADDR()	Data start of module	67
TASK.MOD.SIZE()	Size of module	67
TASK.MOD.MAGIC()	Magic value of module	67
TASK.MOD.MCB()	Structure address of module	67
TASK.MOD.NAME()	Name of module magic	68
TASK.MOD.SECTION()	Address of a specified module's section	68
TASK.MOD.SECNAME()	Name of a module section with a given number	68
TASK.MOD.SECADDR()	Address of a module section with a given number	69
TASK.OS.VERSION()	Version of the used Linux OS	69
TASK.PROC.CODEADDR()	Code start address of process	69
TASK.PROC.CODESIZE()	Code size of process	69
TASK.PROC.DATAADDR()	Data start address of process	70
TASK.PROC.DATASIZE()	Data size of process	70
TASK.PROC.FileName()	Filename of process	70
TASK.PROC.LIST()	List of processes	71
TASK.PROC.MAGIC()	Magic value of process	71
TASK.PROC.MAGIC2SID()	Space ID of process	72
TASK.PROC.NAME()	Name of process	72
TASK.PROC.NAME2TRACEID()	Trace ID of process	72
TASK.PROC.PATH()	Path and file name of executable on target	72
TASK.PROC.PSID()	Process ID	73
TASK.PROC.SID2MAGIC()	Magic value of process	73
TASK.PROC.SPACEID()	Space ID of process	73
TASK.PROC.TCB()	Control structure address of task	73
TASK.PROC.TRACEID()	Trace ID of process	74
TASK.PROC.VMAEND()	End address of a process virtual memory area	74
TASK.PROC.VMASTART()	Start address of a process virtual memory area	74
TASK.VERSION.BUILD()	Build number of Linux awareness	75
TASK.VERSION.DATE()	Build date of Linux awareness	75
Error Messages		76
Appendix		77
Appendix A: insmod patch for Linux 2.4		77
FAQ		80

Training Linux Debugging	(training_rtos_linux.pdf)	1
Introduction		5
Documentation Updates		5
Related Documents and Tutorials		5
Basic Terms on Embedded Linux		6
Linux Components		6
The Linux Awareness		7
Virtual Memory Management in Linux		9
Run-Mode vs. Stop-Mode Debugging		15
Kernel Configuration		17
Setting up a Script for Linux-Aware debugging		20
Linux Setup-Steps and -Commands		20
Example Linux Setup-Scripts		32
Debugging Linux Components		34
The Kernel		34
Kernel Modules		37
Processes		39
Linux Specific Windows		45
Displaying the Task List		45
Kernel Module List		46
File System Information		47
Kernel Log Buffer		48
Device Tree		49
RAM Dump Generation		49
Linux Trace		50
Overview		50
Context ID Trace for Arm Cortex-A		51
OTM Trace for PowerArchitecture based QorIQ Processors		51
Using the LOGGER for Task Switch Trace		52
Troubleshooting		55
FAQ		56

OS Awareness Manual LiteOS

OS Awareness Manual LiteOS	(rtos_liteos.pdf)	1
Overview		3
Brief Overview of Documents for New Users		4
Supported Versions		4

Configuration	5
Quick Configuration Guide	6
Hooks & Internals in LiteOS	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Task Context Display	9
Dynamic Task Performance Measurement	9
Task Runtime Statistics	10
Function Runtime Statistics	10
LiteOS specific Menu	12
LiteOS Commands	13
TASK.Task	Display tasks 13
TASK.MUteX	Display mutexes 13
TASK.QUEue	Display queues 14
TASK.SEMaphore	Display semaphores 14
TASK.TIMer	Display timers 15
LiteOS PRACTICE Functions	16
TASK.CONFIG()	OS Awareness configuration information 16

OS Awareness Manual LynxOS

OS Awareness Manual LynxOS	(rtos_lynx.pdf) 1
History	4
Overview	4
Terminology	4
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Quick Configuration Guide	7
Hooks & Internals in LynxOS	7
Features	8
Display of Kernel Resources	8
Task Stack Coverage	8
Task Context Display	9
MMU Support	9
Symbol Autoloader	10
Dynamic Task Performance Measurement	12
Task Runtime Statistics	12
Function Runtime Statistics	13

LynxOS specific Menu	13
Debugging LynxOS Kernel and User Processes	15
LynxOS Kernel	15
User Processes	17
LynxOS Commands	20
TASK.Driver	Display drivers 20
TASK.MMU.SCAN	Scan process MMU space 21
TASK.Process	Display processes 22
TASK.sYmbol	Process symbol management 23
TASK.sYmbol.DELeTe	Unload process symbols and MMU 23
TASK.sYmbol.LOAD	Load process symbols and MMU 24
TASK.sYmbol.Option	Set symbol management options 25
TASK.Thread	Display threads 27
TASK.Watch	Watch processes 28
TASK.Watch.ADD	Add process to watch list 28
TASK.Watch.DELeTe	Remove process from watch list 28
TASK.Watch.DISable	Disable watch system 29
TASK.Watch.DISableBP	Disable process creation breakpoints 29
TASK.Watch.ENable	Enable watch system 29
TASK.Watch.ENableBP	Enable process creation breakpoints 30
TASK.Watch.View	Show watched processes 30
LynxOS PRACTICE Functions	33
TASK.CONFIG()	OS Awareness configuration information 33
TASK.PROC.SPACE()	Space ID of process 33
TASK.DRIVER.START()	Start address of driver 33
TASK.DRIVER.TEXT()	Address of .text section 34
TASK.DRIVER.DATA()	Address of .data section 34
TASK.DRIVER.BSS()	Address of .bss section 34

OS Awareness Manual MicroC/OS-II

OS Awareness Manual MicroC/OS-II	(rtos_ucos.pdf) 1
History	4
Overview	5
Brief Overview of Documents for New Users	6
Supported Versions	6
Configuration	7
Manual Configuration	7
Automatic Configuration	9
Quick Configuration Guide	9
Hooks & Internals in µC/OS-II	9

Features	11
Display of Kernel Resources	11
Task Stack Coverage	11
Task-Related Breakpoints	12
Task Context Display	13
Dynamic Task Performance Measurement	13
Task Runtime Statistics	14
Task State Analysis	15
Function Runtime Statistics	16
µC/OS-II specific Menu	17
µC/OS-II Commands	18
TASK.Event	Display events 18
TASK.Flag	Display flags 18
TASK.Memory	Display memory partitions 19
TASK.PARtition	Display space partitions 19
TASK.PROCess	Display MPU processes 20
TASK.Task	Display tasks 20
TASK.Tlmer	Display timers 21
µC/OS-II PRACTICE Functions	22
TASK.CONFIG()	OS Awareness configuration information 22
TASK.PAR.AVAIL()	Space partitions 22
TASK.PROC.AVAIL()	MPU processes 22
TASK.STRUCT()	Symbol type name of the TCB structure 23

OS Awareness Manual MicroC/OS-III

OS Awareness Manual MicroC/OS-III(rtos_ucos3.pdf)	1
Overview	3
Brief Overview of Documents for New Users	3
Supported Versions	3
Configuration	4
Quick Configuration Guide	5
Hooks & Internals in µC/OS-II3	5
Features	6
Display of Kernel Resources	6
Task Stack Coverage	6
Task-Related Breakpoints	7
Task Context Display	8
Dynamic Task Performance Measurement	8
Task Runtime Statistics	9
Task State Analysis	9
Function Runtime Statistics	10

μC/OS-III specific Menu	11
μC/OS-III Commands	12
TASK.eventFLAG	Display event flags 12
TASK.MEMory	Display memory partitions 12
TASK.MUTEX	Display mutexes 13
TASK.Queue	Display message queues 13
TASK.SEMaphore	Display semaphores 14
TASK.Task	Display tasks 14
TASK.TiMeR	Display timers 15
μC/OS-III PRACTICE Functions	17
TASK.CONFIG()	OS Awareness configuration information 17
TASK.STRUCT()	OS structure names 17

OS Awareness Manual MicroC3/Compact

OS Awareness Manual MicroC3/Compact	(rtos_uc3cmp.pdf)	1
History		3
Overview		3
Brief Overview of Documents for New Users		4
Supported Versions		4
Configuration		5
Quick Configuration Guide		6
Hooks & Internals in MicroC3/Cmp		6
Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task-Related Breakpoints		8
Dynamic Task Performance Measurement		9
Task Runtime Statistics		9
Function Runtime Statistics		10
MicroC3/Cmp specific Menu		11
MicroC3/Cmp Commands		12
TASK.CYClic	Display cyclic handlers	12
TASK.DaTaQueue	Display data queues	12
TASK.FLaG	Display event flags	13
TASK.MailBoX	Display mailboxes	13
TASK.MemPoolF	Display fixed memory pools	14
TASK.SEMaphore	Display semaphores	14
TASK.TaSK	Display tasks	15
MicroC3/Cmp PRACTICE Functions		16
TASK.CONFIG()	OS Awareness configuration information	16

TASK.ADDR()	Control block address of object ID	16
TASK.CADDR()	Constant block address of object ID	16

OS Awareness Manual MicroC3/Standard

OS Awareness Manual MicroC3/Standard	(rtos_uc3std.pdf)	1
History		4
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Quick Configuration Guide		7
Hooks & Internals in MicroC3/Std		7
Features		8
Display of Kernel Resources		8
Task-Related Breakpoints		9
Dynamic Task Performance Measurement		10
Task Runtime Statistics		10
Function Runtime Statistics		11
MicroC3/Std specific Menu		12
MicroC3/Std Commands		13
TASK.ALarM	Display alarm handlers	13
TASK.CYClic	Display cyclic handlers	13
TASK.DaTaQueue	Display data queues	14
TASK.FLaG	Display event flags	14
TASK.ISR	Display interrupt service routines	15
TASK.MailBoX	Display mailboxes	15
TASK.MemPoolF	Display fixed memory pools	16
TASK.MemPoolL	Display variable memory pools	16
TASK.MsgBuFfer	Display message buffers	17
TASK.MuTeX	Display mutexes	17
TASK.PORT	Display rendezvous ports	18
TASK.SEMaphore	Display semaphores	18
TASK.TaSK	Display tasks	19
MicroC3/Std PRACTICE Functions		20
TASK.CONFIG()	OS Awareness configuration information	20

OS Awareness Manual MQX

OS Awareness Manual MQX	(rtos_mqx.pdf)	1
History		4

Overview	4
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Quick Configuration Guide	7
Hooks & Internals in MQX	7
Features	8
Display of Kernel Resources	8
Task Stack Coverage	8
Task-Related Breakpoints	9
Task Context Display	10
Dynamic Task Performance Measurement	11
Task Runtime Statistics	12
Task State Analysis	13
Function Runtime Statistics	14
MQX specific Menu	16
MQX Commands	17
TASK.Event	Display events 17
TASK.KLog	Display kernel log 17
TASK.LWEvent	Display light weight events 18
TASK.LWMEMPool	Display light weight memory pools 18
TASK.LWMsgQ	Display light weight message queues 19
TASK.LWSem	Display light weight semaphores 19
TASK.MEMPool	Display memory pools 20
TASK.MeSsaGe.POOL	Display message pools 20
TASK.MeSsaGe.QUEUE	Display message queues 21
TASK.MuteX	Display mutexes 22
TASK.SEMaphore	Display semaphores 22
TASK.TASK	Display tasks 23
TASK.TASKQueue	Display task queues 23
MQX PRACTICE Functions	24
TASK.CONFIG()	OS Awareness configuration information 24
TASK.STRUCT()	Structure information 24

OS Awareness Manual MTOS-UX

OS Awareness Manual MTOS-UX	(rtos_mtos.pdf) 1
History	4
Overview	4
Brief Overview of Documents for New Users	4
Supported Versions	5

Configuration	6
Manual Configuration	6
Automatic Configuration	7
Hooks & Internals of MTOS-UX	7
Features	8
SYSC Terminal Emulation	8
Display of Kernel Resources	9
Task Stack Coverage	9
Task Runtime Statistics	10
Task State Analysis	10
Function Runtime Statistics	11
MTOS-UX specific Menu	12
MTOS-UX Commands	13
TASK.DispEvent	Display global event flags 13
TASK.DispFixed	Display fixed pools 13
TASK.DispmBuff	Display message buffers 14
TASK.DispMbx	Display mailboxes 14
TASK.DispPool	Display common pools 15
TASK.DispSem	Display semaphores 16
TASK.DispsVar	Display shared variables 16
TASK.DispTask	Display tasks 17
TASK.DispTlme	Display time & TOD 18
TASK.DispUnit	Display peripheral units 18
TASK.MAP	Mapping suggestion 18
MTOS-UX PRACTICE Functions	19
TASK.CONFIG()	OS Awareness configuration information 19

OS Awareness Manual NetBSD

OS Awareness Manual NetBSD	(rtos_netbsd.pdf) 1
Overview	4
Terminology	4
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Quick Configuration Guide	6
Hooks & Internals in NetBSD	6
Features	8
Display of Kernel Resources	8
Task-Related Breakpoints	8
Task Context Display	9
MMU Support	10

Dynamic Task Performance Measurement	11
Task Runtime Statistics	11
Function Runtime Statistics	12
NetBSD specific Menu	13
Debugging NetBSD Kernel and User Processes	14
NetBSD Kernel	14
User Processes	15
NetBSD Commands	17
TASK.LWP	Display LWPs 17
TASK.MMU.SCAN	Scan process MMU space 17
TASK.Process	Display processes 18
NetBSD PRACTICE Functions	19
TASK.CONFIG()	OS Awareness configuration information 19
TASK.PROC.SPACEID()	Space ID of process 19

OS Awareness Manual NORTi

OS Awareness Manual NORTi	(rtos_norti.pdf)	1
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Quick Configuration Guide		7
Hooks & Internals in NORTi		7
Features		8
Display of Kernel Resources		8
Task Stack Coverage		8
Task-Related Breakpoints		9
Dynamic Task Performance Measurement		10
Task Runtime Statistics		11
Task State Analysis		12
Function Runtime Statistics		13
NORTi specific Menu		15
NORTi Commands		16
TASK.ALarM	Display alarm handlers	16
TASK.CYClc	Display cyclic handlers	16
TASK.DaTaQueue	Display data queues	16
TASK.FLaG	Display event flags	17
TASK.ISR	Display interrupt service routines	17
TASK.MailBoX	Display mailboxes	17
TASK.MemPoolF	Display fixed memory pools	18

TASK.MemPoolL	Display variable memory pools	18
TASK.MsgBuFfer	Display message buffers	19
TASK.MuTeX	Display mutexes	20
TASK.PORt	Display rendezvous ports	20
TASK.SEMaphore	Display semaphores	21
TASK.TaSK	Display tasks	21
NORTi PRACTICE Functions		22
TASK.CONFIG()	OS Awareness configuration information	22

OS Awareness Manual Nucleus PLUS

OS Awareness Manual Nucleus PLUS	(rtos_nucleus.pdf)	1
History		4
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Manual Configuration		6
Automatic Configuration		7
Quick Configuration Guide		7
Hooks & Internals of Nucleus PLUS		8
Features		9
DBUG+ Terminal Emulation		9
Display of Kernel Resources		9
Display of History Component		10
Task Stack Coverage		10
Task-Related Breakpoints		11
Task Context Display		12
SMP Support		12
Dynamic Task Performance Measurement		13
Task Runtime Statistics		13
Task State Analysis		14
Function Runtime Statistics		15
Nucleus specific Menu		17
Debugging Nucleus Processes		17
Nucleus Commands		20
TASK.DynMem	Display dynamic memory status	20
TASK.EventStat	Display event group status	20
TASK.FDT	Display flattened device tree	21
TASK.HIsr	Display HISRs	21
TASK.HISTory	Display Nucleus history	22
TASK.MbxStat	Display mailbox status	22

TASK.PartMem	Display partition memory status	23
TASK.PipeStat	Display pipe status	23
TASK.ProcList	Display process list	24
TASK.QueueStat	Display queue status	24
TASK.REGistry	Display registry entries	25
TASK.SemaStat	Display semaphore status	25
TASK.TaskStat	Display task status	25
TASK.Tlmerstat	Display timer status	26
Nucleus PLUS PRACTICE Functions		27
TASK.CONFIG()	OS Awareness configuration information	27
TASK.DM.AVAIL()	Bytes of dyn. pool	27
TASK.PL.ENTRY()	Entry address of process	27

OS Awareness Manual NuttX

OS Awareness Manual NuttX	(rtos_nuttX.pdf)	1
Overview		3
Brief Overview of Documents for New Users		3
Supported Versions		4
Configuration		5
Quick Configuration Guide		6
Hooks & Internals in NuttX		6
Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task-Related Breakpoints		8
Task Context Display		9
Dynamic Task Performance Measurement		9
Task Runtime Statistics		10
Function Runtime Statistics		11
NuttX specific Menu		12
NuttX Commands		13
TASK.Task	Display tasks	13
TASK.BuiLTinAPP	Display built-in applications	13
NuttX PRACTICE Functions		14
TASK.CONFIG()	OS Awareness configuration information	14

OS Awareness Manual OKL4

OS Awareness Manual OKL4	(rtos_okl4.pdf)	1
Overview		3
Terminology		3

Brief Overview of Documents for New Users	3
Supported Versions	4
Configuration	5
Quick Configuration Guide	5
Hooks & Internals in OKL4	6
Features	7
Display of Kernel Resources	7
Task-Related Breakpoints	7
Task Context Display	8
MMU Support	9
Dynamic Task Performance Measurement	9
Task Runtime Statistics	10
Function Runtime Statistics	10
OKL4 Specific Menu	12
OKL4 Commands	13
TASK.CList	Display capability lists 13
TASK.MMU.SCAN	Scan MMU address space 13
TASK.Space	Display address spaces 14
TASK.Thread	Display threads 15
OKL4 PRACTICE Functions	16
TASK.CONFIG()	OS Awareness configuration information 16

OS Awareness Manual OS21

OS Awareness Manual OS21 (rtos_os21.pdf)	1
History	3
Overview	3
Brief Overview of Documents for New Users	4
Supported Versions	4
Configuration	5
Quick Configuration Guide	6
Hooks & Internals in OS21	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Task Context Display	9
Dynamic Task Performance Measurement	10
Task Runtime Statistics	10
Task State Analysis	12
Function Runtime Statistics	13

OS21 specific Menu	14
OS21 Commands	15
TASK.EVEnT	Display event groups 15
TASK.MeSsaGe	Display message queue 16
TASK.MuTeX	Display mutexes 17
TASK.PARTition	Display partition 18
TASK.SEMaphore	Display semaphores 18
TASK.Task	Display tasks 19
OS21 PRACTICE Functions	21
TASK.CONFIG()	OS Awareness configuration information 21

OS Awareness Manual OS-9

OS Awareness Manual OS-9	(rtos_os9.pdf)	1
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Hooks in OS-9		7
Features		8
Display of Kernel Resources		8
Symbol Relocation		8
Task Runtime Analysis		9
Task State Analysis		9
Function Runtime Statistics		10
Task Selective Debugging		10
System Calls		10
OS-9 Commands		11
sYmbol.RELOCate.Auto	Control automatic relocation	11
sYmbol.RELOCate.Base	Define base address	11
sYmbol.RELOCate.List	List relocation info	12
sYmbol.RELOCate.Magic	Define program magic number	12
sYmbol.RELOCate.Passive	Define passive base address	12
TASK.SYSGLOB	Display time	13
TASK.PROCS	Process table	13
TASK.PROCSL	Extended process table	13
TASK.QUEUES	Process queues	14
TASK.EVENTS	Event table	14
TASK.ALARMS	Alarm table	14
TASK.MDIR	Module table	15
TASK.MFREE	Free memory	15
TASK.DEVS	Device table	15

TASK.IRQS	Interrupt polling table	15
TASK.CCTL	Cache control	16
TASK.EXIT	Exit system call	16
TASK.SEND	Send signal	16
TASK.SysCall	Generic system call	16
OS9 specific Functions		18
TASK.MDIR.ADDRESS()	Program base address from module directory	18

OS Awareness Manual OSE Delta

OS Awareness Manual OSE Delta	(rtos_ose_delta.pdf)	1
History		4
Overview		4
Terminology		5
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Manual Configuration		6
Automatic Configuration		7
Quick Configuration Guide		7
Hooks & Internals in OSE Delta		8
Features		9
Terminal Emulation for dbgprintf		9
Display of Kernel Resources		9
Task Stack Coverage		9
Task-Related Breakpoints		10
Task Context Display		11
MMU Support		12
SMP Support		15
Dynamic Task Performance Measurement		15
Task Runtime Statistics		16
Task State Analysis		17
Function Runtime Statistics		18
OSE Delta specific Menu		19
Debugging OSE Load Modules		20
OSE Delta Commands		25
TASK.DBIOS	Display bios modules	25
TASK.DBLock	Display blocks	25
TASK.DConf	Display kernel configuration	26
TASK.DLoadMod	Display load modules	26
TASK.DPOOL	Display pools	27
TASK.DProc	Display processes	28

TASK.DProGram	Display loaded programs	29
TASK.MMU.SCAN	Scan OSE MMU	30
TASK.RAMLOG	Display ramlog	30
TASK.SYMBOL	Symbol handling of load modules	31
PRACTICE Functions		32
TASK.CONFIG()	OS Awareness configuration information	32
TASK.PG.ADDR()	Segment address of program	32
TASK.PG.RELOC()	Relocation address for program	33
TASK.LM.LIST()	Next magic in load module list	33
TASK.LM.HANDLE()	Install handle of load module	33
TASK.LM.FILENAME()	File name for load module	34
TASK.LM.RELOCINFO()	Relocation information for load module	34
TASK.LM.RELOCITER()	Relocation information of index section	34

OS Awareness Manual OSE Epsilon

OS Awareness Manual OSE Epsilon (rtos_ose_epsilon.pdf)	1
History	3
Overview	3
Terminology	3
Brief Overview of Documents for New Users	4
Supported Versions	4
Configuration	5
Quick Configuration Guide	6
Hooks & Internals in OSE Epsilon	6
Features	7
Display of Kernel Resources	7
Task Runtime Statistics	7
Task State Analysis	8
Function Runtime Statistics	9
Task Stack Coverage	10
OSE Epsilon specific Menu	12
OSE Epsilon Commands	13
TASK.DProc	Display processes 13
TASK.DQueue	Display signal queue 13
OSE Epsilon PRACTICE Functions	15
TASK.CONFIG()	OS Awareness configuration information 15

OS Awareness Manual OSEck

OS Awareness Manual OSEck (rtos_oseck.pdf)	1

Overview	3
Terminology	3
Brief Overview of Documents for New Users	4
Supported Versions	4
Configuration	5
Quick Configuration Guide	6
Hooks & Internals in OSEck	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Dynamic Task Performance Measurement	9
Task Runtime Statistics	10
Task State Analysis	11
Function Runtime Statistics	12
OSEck specific Menu	13
OSEck Commands	14
TASK.PoolL	Display pools 14
TASK.Process	Display processes 15
TASK.SysInfo	Display system information 16
OSEck PRACTICE Functions	17
TASK.CONFIG()	OS Awareness configuration information 17

OS Awareness for OSEK/ORTI

OS Awareness Manual OSEK/ORTI

OS Awareness Manual OSEK/ORTI	(rtos_orti.pdf) 1
History	4
Overview	5
Brief Overview of Documents for New Users	6
Supported Versions	6
Configuration	7
Quick Configuration Guide	7
Hooks and Internals in ORTI	7
Debug Features	10
Display of Kernel Resources	10
Task Stack Coverage	10
Task-Related Breakpoints	11
Task Context Display	12
Dynamic Task Performance Measurement	12

Brief Overview of Documents for New Users	6
Supported Versions	6
Configuration	7
Quick Configuration Guide	7
Hooks and Internals in PikeOS	8
Features	9
Display of Kernel Resources	9
Task Stack Coverage	9
Task-Related Breakpoints	10
Task Context Display	12
MMU Support	14
Symbol Autoloader	17
SMP Support	19
POSIX Personality	19
APEX Personality	20
Linux Personality	20
Dynamic Task Performance Measurement	20
Task Runtime Statistics	21
Function Runtime Statistics	22
PikeOS specific Menu	23
Debugging PikeOS Components	24
PikeOS Kernel	24
System Extensions	25
User Tasks	26
POSIX	28
APEX	30
ELinOS	33
PikeOS Commands	36
EXTension.AXInfo	Display APEX information 36
EXTension.AXProcess	Display APEX processes 36
EXTension.ELModule	Display ELinOS modules 37
EXTension.ELProcess	Display ELinOS processes 37
EXTension.ELThread	Display ELinOS threads 37
EXTension.PXThread	Display POSIX threads 38
TASK.DrvList	Display system information 39
TASK.INFO	Display system information 40
TASK.Option	Set awareness options 40
TASK.ResPart	Display resource partitions 41
TASK.TaskAdspace	Display task address space 41
TASK.TaskFile.ADD	Map file name to task name 41
TASK.TaskFile.view	Display file name to task name mapping 42
TASK.TaskList	Display 'PikeOS' tasks 43

TASK.ThrliSt	Display threads	44
PikeOS PRACTICE Functions		45
TASK.CONFIG()	OS Awareness configuration information	45
TASK.TASK.MAGIC()	magic number of task	45
TASK.TASK.ID()	ID of task	45
TASK.TASK.NAME()	Name of task	46
TASK.TASK.ID2NAME()	Convert task ID to name	46
TASK.TASKNAME2ID()	Convert task name to ID	46
TASK.TASKFILE()	Symbol file name of task	46
EXT.AXPROCESS.THREAD()	PikeOS thread of APEX process	47
EXT.AXPROCESS.THREAD2()	PikeOS thread of APEX process	47
EXT.ELINOS.SPACEID()	Space ID of ELinOS personality	47
EXT.ELPROCESS.NAME()	Name of ELinOS process	47
EXT.ELLIBRARY.ADDRESS()	Load address of ELinOS library	48
EXT.ELLIBRARY.SPACEID()	Space ID of ELinOS library	48
EXT.ELLIBRARY.NAME()	Name of ELinOS library	48
EXT.ELMODULE.MAGIC()	Module magic number of ELinOS module	48
EXT.ELMODULE.NAME()	Name of ELinOS module	49
EXT.ELMODULE.SECADDR()	Section address of ELinOS module	49

OS Awareness Manual PrKERNEL

OS Awareness Manual PrKERNEL	(rtos_prkernel.pdf)	1
Overview		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Quick Configuration Guide		7
Hooks & Internals in PrKERNEL		7
Features		8
Display of Kernel Resources		8
Task Stack Coverage		8
Task-Related Breakpoints		9
Task Context Display		10
Dynamic Task Performance Measurement		11
Task Runtime Statistics		11
Function Runtime Statistics		12
PrKERNEL specific Menu		13
PrKERNEL Commands		14
TASK.ALarM	Display alarm handlers	14
TASK.CYClc	Display cyclic handlers	14
TASK.DaTaQueue	Display data queues	15

TASK.FLaG	Display event flags	15
TASK.MailBoX	Display mailboxes	16
TASK.MemPoolF	Display fixed memory pools	16
TASK.MemPoolL	Display variable memory pools	17
TASK.MsgBuFfer	Display message buffers	17
TASK.MuTeX	Display mutexes	18
TASK.PORTt	Display ports	18
TASK.SEMaphore	Display semaphores	19
TASK.TaSK	Display tasks	19
PrKERNEL PRACTICE Functions		21
TASK.CONFIG()	OS Awareness configuration information	21

OS Awareness Manual pSOS+

OS Awareness Manual pSOS+	(rtos_psos.pdf)	1
History		4
Overview		4
Brief Overview of Documents for New Users		4
Supported Versions		5
Configuration		6
Manual Configuration		6
Automatic Configuration		7
Quick Configuration Guide		8
Hooks & Internals in pSOS+		8
Features		9
Display of Kernel Resources		9
TRACE32 Board Support Package with pROBE+ Terminal Emulation		10
Task Runtime Statistics		10
Task State Analysis		10
Function Runtime Statistics		11
System Calls		11
Task Selective Debugging		11
pSOS specific Menu		12
pSOS Commands for i386, M68k and PPC		13
TASK.QC	Configuration	13
TASK.QD	Date and time	14
TASK.QO	Objects	15
TASK.QP	Partitions	15
TASK.QQ	Queues	16
TASK.QR	Regions	17
TASK.QS	Semaphores	18
TASK.QT	Tasks	19

TASK.QV	Version	19
TASK.SC	System calls	20
pSOSx86 Commands		21
TASK.QC	Configuration	21
TASK.QP	Process table	22
TASK.QT	Time	22
TASK.TASKState	Mark task state words	22
pSOS PRACTICE Functions		24
TASK.CONFIG()	OS Awareness configuration information	24

OS Awareness Manual PXROS

OS Awareness Manual PXROS	(rtos_pxros.pdf)	1
History		3
Overview		3
Brief Overview of Documents for New Users		4
Supported Versions		4
Configuration		5
Quick Configuration Guide		5
Hooks & Internals in PXROS		6
Debug Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task-Related Breakpoints		8
Task Context Display		9
SMP Support		9
Dynamic Task Performance Measurement		10
PXROS Specific Menu		11
Trace Features		12
Task Runtime Statistics		12
Function Runtime Statistics		13
CPU Load Analysis		15
PXROS Specific Menu for Tracing		16
PXROS Commands		17
TASK.ListmbX	Display mailboxes	17
TASK.ListObject	List objects	17
TASK.ListObj.DeLaY	Display delay objects	18
TASK.ListObj.MailBoX	Display mailboxes	18
TASK.ListObj.MemClass	Display memory classes	19
TASK.ListObj.MeSsaGe	Display message objects	19
TASK.ListObj.OPool	Display object pools	20

OS Awareness and Run Mode Debugging for QNX

Run Mode Debugging Manual QNX

Run Mode Debugging Manual QNX(rtos_qnx_run.pdf)	1
Basic Concepts	3
pdebug	3
Switching to Run Mode Debugging	3
The Space ID for Run Mode Debugging	4
Process Debugging	5
Quick Start Example for ARM	7
Switching between Run & Stop Mode Debugging	9
Commands for Run Mode Debugging	13
Breakpoint Conventions	14

OS Awareness Manual QNX

OS Awareness Manual QNX(rtos_qnx_stop.pdf)	1
History	5
Overview	6
Terminology	6
Brief Overview of Documents for New Users	7
Supported Versions	7
Configuration	8
Quick Configuration Guide	8
Hooks & Internals in QNX	8
Debug Features	11
Display of Kernel Resources	11
Task Stack Coverage	11
Task-Related Breakpoints	12
Task Context Display	14
MMU Support	16
Symbol Autoloader	20
SMP Support	22
Dynamic Task Performance Measurement	23
QNX specific Menu	24
Trace Features	26

Task Runtime Statistics	26
Task State Analysis	27
Function Runtime Statistics	28
QNX specific Menu for Tracing	30
Debugging QNX Components	31
Initial Program Loader (IPL)	31
QNX Kernel	31
User Processes	33
Trapping Segmentation Violation	36
QNX Commands	38
TASK.ASINFO	Display address space information 38
TASK.IFS	Display directory of IFS 38
TASK.MMU.SCAN	Scan process MMU space 39
TASK.Option	Set awareness options 40
TASK.PIDIN	Display “pidin” like information 40
TASK.Process	Display processes 41
TASK.QVM	Display VMs 42
TASK.SHMEM	Display contents of shmем 42
TASK.SLOGGER2	Display contents of slogger2 buffers 43
TASK.sYmbol	Process symbol management 44
TASK.sYmbol.DELeTe	Unload process symbols and MMU 44
TASK.sYmbol.DELeTeLib	Unload library symbols 45
TASK.sYmbol.LOAD	Load process symbols and MMU 45
TASK.sYmbol.LOADLib	Load library symbols 46
TASK.sYmbol.Option	Set symbol management options 46
TASK.Thread	Display threads 48
TASK.TLOGger	Display tracelogger buffer 49
TASK.TLOGger.VMLOGger	Copy tracelogger buffer to LOGGER 50
TASK.Watch	Watch processes 51
TASK.Watch.ADD	Add process to watch list 51
TASK.Watch.DELeTe	Remove process from watch list 51
TASK.Watch.DISable	Disable watch system 52
TASK.Watch.DISableBP	Disable process creation breakpoints 52
TASK.Watch.ENABLE	Enable watch system 52
TASK.Watch.ENABLEBP	Enable process creation breakpoints 53
TASK.Watch.View	Show watched processes 53
QNX PRACTICE Functions	56
TASK.ASINFO.SIZE()	Size of address space 56
TASK.ASINFO.START()	Start of address space 56
TASK.CONFIG()	OS Awareness configuration information 57
TASK.CORE.ASSIGN()	Core assignment 57
TASK.CURRENT()	Current process or thread 57

TASK.LIB.ADDRESS()	Address of library	58
TASK.PROC.ID()	Process ID	59
TASK.PROC.MAGIC()	Magic number of process	59
TASK.PROC.NAME()	Name of process	59
TASK.PROC.SID2MAGIC()	Process of space ID	59
TASK.PROC.SPACE()	Space ID of process	60
TASK.PROC.THREADS()	List of threads	60
TASK.PROC.TTB()	TTB of process	60
TASK.QVM.FORMAT()	Machine ID of VM	61
TASK.QVM.MAGIC()	Magic number of VM	61
TASK.QVM.MID()	Machine ID of VM	61
TASK.QVM.NAME()	Name of VM	61
TASK.QVM.VMLIST()	List of VMs	62
Appendix		63
Appendix A: Kernel debug information		63

OS Awareness Manual QXK

OS Awareness Manual QXK	(rtos_qxk.pdf)	1
Overview		3
Brief Overview of Documents for New Users		3
Supported Versions		4
Configuration		5
Quick Configuration Guide		6
Hooks & Internals in QXK		6
Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task-Related Breakpoints		8
Task Context Display		9
Dynamic Task Performance Measurement		10
Task Runtime Statistics		10
Function Runtime Statistics		11
QXK specific Menu		13
QXK Commands		14
TASK.ActiveObj	Display active objects	14
TASK.EXtTHRead	Display extended threads	14
TASK.MuTeX	Display mutexes	15
TASK.SEMaphore	Display semaphores	15
TASK.MEMPool	Display memory pools	16
QXK PRACTICE Functions		17
TASK.CONFIG()	OS Awareness configuration information	17

OS Awareness Manual REALOS	(rtos_realos.pdf)	1
Overview		3
Brief Overview of Documents for New Users		4
Supported Versions		4
Configuration		5
Quick Configuration Guide		6
Hooks & Internals in REALOS		6
Features		7
Display of Kernel Resources		7
Task-Related Breakpoints		8
Task Stack Coverage		9
Task Context Display		9
Dynamic Task Performance Measurement		10
Task Runtime Statistics		10
Function Runtime Statistics		12
REALOS specific Menu		13
REALOS Commands		14
TASK.ALarM	Display alarm handlers	14
TASK.CYClic	Display cyclic handlers	14
TASK.DaTaQueue	Display data queues	15
TASK.FLaG	Display event flags	15
TASK.MailBoX	Display mailboxes	16
TASK.MemPoolF	Display fixed memory pools	16
TASK.MemPoolL	Display variable memory pools	17
TASK.MsgBuFfer	Display message buffers	17
TASK.MuTeX	Display mutexes	18
TASK.SEMaphore	Display semaphores	19
TASK.TaSK	Display tasks	19
REALOS PRACTICE Functions		20
TASK.CONFIG()	OS Awareness configuration information	20

OS Awareness Manual RealTimeCraft

OS Awareness Manual RealTimeCraft	(rtos_realtimecraft.pdf)	1
Brief Overview of Documents for New Users		3
Configuration		4
Quick Configuration		4
Hooks in XEC 68		5
Features		6
Display of Kernel Resources		6

Function Runtime Statistics	6
Task Runtime Analysis	6
Task State Analysis	7
System Call Trace	7
XEC 68 Commands	8
TASK.DeLaY	Delay table 8
TASK.MailBoX	Mailbox table 8
TASK.SEMaphore	Semaphore table 9
TASK.SysCall	Execute XEC 68 system call 9
TASK.TASK	Task table 9

OS Awareness Manual RIOT

OS Awareness Manual RIOT	(rtos_riot.pdf)	1
Overview		3
Terminology		3
Brief Overview of Documents for New Users		4
Supported Versions		4
Restrictions		4
Configuration		6
Quick Configuration Guide		6
Hooks & Internals in RIOT		7
Features		8
Display of Kernel Resources		8
Task Stack Coverage		8
Task-Related Breakpoints		9
Task Context Display		10
Dynamic Task Performance Measurement		11
Task Runtime Statistics		12
Function Runtime Statistics		13
RIOT Specific Menu		15
RIOT Commands		16
TASK.MailBoX	Display mailboxes	16
TASK.MuTeX	Display mutexes	17
TASK.RingBuffer	Display ring buffers	18
TASK.RMutex	Display recursive mutexes	18
TASK.SEMaphore	Display semaphores	19
TASK.TaskList	Display threads	20
TASK.TIMER	Display timers	21
RIOT PRACTICE Functions		22
TASK.CONFIG()	OS Awareness configuration information	22

OS Awareness Manual RTEMS

OS Awareness Manual RTEMS	(rtos_rtems.pdf)	1
Overview		4
Terminology		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Quick Configuration Guide		6
Hooks & Internals in RTEMS		7
Features		8
Display of Kernel Resources		8
Task Stack Coverage		8
Task-Related Breakpoints		9
Dynamic Task Performance Measurement		10
Task Runtime Statistics		11
Task State Analysis		12
Function Runtime Statistics		13
RTEMS specific Menu		15
RTEMS Commands		16
TASK.INFO	Display API information	16
TASK.INTernal.Mutex	Display internal mutexes	16
TASK.INTernal.Thread	Display internal threads	17
TASK.Posix.CondVar	Display POSIX condition variables	17
TASK.Posix.Mutex	Display POSIX mutexes	18
TASK.CLassic.Extension	Display RTEMS extensions	18
TASK.CLassic.MsgQueue	Display RTEMS message queues	19
TASK.CLassic.Partition	Display RTEMS partitions	19
TASK.CLassic.PEriod	Display RTEMS periods	20
TASK.CLassic.POrt	Display RTEMS ports	20
TASK.CLassic.Region	Display RTEMS regions	21
TASK.CLassic.Semaphore	Display RTEMS semaphores	21
TASK.CLassic.Task	Display RTEMS tasks	22
TASK.CLassic.Tlmer	Display RTEMS timers	23
TASK.Thread	Display all threads	23
RTEMS PRACTICE Functions		25
TASK.CONFIG()	OS Awareness configuration information	25
TASK.CLassic.TASKMAX()	Max. number of tasks	25
TASK.CLassic.TASKLIST()	RTEMS task list	25
TASK.CLassic.TASKNAME()	Name of RTEMS task	26

OS Awareness Manual RTX-ARM

OS Awareness Manual RTX-ARM	(rtos_rtxarm.pdf)	1
Overview		3
Brief Overview of Documents for New Users		3
Supported Versions		4
Configuration		5
Quick Configuration Guide		6
Hooks & Internals in RTX-ARM		6
Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task-Related Breakpoints		8
Dynamic Task Performance Measurement		9
Task Runtime Statistics		9
Task State Analysis		10
Function Runtime Statistics		11
RTX-ARM Specific Menu		12
RTX-ARM Commands		13
TASK.Task / TASK.Thread	Display tasks or threads	13
TASK.MsgQueue	Display message queue	14
TASK.Tlmer	Display timers	14
RTX-ARM PRACTICE Functions		15
TASK.CONFIG()	OS Awareness configuration information	15

OS Awareness Manual RTX C Quadros

OS Awareness Manual RTXC Quadros	(rtos_quadros.pdf)	1
Overview		4
Terminology		4
Brief Overview of Documents for New Users		5
Supported Versions		5
Configuration		6
Quick Configuration Guide		7
Hooks & Internals in RTXC Quadros		7
Features		8
Terminal Emulation		8
Display of Kernel Resources		8
Task Stack Coverage		9
Task-Related Breakpoints		9
Task Context Display		10

Dynamic Task Performance Measurement	11
Task Runtime Statistics	12
Task State Analysis	13
Function Runtime Statistics	14
RTXC Quadros specific Menu	16
RTXC Quadros Commands	17
TASK.ALARm	Display alarms 17
TASK.CouNTER	Display counters 18
TASK.EVENt	Display event sources 19
TASK.EXCEPTION	Display exceptions 19
TASK.LEVEL	Display levels 20
TASK.MAILBOX	Display mailboxes 21
TASK.MUTEX	Display mutex 21
TASK.PARTITION	Display partitions 22
TASK.PIPE	Display pipes 22
TASK.QUEUE	Display queues 23
TASK.SEMAPHORE	Display semaphores 23
TASK.TASK	Display tasks 24
TASK.THREAD	Display threads 25
RTXC Quadros PRACTICE Functions	26
TASK.CONFIG()	OS Awareness configuration information 26
TASK.VERSION()	Awareness information 26
TASK.TASK.LIST()	Next task magic number in task list 27
TASK.TASK.NAME()	Task name 27
TASK.TASK.ID2MAGIC()	Task magic number of task ID 27
TASK.THREAD.LIST()	Next thread magic number in the thread list 28
TASK.THREAD.NAME()	Name of thread 28
TASK.THREAD.ID2MAGIC()	Thread magic number of thread ID 28
TASK.SEMAPHORE.ID2MAGIC()	Magic number of a given semaphore ID 28
TASK.SEMAPHORE.LIST()	Next magic number in the semaphore list 29
TASK.SEMAPHORE.NAME()	Name of semaphore 29
TASK.SEMAPHORE.STATE()	State of semaphore 29
TASK.SEMAPHORE.COUNT()	Count of semaphore 29
TASK.SEMAPHORE.WAITERS.COUNT()	Waiting tasks 30
TASK.SEMAPHORE.WAITERS.LIST()	Next task magic number 30
TASK.MUTEX.LIST()	Next mutex magic number in mutex list 30
TASK.MUTEX.NAME()	Name of mutex 31
TASK.MUTEX.ID2MAGIC()	Mutex magic number of mutex ID 31
TASK.MUTEX.WAITERS.COUNT()	Tasks waiting on mutex 31
TASK.MUTEX.WAITERS.LIST()	Next task magic number 32
TASK.QUEUE.LIST()	Next queue magic number in queue list 32
TASK.QUEUE.NAME()	Name of queue 32
TASK.QUEUE.ID2MAGIC()	Queue magic number of queue ID 33

TASK.QUEUE.WAITERS.COUNT()	Tasks waiting on this queue	33
TASK.QUEUE.WAITERS.LIST()	Next task magic number in waiting list	33
TASK.PIPE.LIST()	Next pipe magic number in pipe list	34
TASK.PIPE.NAME()	Name of pipe	34
TASK.PIPE.ID2MAGIC()	Magic number of pipe ID	34

OS Awareness Manual Rubus OS

OS Awareness Manual Rubus OS	(rtos_rubus.pdf)	1
History		3
Overview		3
Brief Overview of Documents for New Users		3
Supported Versions		4
Configuration		5
Manual Configuration		5
Automatic Configuration		6
Hooks in Rubus OS		6
Features		7
Display of Kernel Resources		7
Task Stack Coverage		7
Task Runtime Statistics		8
Task State Analysis		9
Function Runtime Statistics		10
Rubus specific Menu		11
Rubus Commands		12
TASK.MonDev	I/O device list	12
TASK.MonFile	Open file list	12
TASK.MonLabel	Rubus information	12
TASK.MonMsg	Message queue information	13
TASK.MonMuteX	Blue MUTEX table	13
TASK.MonRSched	Red thread table	14
TASK.MonRSList	Red schedule table	14
TASK.MonThread	Blue thread table	14
Rubus PRACTICE Functions		15
TASK.CONFIG()	OS Awareness configuration information	15

OS Awareness Manual Sciopta

OS Awareness Manual Sciopta	(rtos_sciopta.pdf)	1
Overview		3
Terminology		3
Brief Overview of Documents for New Users		4

Supported Versions	4
Configuration	5
Quick Configuration Guide	5
Hooks & Internals in Sciopta	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Dynamic Task Performance Measurement	9
Task Runtime Statistics	9
Function Runtime Statistics	10
Sciopta Specific Menu	11
Sciopta Commands	12
TASK.ERRmsg	Display last error 12
TASK.ModList	Display module list 12
TASK.POOL	Display pool contents 12
TASK.POolList	Display pool list 13
TASK.POolStat	Display pool statistics 13
TASK.Process	Display process 14
TASK.ProcList	Display process list 14
TASK.Queue	Display queue contents 15
Sciopta PRACTICE Functions	16
TASK.CONFIG()	OS Awareness configuration information 16
TASK.CURRENT()	ID of process 16
TASK.ENTRY()	Entry address of process 17

OS Awareness Manual SMX

OS Awareness Manual SMX(rtos_smx.pdf)	1
History	4
Overview	4
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Quick Configuration Guide	7
Hooks & Internals in SMX	7
Features	8
Display of Kernel Resources	8
Task Stack Coverage	8
Task-Related Breakpoints	9
Task Context Display	10

Dynamic Task Performance Measurement	11
Task Runtime Statistics	12
Task State Analysis	13
Function Runtime Statistics	14
SMX specific Menu	16
SMX Commands	17
TASK.BLOCK	Display blocks 17
TASK.BUCKet	Display buckets 17
TASK.ConFigtab	Display configuration 18
TASK.EvtQueue	Display event queues 18
TASK.EvtTable	Display event tables 19
TASK.eXCHanGe	Display exchanges 20
TASK.LSR	Display LSRs 20
TASK.MeSsaGe	Display messages 21
TASK.PIPE	Display pipes 21
TASK.POOL	Display pools 22
TASK.SEMaphore	Display semaphores 23
TASK.TASK	Display tasks 24
TASK.TIMer	Display timers 24
TASK.TRACE	Display event buffer 25
TASK.TRACEVM	Copy event buffer to LOGGER 25
SMX PRACTICE Functions	27
TASK.CONFIG()	OS Awareness configuration information 27

OS Awareness Manual SYS/BIOS

OS Awareness Manual SYS/BIOS	(rtos_sysbios.pdf) 1
Overview	3
Brief Overview of Documents for New Users	4
Supported Versions	4
Configuration	5
Quick Configuration Guide	5
Hooks & Internals in SYS/BIOS	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Task Context Display	9
Dynamic Task Performance Measurement	10
Task Runtime Statistics	11
Function Runtime Statistics	11
SYS/BIOS Specific Menu	13

SYS/BIOS Commands	14
TASK.CLock	Display clocks 14
TASK.EVenT	Display events 14
TASK.HeapMem	Display heap memories 15
TASK.HWI	Display HWIs 15
TASK.MailBoX	Display mailboxes 16
TASK.MODule	Display used modules 16
TASK.SEMaphore	Display semaphores 17
TASK.SWI	Display SWIs 17
TASK.SYSMIN	Display SysMin buffer 18
TASK.TaSK	Display tasks 18
SYS/BIOS PRACTICE Functions	19
TASK.CONFIG()	OS Awareness configuration information 19

OS Awareness Manual ThreadX

OS Awareness Manual ThreadX (rtos_threadx.pdf)	1
History	4
Overview	4
Terminology	5
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Quick Configuration Guide	7
Hooks & Internals in ThreadX	7
Features	8
Display of Kernel Resources	8
Task Stack Coverage	8
Task-Related Breakpoints	9
Task Context Display	10
SMP Support	11
Dynamic Task Performance Measurement	12
Task Runtime Statistics	12
Task State Analysis	13
Function Runtime Statistics	15
ThreadX specific Menu	17
ThreadX Commands	18
TASK.BLockmem	Display block memory pools 18
TASK.BYtemem	Display byte memory pools 18
TASK.EVenT	Display event flags 19
TASK.ExecLOG	Display thread performance log 19
TASK.MUtex	Display mutexes 20

TASK.QUEUE	Display queues	20
TASK.SEMAPHORE	Display semaphores	21
TASK.THREAD	Display threads	22
TASK.TIMER	Display application timers	23
TASK.TRACE	Display event trace buffer	23
TASK.TRACEVM	Copy event trace buffer to LOGGER	23
ThreadX PRACTICE Functions		25
TASK.CONFIG()	OS Awareness configuration information	25
TASK.TH.MAGIC()	Magic number of thread	25
TASK.BY.MAGIC()	Magic number of byte pool	25
TASK.BL.MAGIC()	Magic number of block pool	26

OS Awareness Manual uClinux

OS Awareness Manual uClinux	(rtos_uclinux.pdf)	1
History		5
Overview		6
Terminology		6
Brief Overview of Documents for New Users		7
Supported Versions		7
Configuration		8
Quick Configuration Guide		8
Hooks & Internals in uClinux		8
Features		10
Terminal Emulation		10
Display of Kernel Resources		10
Task Stack Coverage		10
Task-Related Breakpoints		11
Task Context Display		12
Symbol Autoloader		13
Dynamic Task Performance Measurement		14
Task Runtime Statistics		14
Task State Analysis		15
Function Runtime Statistics		16
uClinux specific Menu		16
Debugging uClinux Kernel and User Processes		18
uClinux Kernel		18
User Processes		19
Kernel Modules		21
uClinux Commands		22
TASK.DMESG	Display the kernel ring buffer	22

TASK.DTask	Display tasks	22
TASK.DTB	Display the device tree blob	23
TASK.DTS	Display the device tree source	23
TASK.FS	Display file system internals	24
TASK.FS.MountDevs	Display mounted devices	24
TASK.FS.PROC	Display /proc file system	24
TASK.FS.Types	Display file system types	24
TASK.MODule	Display kernel modules	25
TASK.MAPS	Display process maps	25
TASK.NET	Display network devices	25
TASK.Option	Set awareness options	26
TASK.PS	Display “ps” output	26
TASK.sYmbol	Process/Module symbol management	28
TASK.sYmbol.DELeTe	Unload process symbols	28
TASK.sYmbol.DELeTeLib	Unload library symbols	29
TASK.sYmbol.DELeTeMod	Unload module symbols	29
TASK.sYmbol.LOAD	Load process symbols	30
TASK.sYmbol.LOADLib	Load library symbols	30
TASK.sYmbol.LOADMod	Load module symbols	31
TASK.sYmbol.Option	Set symbol management options	32
TASK.VMAINFO	Display vmallocated areas	34
TASK.Watch	Watch processes	35
TASK.Watch.ADD	Add process to watch list	36
TASK.Watch.DELeTe	Remove process from watch list	36
TASK.Watch.DISable	Disable watch system	36
TASK.Watch.DISableBP	Disable process creation breakpoints	37
TASK.Watch.ENable	Enable watch system	37
TASK.Watch.ENableBP	Enable process creation breakpoints	37
TASK.Watch.Option	Set watch system options	38
TASK.Watch.View	Show watched processes	39
uCLinux PRACTICE Functions		41
TASK.CONFIG()	OS Awareness configuration information	41
TASK.ERROR.CODE()	Error code	41
TASK.ERROR.HELP()	Error help ID	42
TASK.LIB.ADDRESS()	Load address of library	42
TASK.LIB.CODESIZE()	Code size of library	42
TASK.MOD.CODEADDR()	Code start address of module	42
TASK.MOD.DATAADDR()	Data start address of module	43
TASK.MOD.MAGIC()	Magic value of module	43
TASK.MOD.NAME()	Name of module magic	43
TASK.MOD.SECTION()	Address of module	44
TASK.PROC.CODEADDR()	Code start address of process	44
TASK.PROC.CODESIZE()	Code size of process	44

TASK.PROC.DATAADDR()	Data start address of process	45
TASK.PROC.DATASIZE()	Data size of process	45
TASK.PROC.MAGIC()	Magic value of process	45
TASK.PROC.NAME()	Name of process	45
TASK.PROC.PSID()	Process ID of process	46

OS Awareness Manual uiPLUS

OS Awareness Manual uiPLUS	(rtos_uiplus.pdf)	1
History		3
Overview		3
Brief Overview of Documents for New Users		3
Supported Versions		4
Configuration		5
Hooks & Internals of µiPLUS		5
Features		6
Display of Kernel Resources		6
Task Runtime Statistics		6
Task State Analysis		7
Function Runtime Statistics		7
Task Stack Coverage		7
µiPLUS specific Menu		7
µiPLUS Commands		8
TASK.UIDTQ	Display data queues	8
TASK.UIFLG	Display event flags	8
TASK.UIMBX	Display mailboxes	9
TASK.UIMPF	Display fixed memory pools	9
TASK.UIMPL	Display variable memory pools	9
TASK.UISEM	Display semaphores	10
TASK.UITSK	Display tasks	10
µiPLUS PRACTICE Functions		12

OS Awareness Manual VDK

OS Awareness Manual VDK	(rtos_vdk.pdf)	1
History		3
Overview		3
Terminology		3
Brief Overview of Documents for New Users		3
Supported Versions		4
Configuration		5

Quick Configuration Guide	5
Hooks & Internals in VDK	6
Features	7
Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Dynamic Task Performance Measurement	9
VDK Specific Menu	9
VDK Commands	10
TASK.DevFlag	Display device flags 10
TASK.MemPool	Display memory pools 10
TASK.Semaphore	Display semaphores 11
TASK.Thread	Display threads 11
VDK PRACTICE Functions	12
TASK.CONFIG()	OS Awareness configuration information 12

OS Awareness Manual VxWorks

OS Awareness Manual VxWorks(rtos_vxworks.pdf)	1
History	4
Overview	4
Brief Overview of Documents for New Users	5
Supported Versions	5
Configuration	6
Quick Configuration Guide	7
Hooks & Internals in VxWorks	7
Features	8
Display of Kernel Resources	8
Task Stack Coverage	8
Task-Related Breakpoints	9
Task Context Display	10
MMU Support	12
Symbol Autoloader	15
SMP Support	17
Debugging Modules	17
Debugging Real Time Processes	18
Debugging Protection Domains	19
Dynamic Task Performance Measurement	20
Task Runtime Statistics	21
Function Runtime Statistics	24
Task State Analysis	25

VxWorks specific Menu	27
VxWorks Commands	28
TASK.LKUP	Show system symbol table 28
TASK.MemPShow	Show memory partition 28
TASK.MMU.SCAN	Scan RTP MMU entries 29
TASK.MMU.SCANSIZE	Scan PD MMU entries 29
TASK.ModShow	Show loaded modules 30
TASK.MsgQShow	Show message queues 30
TASK.Option	Set awareness options 31
TASK.PDShow	Show protection domains 31
TASK.RELOC	Relocate system symbols 32
TASK.RTPShow	Show loaded RTPs 32
TASK.SemShow	Show semaphores 33
TASK.SHLShow	Show loaded libraries 33
TASK.TaskInfo	Task information 33
TASK.WDShow	Show watchdogs 34
VxWorks PRACTICE Functions	35
TASK.AVAIL()	Availability of object lists 35
TASK.CONFIG()	OS Awareness configuration information 35
TASK.MODLIST()	Next module magic number 36
TASK.MODNAME()	Module name of module 36
TASK.MODULE()	Segment address of module 36
TASK.RTP.ID()	RTP ID of rtp name 37
TASK.RTP.SEGADDR()	Segment address of RTP 37
TASK.RTP.SEGSIZE()	Segment size of RTP 37
TASK.RTP.SPACEID()	Space ID of RTP ID 38
TASK.RTP.TTB()	TTB address of RTP ID 38
TASK.SHL.ID()	ID of library name 38
TASK.SHL.SEGADDR()	Segment address of library 39
TASK.SHL.SEGSIZE()	Segment size of library 39
TASK.TASKLIST()	Next task magic number in task list 39
TASK.TASKNAME()	Task name of task 40

OS Awareness Manuals for Windows

OS Awareness Manual Windows CE4/CE5

OS Awareness Manual Windows CE4/CE5	(rtos_windows_ce.pdf) 1
History	5
Overview	5
Terminology	5
Brief Overview of Documents for New Users	5

Supported Versions	6
Configuration	7
Manual Configuration	7
Automatic Configuration	8
Quick Configuration Guide	8
Hooks & Internals in Windows CE	9
Features	10
Display of Kernel Resources	10
Task Stack Coverage	10
Task Context Display	11
MMU Support	12
Symbol Autoloader	16
Dynamic Task Performance Measurement	17
Task Runtime Statistics	17
Task State Analysis	19
Function Runtime Statistics	20
Windows CE specific Menu	21
Debugging Windows CE Kernel and User Processes	23
Windows CE Kernel	23
User Processes	27
Trapping Unhandled Exceptions	30
Windows CE Commands	31
TASK.DLL	Display libraries 31
TASK.Event	Display events 31
TASK.MMU.SCAN	Scan process MMU space 32
TASK.MODule	Display libraries 33
TASK.Mutex	Display mutexes 33
TASK.Option	Set awareness options 34
TASK.Process	Display processes 34
TASK.ROM.FILE	Display built-in files 35
TASK.ROM.MODule	Display built-in modules 35
TASK.Semaphore	Display semaphores 35
TASK.sYmbol	Process/DLL symbol management 36
TASK.sYmbol.DELeTe	Unload process symbols and MMU 36
TASK.sYmbol.DELeTeDLL	Unload DLL symbols and MMU 37
TASK.sYmbol.LOAD	Load process symbols and MMU 37
TASK.sYmbol.LOADDLL	Load DLL symbols and MMU 39
TASK.sYmbol.Option	Set symbol management options 40
TASK.Thread	Display threads 42
TASK.Watch	Watch processes 43
TASK.Watch.ADD	Add process to watch list 43
TASK.Watch.DELeTe	Remove process from watch list 43

TASK.Watch.DISable	Disable watch system	45
TASK.Watch.DISableBP	Disable process creation breakpoints	45
TASK.Watch.ENABLE	Enable watch system	45
TASK.Watch.ENABLEBP	Enable process creation breakpoints	46
TASK.Watch.Option	Set watch system options	46
TASK.Watch.View	Show watched processes	47
TASK.WatchDLL	Watch DLLs	50
TASK.WatchDLL.ADD	Add DLL to watch list	50
TASK.WatchDLL.DELete	Remove DLL from watch list	50
TASK.WatchDLL.DISable	Disable DLL watch system	51
TASK.WatchDLL.DISableBP	Disable DLL creation breakpoints	51
TASK.WatchDLL.ENABLEBP	Enable DLL creation breakpoints	52
TASK.WatchDLL.ENABLE	Enable DLL watch system	52
TASK.WatchDLL.Option	Set DLL watch system options	52
TASK.WatchDLL.View	Show watched DLLs	53
Windows CE PRACTICE Functions		55
TASK.CONFIG()	OS Awareness configuration information	55
TASK.CURRENT()	'vmbase' address of process	55
TASK.DLL.CODEADDR()	Address of code segment	55
TASK.DLL.DATAADDR()	Address of data segment	56
TASK.LOG2PHYS()	Convert virtual address to physical address	56
TASK.PROC.CODEADDR()	Address of code segment	56
TASK.PROC.DATAADDR()	Address of data segment	57
TASK.PROC.SPACEID()	Space ID of process	57
TASK.ROM.ADDR()	Section address of ROM module	57
TASK.Y.O()	Symbol option parameters	58

OS Awareness Manual Windows CE6/EC7/EC20

OS Awareness Manual Windows CE6/EC7/EC20	(rtos_windows_ce6.pdf)	1
History		5
Overview		5
Terminology		5
Brief Overview of Documents for New Users		5
Supported Versions		6
Configuration		7
Manual Configuration		7
Automatic Configuration		8
Quick Configuration Guide		8
Hooks & Internals in Windows CE		8
Features		9
Display of Kernel Resources		9

Task-Related Breakpoints	9	
Thread Stack Coverage	10	
Task Context Display	11	
MMU Support	12	
Symbol Autoloader	17	
SMP Support	18	
Dynamic Task Performance Measurement	18	
Task Runtime Statistics	19	
Function Runtime Statistics	20	
Windows CE specific Menu	22	
Debugging Eboot	24	
Debugging Windows CE Kernel	25	
Downloading the Kernel	25	
Debugging the Kernel Startup	26	
Debugging the Kernel	27	
Debugging User Processes and DLLs	28	
Debugging the Process	28	
Debugging DLLs	30	
Trapping Unhandled Exceptions	31	
Windows CE Commands	32	
TASK.HaNDle	Display global handles	32
TASK.MMU.SCAN	Scan process MMU space	32
TASK.Option	Set awareness options	33
TASK.Process	Display processes	34
TASK.ROM.FILE	Display built-in files	35
TASK.ROM.MODule	Display built-in modules	35
TASK.sYmbol	Process/DLL symbol management	36
TASK.sYmbol.DELeTe	Unload process symbols and MMU	36
TASK.sYmbol.DELeTeDLL	Unload DLL symbols and MMU	36
TASK.sYmbol.DELeTeRM	Unload ROM module symbols	37
TASK.sYmbol.LOAD	Load process symbols and MMU	37
TASK.sYmbol.LOADDLL	Load DLL symbols and MMU	37
TASK.sYmbol.LOADRM	Load ROM module symbols	38
TASK.sYmbol.Option	Set symbol management options	38
TASK.Thread	Display threads	39
TASK.Watch	Watch processes	40
TASK.Watch.ADD	Add process to watch list	40
TASK.Watch.DELeTe	Remove process from watch list	40
TASK.Watch.DISable	Disable watch system	41
TASK.Watch.DISableBP	Disable process creation breakpoints	41
TASK.Watch.ENable	Enable watch system	41
TASK.Watch.ENableBP	Enable process creation breakpoints	42

TASK.Watch.Option	Set watch system options	42
TASK.Watch.View	Show watched processes	43
TASK.WatchDLL	Watch DLLs	46
TASK.WatchDLL.ADD	Add DLL to watch list	46
TASK.WatchDLL.DELeTe	Remove DLL from watch list	46
TASK.WatchDLL.DISable	Disable DLL watch system	47
TASK.WatchDLL.DISableBP	Disable DLL creation breakpoints	47
TASK.WatchDLL.ENABLE	Enable DLL watch system	47
TASK.WatchDLL.ENABLEBP	Enable DLL creation breakpoints	48
TASK.WatchDLL.Option	Set DLL watch system options	48
TASK.WatchDLL.View	Show watched DLLs	49
Windows CE PRACTICE Functions		51
TASK.CONFIG()	OS Awareness configuration information	51
TASK.DLL.CODEADDR()	Address of code segment	51
TASK.DLL.CURRENT()	'magic' of DLL	51
TASK.DLL.DATAADDR()	Address of data segment	52
TASK.DLL.MAGIC()	'magic' of DLL	52
TASK.DLL.SECADDR()	Address of section	52
TASK.DLL.SECNUM()	Number of sections	53
TASK.LOG2PHYS()	Convert virtual address to physical address	53
TASK.PROC.CODEADDR()	Address of code segment	53
TASK.PROC.DATAADDR()	Address of data segment	54
TASK.PROC.M2S()	Convert process magic number to space ID	54
TASK.PROC.MAGIC()	Process magic number of process	54
TASK.PROC.S2M()	Convert space ID to process magic number	54
TASK.PROC.SPACEID()	Space ID of process	55
TASK.ROM.ADDR()	Section address of ROM module	55
TASK.ROM.MAGIC()	'Magic' of ROM module	55
TASK.ROM.SECADDR()	Address of section	56
TASK.ROM.SECNUM()	Number of sections	56
TASK.THREAD.LIST()	Thread list	56
TASK.THREAD.PROC()	Process of thread	57
TASK.Y.O()	Symbol option parameters	57

OS Awareness Manual Windows Standard

OS Awareness Manual Windows Standard	(rtos_windows.pdf)	1
History		5
Overview		6
Terminology		6
Brief Overview of Documents for New Users		7
Supported Versions		7

Configuration	8	
Quick Configuration Guide	8	
Hooks & Internals in Windows	8	
Features	9	
Display of Kernel Resources	9	
Task-Related Breakpoints	9	
Task Context Display	11	
MMU Support	12	
Symbol Autoloader	14	
SMP Support	16	
Crash Dump Analysis	16	
Dynamic Task Performance Measurement	17	
Task Runtime Statistics	17	
Function Runtime Statistics	18	
Windows Specific Menu	20	
Debugging Windows Components	22	
Windows Kernel	22	
User Processes	22	
Kernel Modules	26	
Windows Commands	28	
TASK.CrashDump	Windows crash dump analysis	28
TASK.CrashDump.LOADNT	Load the kernel debug symbols	28
TASK.CrashDump.LOADREG	Load the registers from the crash dump	28
TASK.KDBG.SET	Set kernel debugger data block address	29
TASK.MODule	Display kernel modules	29
TASK.NTBASE	Set kernel base address	30
TASK.Process	Display processes	31
TASK.sYmbol	Process/module symbol management	32
TASK.sYmbol.DELeTe	Unload process symbols	32
TASK.sYmbol.DELeTeDLL	Unload library symbols	33
TASK.sYmbol.DELeTeKM	Unload module symbols	33
TASK.sYmbol.DELeTeUM	Unload UEFI module symbols	33
TASK.sYmbol.LOAD	Load process symbols	34
TASK.sYmbol.LOADDLL	Load library symbols	34
TASK.sYmbol.LOADKM	Load module symbols	35
TASK.sYmbol.LOADNT	Load the kernel symbols	35
TASK.sYmbol.LOADUM	Load UEFI runtime service module symbols	35
TASK.sYmbol.Option	Set symbol management options	36
TASK.Thread	Display threads	37
TASK.UefiMODule	Display UEFI runtime service modules	37
PRACTICE Functions	38	
TASK.CONFIG()	OS Awareness configuration information	38

TASK.KDBG()	Kernel debugger data block	38
TASK.KERNELPT()	Kernel page table	38
TASK.LIB.DEBUG()	Library with debug information	39
TASK.LIB.GUID()	GUID of library	39
TASK.LIB.MACHINE()	32bit or 64bit setting of library	40
TASK.LIB.MAGIC()	Magic number of library	40
TASK.LIB.PDBPATH()	Path to PDB file of library	41
TASK.MOD.BASE()	Base address of module	41
TASK.MOD.DEBUG()	Module with debug information	41
TASK.MOD.ENTRY()	Entry address of module	42
TASK.MOD.GUID()	GUID of module	42
TASK.MOD.MACHINE()	32bit or 64bit setting of the module	42
TASK.MOD.MAGIC()	Magic number of module name	43
TASK.MOD.PDBPATH()	Path to PDB file of module	43
TASK.MOD.YF2M()	Magic number of module symbol file	43
TASK.NTBASE()	Kernel base address	43
TASK.PHYMEMBLOCK()	Kernel physical memory descriptor	44
TASK.PROC.DEBUG()	Process with debug information	44
TASK.PROC.GUID()	GUID of the process magic	44
TASK.PROC.MACHINE()	32-bit or 64-bit setting of process	45
TASK.PROC.MAGIC()	Magic value of process	45
TASK.PROC.PDBPATH()	Path to PDB file of process	45
TASK.PROC.SID2MAGIC()	Magic number of process	46
TASK.PROC.SPACEID()	Space ID of process	46
TASK.PROC.TRACEID()	Trace ID of process	46
TASK.UMOD.MACHINE()	32-bit or 64-bit setting of UEFI module	47
TASK.UMOD.MAGIC()	Magic value of UEFI module	47
TASK.UMOD.PDBPATH()	Path to PDB file of UEFI module	47

OS Awareness Manual XOS

OS Awareness Manual XOS	(rtos_xos.pdf)	1
History		3
Overview		4
Terminology		4
Brief Overview of Documents for New Users		4
Supported Versions		4
Configuration		5
Automatic Configuration		5
Quick Configuration Guide		6
Hooks & Internals in XOS		6
Features		7

Display of Kernel Resources	7
Task Stack Coverage	7
Task-Related Breakpoints	8
Dynamic Task Performance Measurement	9
Task Runtime Statistics	9
Function Runtime Statistics	10
XOS specific Menu	12
XOS Commands	13
TASK.Thread	Display threads 13
TASK.TIMER	Display timers 13
XOS PRACTICE Functions	14
TASK.CONFIG()	OS Awareness configuration information 14

OS Awareness Manual Zephyr

OS Awareness Manual Zephyr	(rtos_zephyr.pdf)	1
History		4
Overview		4
Terminology		5
Brief Overview of Documents for New Users		5
Supported Versions		6
Configuration		6
Quick Configuration Guide		7
Hooks & Internals in Zephyr		7
Features		9
Display of Kernel Resources		9
Task Stack Coverage		10
Task-Related Breakpoints		11
Dynamic Task Performance Measurement		12
Task Runtime Statistics		12
Function Runtime Statistics		13
Zephyr specific Menu		14
Zephyr Commands for v1.0		15
TASK.Context	Display contexts	15
TASK.Event	Display microkernel events	15
TASK.Fiber	Display fibers	16
TASK.FIFO	Display microkernel FIFOs	16
TASK.MailBoX	Display microkernel mailboxes	17
TASK.Map	Display microkernel maps	17
TASK.MuTeX	Display microkernel mutexes	18
TASK.NanoFifo	Display nanokernel FIFOs	18

TASK.NanoLifo	Display nanokernel LIFOs	19
TASK.NanoSem	Display nanokernel semaphores	19
TASK.NanoSTack	Display nanokernel stacks	20
TASK.PIPE	Display microkernel pipes	21
TASK.Pool	Display microkernel pools	21
TASK.Semaphore	Display microkernel semaphores	22
TASK.Task	Display tasks	22
TASK.TIMER	Display microkernel timers	23
Zephyr Commands for v1.7		24
TASK.ALERT	Display alerts	24
TASK.MailBOX	Display mailboxes	24
TASK.MEMSLAB	Display memslabs	25
TASK.MSGQ	Display msgqs	25
TASK.MUTEX	Display mutexes	25
TASK.SEMaphore	Display semaphores	26
TASK.THREAD	Display threads	26
TASK.TIMER	Display timers	27
TASK.PIPE	Display pipes	27
TASK.QUEUE	Display queues	27
TASK.ZSTACK	Display zstacks	28
Zephyr PRACTICE Functions		29
TASK.CONFIG()	OS Awareness configuration information	29

Hypervisor Debugging

Hypervisor Debugging User Guide

Hypervisor Debugging User Guide	(hypervisor_user.pdf)	1
Introduction		4
Features		4
Intended Audience		5
Prerequisites		5
Contacting Support		6
Related Documents		7
Restrictions		7
Terms, Abbreviations and Definitions		8
In a Nutshell		14
TRACE32 features for hypervisor debugging		14
Virtualization of physical memory		16
Symbols		17
Machine specific commands		17

About Hypervisors and Virtualization	21
Types of Hypervisors	21
Virtualization of Resources	22
Physical Cores and Virtual CPUs	23
What to know about Hypervisors in TRACE32	24
Run-Mode vs Stop-Mode Debugging	24
Machines in TRACE32	25
Guests in TRACE32	26
CPU Modes	27
Symbol Management	28
Menus and Commands	28
Functional Details of the TRACE32 Hypervisor Support	29
Configuration of TRACE32	34
SMP or AMP Configuration	34
System Settings	35
Configuring the Hypervisor	36
Configuring the Guests	36
Example Script	38
Special Architectural Considerations	39
Configuring the MMU	40
Starting up the System	44
Attaching to the Boot Core	44
Loading the Images	45
Debugging Startup Sequence	48
Attaching to SMP Cores	48
Using Hypervisor Features	50
Viewing the System	50
Viewing Registers	56
Viewing Variables	60
Viewing the Call Stack	61
Interpreting/Understanding the Call Stack	62
Setting Breakpoints	65
Using OS Awarenesses	66
Viewing Page Tables	67
Debugging Scenarios	69
Debugging a Hypervisor	69
Debugging Guests	70
Debugging Processes	71
Tracing	72
Multicore Considerations	72

Hypervisor Awareness Manual Wind River Hypervisor

Hypervisor Awareness Manual Wind River Hypervisor(hv_windriver.pdf)	1
Overview	4
Terminology	4
Brief Overview of Documents for New Users	4
Supported Versions	5
Configuration	6
Quick Configuration Guide	6
Hooks and Internals in Wind River Hypervisor	7
Features	8
Display of Hypervisor Resources	8
Task Stack Coverage	8
Task-Related Breakpoints	9
Task Context Display	10
MMU Support	11
Symbol Autoloader	15
SMP Support	17
Dynamic Task Performance Measurement	17
Task Runtime Statistics	18
Function Runtime Statistics	19
Wind River Hypervisor specific Menu	20
Debugging Wind River Hypervisor Components	21
Hypervisor	21
Virtual Boards	22
Wind River Hypervisor Commands	24
TASK.ThrList	Display hypervisor threads 24
TASK.VirtBoard	Display virtual boards 24
TASK.ConfigVec	Display configuration vector files 25
TASK.REGistry	Display registry 25
TASK.SysInfo	Display system information 25
TASK.CoreState	Display core information 25
Wind River Hypervisor PRACTICE Functions	26
TASK.CONFIG()	Configuration information 26
TASK.PRIV2HYP()	Linear address 26
TASK.THREAD.ID()	ID of thread 26
TASK.THREAD.MAGIC()	Magic of thread 27
TASK.THREAD.PC()	PC of thread 27
TASK.THREAD.TTB()	TTB address of thread 27

TASK.VIRTBOARD.BASE()	Physical base address of virtual board	27
TASK.VIRTBOARD.ID()	ID of virtual board	28
TASK.VIRTBOARD.MAGIC()	Magic of virtual board	28
TASK.VIRTBOARD.START()	Start address of virtual board	28

GDB Support

TRACE32 as GDB Front-End

TRACE32 as GDB Front-End(frontend_gdb.pdf)	1
History	4
Introduction	5
Documentation Updates	5
Related Documents	6
Supported Architectures	6
TRACE32 Setup	7
Configuration File	7
T32Start	7
Connection Setup	8
Debugging Virtual Targets	8
GNU GDBserver	10
UndoDB Reversible Debugger	15
KGDB	16
Troubleshooting	17
GDB Front-End SYStem Commands	18
SYStem.CPU	Select target CPU 18
SYStem.Mode	Establish communication to debug agent 18
SYStem.Option.IMASKASM	Disable interrupts while single stepping 19
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 19
SYStem.Option.MMUSPACES	Separate address spaces by space IDs 19
SYStem.Option.OVERLAY	Enable overlay support 20
SYStem.RESetOut	Reset target 20
SYStem.GDBconfig.BREAKSOFT	Use software breakpoint 20
SYStem.GDBconfig.EXTENDED	Enable/disable gdb extended mode 21
SYStem.GDBconfig.GDBSERVER	Remote target is a gdbserver 21
SYStem.GDBconfig.INFERIORID	Set inferior ID 21
SYStem.GDBconfig.MONITOR	Send monitor command to GDB Back-End 22
SYStem.GDBconfig.NONSTOP	Enable/disable non-stop mode 22
SYStem.PORT	Set communication settings 22
SYStem.GDBSignal	Define signal handling 23

GDB Front-End TASK Commands	24
TRACE32 as GDB Back-End	
TRACE32 as GDB Back-End	1
Introduction	3
Documentation Updates	3
Related Documents	3
Supported Architectures	4
Operation Theory and Restrictions	5
TRACE32 Setup	6
GDB Front-Ends Setup	7
The GNU Project Debugger GDB	7
Eclipse	9
Microsoft Visual Studio	12
Visual Studio Code	14
Remote Serial Protocol	17
Protocol Extensions	17
Symmetrical Multiprocessing Support	17

Converter from GDB to PRACTICE

Converter from GDB to PRACTICE	1
Introduction	3
Launching Converter	4
Converter Limitations	5
Converter-Specific Reserved Identifiers	6
Using History Convenience Variables in CMM Script	7
Using PRACTICE Commands from GDB Scripts	8
Supported Commands	9
Getting In and Out of GDB (TRACE32)	9
Running Programs Under GDB (TRACE32)	9
Stopping and Continuing	10
Examining the Stack	13
Examining Source Files	14
Examining Data	15
C Preprocessor Macros	16
Examining the Symbol Table	17
Using GDB (TRACE32) with Different Languages	18
Altering Execution	18

GDB (TRACE32) Files	18
Specifying a Debugger Target	19
Controlling GDB (TRACE32)	19
Command Files	20
Controlled Output	20
User Interface	21
Others	21

Virtual Targets

Virtual Targets User's Guide

Virtual Targets User's Guide (virtual_targets.pdf)	1
Introduction	3
Intended Audience	4
How This Manual is Organized	4
Related Documents	4
Contacting Support	5
List of Abbreviations	6
Installing TRACE32 Front-End	7
Installing TRACE32 Front-End in MS Windows	7
Installing TRACE32 Front-End in Linux	11
Starting TRACE32 Front-End	17
Starting TRACE32 Front-End in Windows	18
Starting TRACE32 Front-End in Linux	19
Connecting to Virtual Targets	20
Tracing Virtual Targets	22
GDI, MDI, and MCD	22
CADI	22
Supported Virtual Targets	24

Debug Back-Ends

TRACE32 Debug Back-Ends

TRACE32 Debug Back-Ends (backend_overview.pdf)	1
History	3
Introduction	4

PowerView System Configurations	6
Supported Debug Back-Ends	9
CSWP Debug Back-End	9
GTL Debug Back-End	9
Sneakpeek Debug Back-End	10
Verilog Debug Back-End	10
XCP Debug Back-End	11
Infineon DAS Debug Back-End	11
Intel DCI Debug Back-End	11
Tessent Embedded Analytics Debug Back-End	12

CSWP Debug Back-End

CSWP Debug Back-End	(backend_cswp.pdf)	1
History		3
Introduction		4
Set Up the Debug and Trace Session		5
Start TRACE32 PowerView		5
Install the USB Drivers		5
Configure the Debug Session for Arm CoreSight		5
Configure the Trace Recording		8
Command Reference		10
SYStem.CSWP	Configure CSWP communication	10
SYStem.CSWP.Connect	Connect to server	10
SYStem.CSWP.DisConnect	Terminate CSWP session	10
SYStem.CSWP.Init	Initialize CSWP session	10
SYStem.CSWP.SETDEVICE	Set device list	11

GTL Debug Back-End

GTL Debug Back-End	(backend_gtl.pdf)	1
History		4
Introduction		5
Related Documents		5
Contacting Support		5
Abbreviations and Definitions		7
System Architecture		8
PowerView System Configurations		9
Configuring the GTL Plug-in		12
Keep the Graphical User Interface Responsive		16

Timing Adaption	17
Troubleshooting the GTL Back-End	18
JTAG specific	18
Command Reference	20
SYStem.GTL	Configure GTL debug port 20
SYStem.GTL.ARMDAPNAME	Configure name of DAP level transactor 20
SYStem.GTL.CONNECT	Connect to emulation or simulation 21
SYStem.GTL.DISCONNECT	Disconnect from emulation or simulation 21
SYStem.GTL.DMANAME	Name of DMA transactor 22
SYStem.GTL.EXPLore	Display plug-in capabilities 22
SYStem.GTL.GPIONAME	Name of GPIO transactor 24
SYStem.GTL.JTAGPROBENAME	Name of JTAG probe transactor 24
SYStem.GTL.LIBname	Name of 3rd-party plug-in library 24
SYStem.GTL.MODELCOMMAND	Execute command in plug-in 25
SYStem.GTL.MODELCONFIG	Configure emulation options 25
SYStem.GTL.MODELNAME	Select emulation 25
SYStem.GTL.PREBUNDLE	Configure call optimization 26
SYStem.GTL.RESet	Reset GTL settings 26
SYStem.GTL.RESetRESistant	Exempt GTL settings from reset commands 27
SYStem.GTL.SERVERCONFIG	Configure server options 27
SYStem.GTL.SHAREDMODEL	Connect debug port to existing connection 28
SYStem.GTL.SWDNAME	Communicate with target via SWD 28
SYStem.GTL.TRACENAME	Name of trace transactor 29
SYStem.GTL.TransactorConfig	Preconfigure a certain transactor 29

Sneakpeek Debug Back-End

Sneakpeek Debug Back-End	(backend_sneakpeek.pdf) 1
History	3
Introduction	4
Related Documents	4
Contacting Support	4
PowerView System Configurations	6
Configuring Sneakpeek	7
Command Reference	8
SYStem.SNEAKPEEK	Configure SNEAKPEEK debug port 8
SYStem.SNEAKPEEK.CONNECT	Connect to emulation or simulation 8
SYStem.SNEAKPEEK.DISCONNECT	Disconnect from emulation/simulation 9
SYStem.SNEAKPEEK.EXPLore	Display plug-in capabilities 9
SYStem.SNEAKPEEK.MODELCOMMAND	Execute command in plug-in 10
SYStem.SNEAKPEEK.MODELCONFIG	Configure emulation options 10

Debugging via USB User's Guide

Debugging via USB User's Guide	(usbdebug_user.pdf)	1
History		4
Introduction		5
System Requirements		5
Installation of the USB Driver		6
Install the USB Driver on Windows		6
Install the USB Driver on Linux		6
Start a TRACE32 Session for USB Debugging		7
Overview of Configuration Scenarios		7
Start the TRACE32 Session via T32Start		10
Start the TRACE32 Session without T32Start		15
Troubleshooting		17
Select a USB Device via the GUI		18
Select a USB Device via the TRACE32 Commands		20
USB Specific SYStem.CONFIG Commands		21
SYStem.CONFIG.state	Open configuration window for USB debugging	21
SYStem.CONFIG.USB	USB configuration	23
SYStem.CONFIG.USB.RESet	Reset configuration of interface type	25
SYStem.CONFIG.USB.SELect	Connected USB devices	26
SYStem.CONFIG.USB.SELect.view	List connected USB devices	26
SYStem.CONFIG.USB.SELect.SHOWDEvice	Filter the USB device tree	28
SYStem.CONFIG.USB.SELect.ExpandAll	Expand tree	28
SYStem.CONFIG.USB.SELect.CollapseAll	Collapse tree	29
SYStem.CONFIG.USB.SET	Configure all parameters of USB device	30
SYStem.CONFIG.USB.SETBusPort	Configure device by bus port	30
SYStem.CONFIG.USB.SETDEfaults	Apply default USB settings	31
SYStem.CONFIG.USB.SETDEvice	Configure device by VID/PID	31

Verilog Debug Back-End

Verilog Debug Back-End	(backend_verilog.pdf)	1
Introduction		3
Related Documents		3
Contacting Support		3
Abbreviations and Definitions		5
Features		6

Supported Transactors and Simulators	6
JTAG Transactor	6
System Architectures	7
PowerView System Configurations	8
RTL Simulator Integration	11
Step 1: Connecting Signals	11
Step 2: Loading the trace32_verilog_transactor.so	12
Connecting TRACE32 to the Verilog Transactor	14
Keep the Graphical User Interface Responsive	15
Timing Adaption	16
Troubleshooting the JTAG Transactor	17

XCP Debug Back-End

XCP Debug Back-End	(backend_xcp.pdf)	1
History		4
Introduction		5
Related Documents and Tutorials		5
Contacting Support		5
Supported Protocol Variants		7
Target Resources		7
PowerView System Configurations		8
Configuring XCP		11
Reducing XCP Traffic		12
Command Reference		13
SYStem.CONFIG.state	Open XCP configuration window	13
SYStem.CONFIG.XCP	XCP specific settings	14
SYStem.CONFIG.XCP.Connect	Explicitly try to connect to an XCP slave	14
SYStem.CONFIG.XCP.ConnectMode	Configure automatic (dis)connect	15
SYStem.CONFIG.XCP.CONNECTTIMEOUT	Configure time out connection	15
SYStem.CONFIG.XCP.DETECT.state	Detect XCP slaves	16
SYStem.CONFIG.XCP.DBGWRITE	Set DBGWRITE behavior manually	16
SYStem.CONFIG.XCP.DEFAULTSLAVEPORT	Configure port of XCP slave	16
SYStem.CONFIG.XCP.DisConnect	Explicitly disconnect from the XCP slave	17
SYStem.CONFIG.XCP.LOCK	Request exclusive access	17
SYStem.CONFIG.XCP.PROTOCOL	Select the protocol variant	18
SYStem.CONFIG.XCP.SLAVE	Network address	18
SYStem.CONFIG.XCP.SLAVEMode	Configure connect mode of XCP slave	19
SYStem.CONFIG.XCP.slaveINFO	Slave information	20

SYStem.CONFIG.XCP.TEXTserviceFilter	Filter text service output	20
SYStem.CONFIG.XCP.UNLOCK	End exclusive access	21
SYStem.DETECT.XCPTRI	Match XCP resources to access ports	21
XCP Specific Functions		22
In This Section		22
SYStem.CONFIG.XCP.Connected()	Current XCP connection state	22
SYStem.CONFIG.XCP.ConnectMode()	XCP connection mode	22
SYStem.CONFIG.XCP.INFO()	Numeric value of slave property	23
SYStem.CONFIG.XCP.INFO.STR()	String value of slave property	23

Native Process Debugger

Native Process Debugger	(windows_debugger.pdf)	1
Operation Theory		3
Related Documents and Tutorials		3
Configuration		4
Starting a New Process		7
Attach to a Running Process		8
Troubleshooting		9
Native Process Debugger Specific SYStem Commands		10
SYStem.Mode	Establish the communication with the process	10
SYStem.Option.IMASKASM	Disable interrupts while single stepping	10
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	10
SYStem.PROCess	Set the process command line	11
SYStem.CurrentDir	Set the current directory for the process	11
Native Process Debugger Specific TASK Commands		12
TASK.ListPROC	Display the list of running processes	12
TASK.LIB	Display the list of loaded libraries	12
TASK.FREEZE	Freeze a selected thread	13
TASK.THAW	Resume a frozen thread	13

TRACE32 as TCF Agent

TRACE32 as TCF Agent	(app_tcf_setup.pdf)	1
Introduction		4
Restrictions		5
Documentation Updates		5
Related Documents and Tutorials		5

Initial Setup and Requirements	6
TRACE32	6
Eclipse	7
Wind River Workbench	9
Synopsys MetaWare IDE	11
TRACE32 Setup	14
Installing the TRACE32 TCF Eclipse Plug-In	14
Option A: Manual Configuration	16
Option B: Select Executable and Configuration File	21
Establish a Debug Session	24
Start TRACE32	24
TCF Discovery	26
Manual Debug Target Setup	28
Open Debug Perspective Automatically	30
TRACE32 View	31
Troubleshooting	32
TRACE32	32
Eclipse	32
Help Us to Help You	33
TCF Commands	35
SYStem.TCFconfig	TCF-specific setups 35
SYStem.TCFconfig.TASKCONTEXT	Enable/disable task contexts 35

3rd-Party Tool Integrations

Integration for CodeBlocks

Integration for CodeBlocks	(int_codeblock.pdf) 1
Overview	3
Supported Code::Blocks versions	4
Plug-in Installation	5
Plug-in and TRACE32 Configuration	6
Plug-in Configuration	6
TRACE32 Configuration	7
Plug-in Menu and Windows	8
Plug-in Menu	8
Memory window	11
Watches window	12
Registers window	12

Breakpoints List	13
Debugging Example Application	14

Integration for EasyCase

Integration for EasyCase (int_easycase.pdf)	1
Overview	4
Brief Overview of Documents for New Users	4
Operation Theory	5
Installation	6
Startup Sequence	7
Menu Commands	8
EasyCODE V6.x Menu	8
EasyCODE V7.x Menu	11
Working with the EasyCODE Integration	14
Known Bugs	14

Integration for eXDI2 on Windows CE Platform Builder

Integration for eXDI2 on Windows CE Platform Builder (int_exdi2.pdf)	1
Overview	3
Concept of hardware-assisted debugging	4
How hardware-assisted debugging modifies eXDI Architecture?	5
Driver installation and configuration	7
Getting necessary files	10
Creating OS Design	11
Downloading Windows CE image to target and booting system	16
Adding example application to Windows CE image	19
Debugging Windows CE	22
Loading EXE/DLL modules symbols in TRACE32	22
Preparing Windows CE image	23
Driver configuration	23
Debugging session	24
Debugging hardware bring-up	30
Hardware-assisted debugging and KITL	32
Using TRACE32 FDX for KITL Kernel Transport	33
FDX Overview	34

Architecture of KITL over FDX	34
Enabling KITL over FDX	35
Download service	38
Debugging timings	39
Memory caching	39
Troubleshooting	40

Integration with LabView

Integration with LabView (int_labview.pdf)	1
Overview	4
Installation and Configuration of VIs for TRACE32	5
Virtual Instruments for TRACE32 Description	8
Trace32_Init.vi	8
Trace32_Terminate.vi	10
Trace32_AddBreakpoint.vi	12
Trace32_ClrBreakpoint.vi	14
Trace32_ControlBreakpoint.vi	16
Trace32_MemoryRead.vi	19
Trace32_MemoryWrite.vi	22
Trace32_ReadCpuRegister.vi	24
Trace32_WriteCpuRegister.vi	26
Trace32_ReadVariable.vi	28
Trace32_WriteVariable.vi	31
Trace32_ReadVariableFloat.vi	33
Trace32_WriteVariableFloat.vi	35
Trace32_ReadVariableDouble.vi	37
Trace32_WriteVariableDouble.vi	39
Trace32_GetState.vi	41
Trace32_GetSymbol.vi	43
Trace32_GetSymbolFromAddress.vi	45
Trace32_GetAddressFromFile.vi	47
Trace32_GetSourceFileName.vi	49
Trace32_PracticeCmd.vi	51
Trace32_PracticeScript.vi	53
Trace32_PracticeGetState.vi	55
Trace32_RunControl.vi	57
Trace32_System.vi	59
Trace32_Quit.vi	61
Additional Controls for TRACE32	63
Trace32_RunCmd.ctf	63

Trace32_SystemCmd.ctl	64
Trace32_OperationBreakpoint.ctl	65
Trace32_State.ctl	65
Trace32_Error.ctl	66
Quick Start	67

Integration for Rhapsody in C/C++

Integration for Rhapsody in C/C++	(int_rhapsody_cpp.pdf)	1
Overview		3
Architecture of Driver		4
Driver Installation		5
First Run of Integration Driver		6
Selecting Rhapsody Version		8
Preparing Rhapsody Environment		9
Preparing TRACE32		10
Rhapsody Helpers Configuration		12
Rebuilding OXF LangCpp libraries for eCos, GCC and PPC		15
Rebuilding OXF LangC Libraries for OSE, DIAB and PPC		17
Rebuilding OXF LangC Libraries for eCos, GCC and PPC		18
Preparing C/C++ Application for Animation		26
Integration Features		29
Locating Source in Rhapsody		30
Go and Break in TRACE32		35

Integration for Rhapsody in MicroC

Integration for Rhapsody in MicroC	(int_rhapsody_mc.pdf)	1
Overview		3
Brief Overview of Documents for New Users		3
Operation Theory		4
Installation		5
Command Line Parameters		6
Startup Sequence		7
Working with the TRACE32 GBA Driver		7

Integration for Simulink

Integration for Simulink	(int_simulink.pdf)	1
History		4
Introduction		4
Installation		5
System Requirements		5
License Requirements		5
Installing the TRACE32 Integration for Simulink		5
Updating the TRACE32 Integration for Simulink		6
Deinstallation		7
Select Connectivity API		8
Custom Toolchains		9
rtiostream API		10
Demo Project		10
Functional Overview		13
Build Process		22
Configuration of Models		23
Code Coverage Measurement		25
Code Execution Profiling		26
Stack Profiling		26
Stack Memory Information		27
PRACTICE Callbacks		28
Headless Mode		30
DebugIOTool Debugger Abstraction Interface		31
Board Descriptions		32
Configuration of Models		33
Troubleshooting		34
Known Issues		35
Help Us Help You - Export TRACE32 Information		36

Integration for Xilinx ISE

Integration for Xilinx ISE	(int_ise.pdf)	1
Introduction		3
Getting Started		4
Usage on 64-bit Machines		5
Configuring TRACE32		6
Configuring Xilinx ISE 11 and ISE 12		7

Configuring Xilinx ISE 13 and ISE 14	8
Using Xilinx IMPACT via the TRACE32 Debug Interface	9
Using Xilinx ChipScope via the TRACE32 Debug Interface	10
Compatibility	11
Troubleshooting	12
Contacting Support	14

Integration for Xilinx Vivado

Integration for Xilinx Vivado (int_vivado.pdf)	1
Introduction	4
Restrictions	4
System Requirements	5
Setting up the XVCD Bridge	6
Installation	6
Configuration	8
Starting the XVCD Bridge	12
Starting the Bridge with Automatic Start of hw_server	12
Starting the Bridge with Manual Start of hw_server	14
Connecting to Vivado	16
Troubleshooting	17
Closing the XVCD Bridge	18

Integration for X-Tools and X32

Integration for X-Tools and X32	(int_xtools.pdf)	1
Overview		3
Brief Overview of Documents for New Users		3
Operation Theory		4
Installation		4
Startup Sequence		5
Menu Commands		6
Set Breakpoint	Set breakpoint on current line	6
Delete Breakpoint	Delete breakpoint on current line	6
List of all Breakpoints	Lists the breakpoints	6
Go	Continue application	6
Break	Stop application	6
Go until Cursor	Continue application until this line	7
Step Over	Step over function call	7

Step Into	Step into function call	7
Watch Variable	Add variable to watch window	7
Working with the X-TOOLS extensions		8
Known Problems		9

TRACE32 Instruction Set Simulators

API for TRACE32 Instruction Set Simulator

API for TRACE32 Instruction Set Simulator	(simulator_api.pdf)	1
Overview		3
Peripheral Simulation Model		7
Standard function		7
Registers		11
Timers		14
Stall		15
Ports		16
Terminals		18
Communication		19
Files		19
Deprecated functions		21
Practical script commands		22
Peripheral model example		26
Environment		26
Source code listing		29
Initialization scripts		38

Simulator for 68K/ColdFire

Simulator for 68K/ColdFire	(simulator_68k.pdf)	1
History		4
TRACE32 Simulator License		5
Quick Start of the Simulator		6
Peripheral Simulation		8
Troubleshooting		8
FAQ		8
68K and HC16 specific Implementations		9
Memory Classes		9

68K and HC16 specific SYStem Commands		10
SYStem.CONFIG	Configure debugger according to target topology	10
SYStem.CPU	Select CPU type	10
SYStem.LOCK	Lock and tristate the debug port	10
SYStem.MemAccess	Select run-time memory access method	11
SYStem.Mode	Establish the communication with the simulator	11
SYStem.Option.BASE	Select peripheral base address	12
SYStem.Option.MISALIGN	Allow mis-alignment in data accesses	12
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	12
SYStem.Option.PIPELINE	Pre-fetching simulation	13
SYStem.RESetOut	CPU reset command for 68K simulator	14
SYStem.RESetOut	CPU reset command	14
SYStem.Option.IMASKASM	Disable interrupts while single stepping	15
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	15
CPU specific MMU Commands		16
MMU.DUMP	Page wise display of MMU translation table	16
MMU.List	Compact display of MMU translation table	18
MMU.SCAN	Load MMU table from CPU	20
CPU specific TrOnchip Commands		22
TrOnchip	Onchip triggers	22

Simulator for ARC

Simulator for ARC	(simulator_arc.pdf)	1
History		4
Introduction		4
Supported ARC Cores		4
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		6
TRACE32 License		6
Troubleshooting		7
FAQ		7
Quick Start		8
Configure the Debugger to Use the ARCINT Interface		12
Use T32Start		12
Modify an Existing Configuration File		13
Create a New Configuration File		13
ARCINT specific SYStem Commands		14
SYStem.LIBrary	Set path to debug driver of simulator	14
SYStem.PROPeRties	Control properties of the used simulator (usually nSIM)	15

SYStem.PROPerTieS.ADD	Add a property to configure the simulator	15
SYStem.PROPerTieS.Delete	Remove a property to configure the simulator	15
SYStem.PROPerTieS.List	Show all property sets to configure the simulator	16
SYStem.PROPerTieS.Modify	Change property set to configure simulator	16
Access Classes		17
CPU specific SETUP Command		18
SETUP.DIS	Disassembler configuration	18
CPU specific SYStem Commands		19
SYStem.CPU	Select CPU type	19
SYStem.MemAccess	Select run-time memory access method	20
SYStem.Mode	Select target reset mode	21
SYStem.Option	Set a target-specific option	22
SYStem.Option.detectOTrace	Disable auto-detection of on-chip trace	22
SYStem.Option.Endianness	Set the target endianness	22
SYStem.Option.HotBreakPoints	Set breakp. when CPU is running	23
SYStem.Option.IMASKASM	Disable interrupts while single stepping	23
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	24
SYStem.Option.LimmBreakPoints	Software breakpoints with extra NOPs	24
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	25
SYStem.Option.OVERLAY	Enable overlay support	26
SYStem.Option.RegNames	Enable trivial names for core registers	26
SYStem.Option.TIMEOUT	Define maximum time for core response	27
SYStem.state	Show SYStem settings window	27
On-chip Breakpoints/Actionpoints		28
Using On-chip Breakpoints		28
Breakpoints in a ROM Area		28
Limitations		29
TrOnchip.CONVert	Allow extension of address range of breakpoint	30
TrOnchip.VarCONVert	Convert breakpoints on scalar variables	32
TrOnchip.OnchipBP	Number of on-chip breakpoints used by debugger	33
TrOnchip.RESet	Set on-chip trigger to default state	34
TrOnchip.state	Display on-chip trigger window	34
TrOnchip.MCD	MCD API specific on-chip triggers	35
TrOnchip.MCD.McdBreakPoints	Set onchip breakpoint via MCD API	35
TrOnchip.MCD.CoreHalted	Workaround to detect core-halt via MCD API	35
TrOnchip.MCD.CoreRunning	Workaround to start/step core via MCD API	36
CPU specific MMU Commands		37
MMU.DUMP	Page wise display of MMU translation table	37
MMU.List	Compact display of MMU translation table	39
MMU.SCAN	Load MMU table from CPU	40
MMU.Init	Invalidate TLB entries	41
MMU.Set	Set an MMU TLB entry	41

Simulator for Arm and XSCALE	(simulator_arm.pdf)	1
Introduction		5
TRACE32 Simulator License		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		6
Quick Start of the Simulator		7
Peripheral Simulation		9
Troubleshooting		9
FAQ		9
Memory Classes		10
Virtual Terminal		11
Semihosting		11
Coprocessors		12
ARM specific SYStem Commands		13
SYStem.CPU	Select the used CPU	13
SYStem.CONFIG	Configure debugger according to target topology	13
SYStem.CONFIG.SMMU	Internal use	14
SYStem.Mode	Establish the communication with the simulator	15
SYStem.Option.Alignment	Enable alignment exceptions	16
SYStem.Option.BigEndian	Define byte order (endianness)	16
SYStem.Option.DisMode	Define disassembler mode	16
SYStem.Option.DUALPORT	Implicitly use run-time memory access	17
SYStem.Option.IMASKASM	Disable interrupts while single stepping	17
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	18
SYStem.Option.MACHINESPACES	Address extension for guest OSeS	19
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	20
SYStem.Option.OVERLAY	Enable overlay support	21
SYStem.Option.REALTIME	Stall the simulator if faster than real processor	21
SYStem.Option.ZoneSPACES	Enable symbol management for Arm zones	22
SYStem.RESetOut	CPU reset command	28
SYStem.state	Display SYStem.state window	28
ARM Specific TrOnchip Commands		29
TrOnchip.RESet	Reset on-chip trigger settings	29
TrOnchip.Set	Set bits in the vector catch register	29
TrOnchip.StepVector	Step into exception handler	30
TrOnchip.StepVectorResume	Catch exceptions and resume single step	30
TrOnchip.state	Display on-chip trigger window	31
CPU specific MMU Commands		32
MMU.DUMP	Page wise display of MMU translation table	32

MMU.List	Compact display of MMU translation table	36
MMU.SCAN	Load MMU table from CPU	38
CPU specific SMMU Commands		40
SMMU	Hardware system MMU (SMMU)	40
SMMU.ADD	Define a new hardware system MMU	50
SMMU.Clear	Delete an SMMU	52
SMMU.CtxtDescTable	List a context descriptor table	52
SMMU.DumpQueue.<queue>	Dump entries of a queue	53
SMMU.DumpQueue.CMD	Dump cmd queue entries	55
SMMU.DumpQueue.Event	Dump event queue entries	56
SMMU.Register	Peripheral registers of an SMMU	57
SMMU.Register.ContextBank	Display registers of context bank	58
SMMU.Register.Global	Display global registers of SMMU	59
SMMU.Register.MMUregs	Display MMU specific registers	59
SMMU.Register.S1Context	Display stage 1 context descriptor registers	60
SMMU.Register.StreamTblEntry	Display stream table entry registers	60
SMMU.Register.StreamMapRegGrp	Display registers of an SMRG	61
SMMU.RESet	Delete all SMMU definitions	62
SMMU.SSDtable	Display security state determination table	63
SMMU.StreamMapRegGrp	Access to stream map table entries	64
SMMU.StreamMapRegGrp.ContextReg	Display context bank registers	65
SMMU.StreamMapRegGrp.Dump	Page-wise display of SMMU page table	67
SMMU.StreamMapRegGrp.list	List page table entries	69
SMMU.StreamTable	Display a stream table	70
SMMU.StreamTblEntry	Access to a stream table entry	82
SMMU.StreamTblEntry.Dump	Page-wise display of SMMU page table	84
SMMU.StreamTblEntry.list	List page table entries	85
SMMU.StreamTblEntry.Register	Display STE or CD registers	86

Simulator for C166/ST10

Simulator for C166/ST10	(simulator_c166.pdf)	1
TRACE32 Simulator License		4
Quick Start of the Simulator		5
Peripheral Simulation		7
Troubleshooting		8
FAQ		8
CPU specific SYStem Settings and Restrictions		9
SYStem.CONFIG	Configure debugger according to target topology	9
SYStem.CONFIG.DAP	Describe the target configuration	9
SYStem.CPU	CPU type	9

SYStem.LOCK	Lock and tristate the debug port	9
SYStem.MemAccess	Select run-time memory access method	10
SYStem.Mode	Establish the communication with the simulator	10
SYStem.Option.DUALPORT	Run-time memory access for all windows	11
SYStem.Option.IMASKASM	Disable interrupts while single stepping	11
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	12
Special Functions		13
TrOnchip Commands		14
TrOnchip.state	Display on-chip trigger window	14
TrOnchip.RESet	Set on-chip trigger to default state	14
Memory Classes		15

Simulator for H8/300, H8/300H and H8S

Simulator for H8/300, H8/300H and H8S	(simulator_h8.pdf)	1
TRACE32 Simulator License		4
Quick Start of the Simulator		5
Peripheral Simulation		7
Troubleshooting		7
FAQ		7
Specific SYStem Commands		8
SYStem.CONFIG	Configure debugger according to target topology	8
SYStem.CPU	CPU type	8
SYStem.LOCK	Lock and tristate the debug port	8
SYStem.MemAccess	Select run-time memory access method	9
SYStem.Mode	Establish the communication with the simulator	9
SYStem.Option.Advanced	Advanced addressing mode	10
SYStem.Option.EXR	EXR mode setting	10
SYStem.Option.IMASKASM	Disable interrupts while single stepping	10
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	11
CPU specific TrOnchip Commands		12
TrOnchip	Onchip triggers	12
Memory Classes		13

Simulator for HC08/MSC08

Simulator for HC08/MSC08	(simulator_hc08.pdf)	1
TRACE32 Simulator License		4
Quick Start of the Simulator		5
Peripheral Simulation		7

Troubleshooting	8
FAQ	8
CPU specific SYStem Settings and Restrictions	9
SYStem.CPU	Select CPU type 9
SYStem.Mode	Establish the communication with the simulator 9
SYStem.MemAccess	Select run-time memory access method 10
SYStem.Option.IMASKASM	Disable interrupts while single stepping 10
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 11
TrOnchip Commands	12
TrOnchip.state	Display on-chip trigger window 12
TrOnchip.RESet	Set on-chip trigger to default state 12
Memory Classes	13

Simulator for HC12/MCS12

Simulator for HC12/MCS12	(simulator_hc12.pdf) 1
TRACE32 Simulator License	5
Quick Start of the Simulator	6
Peripheral Simulation	8
FAQ	8
CPU specific SYStem Settings and Restrictions	9
SYStem.CPU	Select CPU type 9
SYStem.LOCK	Lock and tristate the debug port 9
SYStem.MemAccess	Select run-time memory access method 10
SYStem.Mode	Establish the communication with the simulator 10
SYStem.Option.BASE	Base address of internal registers 11
SYStem.Option.DUALPORT	Run-time memory access for all windows 11
TrOnchip Commands	12
TrOnchip.state	Display on-chip trigger window 12
TrOnchip.RESet	Set on-chip trigger to default state 12
Memory Classes	13
Banked Applications	14
Background and Compatibility Information	14
SYStem.Option.GLOBAL	Memory accesses are done global 14
SYStem.Option.PAGING	Banked applications 15
SYStem.Option.RAMHM	Alternate RAM mapping 15
SYStem.Option.ROMHM	ROM in second half of map 15
SYStem.Option.TRANS	Transparent mode 15
Using the MMU for HC12DA/DG/DT128	18
SYStem.Option.MEMEXP	Memory expansion 18

Simulator for Intel® x86/x64

Simulator for Intel® x86/x64	(simulator_x86.pdf)	1
TRACE32 Simulator License		4
Quick Start of the Simulator		6
Peripheral Simulation		8
x86 Specific Implementations		9
Access Classes		9
Memory Model		20
Segmentation		21
Troubleshooting		23
FAQ		23
Emulation Modes		24
SYStem.CONFIG	Configure debugger according to target topology	24
SYStem.CPU	CPU type	24
SYStem.LOCK	Lock and tristate the debug port	24
SYStem.MemAccess	Select run-time memory access method	25
SYStem.Mode	Establish the communication with the simulator	25
SYStem Settings and Restrictions		27
SYStem.Option.Address32	Use 32 bit address display only	27
SYStem.Option.IMASKASM	Disable interrupts while single stepping	27
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	27
SYStem.Option.MACHINESPACES	Address extension for guest OSes	28
SYStem.Option.MEMoryMODEL	Define memory model	28
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	31
SYStem.Option.REL	Relocation register	32
SYStem.Option.ZoneSPACES	Enable symbol management for zones	32
CPU specific MMU Commands		35
MMU.DUMP	Page wise display of MMU translation table	35
MMU.List	Compact display of MMU translation table	38
MMU.SCAN	Load MMU table from CPU	40
CPU specific TrOnchip Commands		42
TrOnchip	Onchip triggers	42

Simulator for MIPS

Simulator for MIPS	(simulator_mips.pdf)	1

TRACE32 Simulator License	5
Quick Start of the Simulator	6
Peripheral Simulation	8
Troubleshooting	9
FAQ	9
Memory Classes	10
Belated Trace Analysis	11
MIPS specific SYStem Commands	12
SYStem.CONFIG	Configure debugger according to target topology 12
SYStem.CPU	Select the used CPU 12
SYStem.LOCK	Lock and tristate the debug port 12
SYStem.MemAccess	Select run-time memory access method 13
SYStem.Option.OVERLAY	Enable overlay support 13
SYStem.Option.MMUSPACES	Separate address spaces by space IDs 14
SYStem.Mode	Establish the communication with the target 15
SYStem.Option.Address32	Use 32-bit addresses 15
SYStem.Option.DisMode	Define disassembler mode 16
SYStem.Option.Endianness	Define endianness of target memory 17
SYStem.Option.IMASKASM	Disable interrupts while ASM single stepping 17
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 17
SYStem.RESetOut	CPU reset command 17
CPU specific MMU Commands	18
MMU.DUMP	Page wise display of MMU translation table 18
MMU.FORMAT	Define MMU table structure 19
MMU.List	Compact display of MMU translation table 23
MMU.SCAN	Load MMU table from CPU 24
TrOnchip Commands	26
TrOnchip.state	Display on-chip trigger window 26
TrOnchip.RESet	Set on-chip trigger to default state 26

Simulator for NIOS-II

Simulator for NIOS-II	(simulator_nios.pdf)	1
History		4
Introduction		4
TRACE32 Simulator License		5
Start the Prepared Demo		6
Quick Start		8
General Restrictions		10

CPU specific SYSTem Commands		11
SYSTem.CPU	Select CPU type	11
SYSTem.LOCK	Lock and tristate the debug port	11
SYSTem.MemAccess	Select run-time memory access method	11
SYSTem.Mode	Establish the communication with the simulator	12
SYSTem.CONFIG	Configure debugger according to target topology	12
SYSTem.Option.DCFLUSH	Flush data cache before Step or Go	13
SYSTem.Option.Endianness	Select endianness of core	13
SYSTem.Option.EXCADDR	Define exception address	13
SYSTem.Option.FEXCADDR	Define fast TLB miss exception address	13
SYSTem.Option.FPH	Enable the simulation of floating point instructions	14
SYSTem.Option.ICFLUSH	Invalidate instruction cache before go/step	14
SYSTem.Option.IMASKASM	Mask interrupts during assembler step	14
SYSTem.Option.IMASKHLL	Mask interrupts during HLL step	15
SYSTem.Option.IVRCode	Define code for interrupt vector instruction	15
SYSTem.Option.MMUSPACES	Separate address spaces by space IDs	15
SYSTem.Option.MULDIV	Define if mul and div instructions are supported	17
SYSTem.Option.SIMMMU	Define properties of simulated MMU	17
TrOnchip Commands		18
TrOnchip.state	Display on-chip trigger window	18
TrOnchip.RESet	Set on-chip trigger to default state	18
CPU specific MMU Commands		19
MMU.DUMP	Page wise display of MMU translation table	19
MMU.List	Compact display of MMU translation table	21
MMU.SCAN	Load MMU table from CPU	22
Memory Classes		24
Overview		24
Peripheral Simulation		25
FAQ		25

Simulator for PowerPC

Simulator for PowerPC	(simulator_ppc.pdf)	1
TRACE32 Simulator License		5
Quick Start of the Simulator		6
Peripheral Simulation		8
Troubleshooting		8
FAQ		8
Memory Classes		9
Simulated Registers		10

CPU specific SYSTem Commands		12
SYSTem.CPU	Select CPU type	12
SYSTem.MemAccess	Select run-time memory access method	12
SYSTem.Mode	Establish the communication with the simulator	12
SYSTem.Option.Address32	Define address format display	13
SYSTem.Option.DisMode	Simulator operation mode	14
SYSTem.Option.DUALPORT	Run-time memory access for all windows	14
SYSTem.Option.IMASKASM	Disable interrupts while single stepping	15
SYSTem.Option.IMASKHLL	Disable interrupts while HLL single stepping	15
SYSTem.Option.MMUSPACES	Separate address spaces by space IDs	16
SYSTem.Option.MACHINESPACES	Address extension for guest OSeS	17
SYSTem.Option.NOTRAP	Use alternative software breakpoint instruction	17
SYSTem.Option.OVERLAY	Enable overlay support	18
SYSTem.Option.REALTIME	Stall the simulator if faster than real processor	18
SYSTem.Option.TranslationSPACE	Identify user and hypervisor modes	19
SYSTem.Option.ZoneSPACES	Enable symbol management for zones	20
CPU specific MMU Commands		23
MMU.DUMP	Page wise display of MMU translation table	23
MMU.FORMAT	Define MMU table structure	25
MMU.List	Compact display of MMU translation table	30
MMU.SCAN	Load MMU table from CPU	32
TrOnchip Commands		34
TrOnchip.state	Display on-chip trigger window	34
TrOnchip.RESet	Set on-chip trigger to default state	34

Simulator for SuperH

Simulator for SuperH	(simulator_sh.pdf)	1
History		3
TRACE32 Simulator License		4
Quick Start of the Simulator		6
Peripheral Simulation		8
Troubleshooting		9
FAQ		9
CPU specific SYSTem Commands		10
SYSTem.CONFIG	Configure debugger according to target topology	10
SYSTem.CPU	CPU type	10
SYSTem.MemAccess	Select run-time memory access method	11
SYSTem.Mode	Establish the communication with the simulator	11
SYSTem.Option.HOOK	Compare PC to hook address	12
SYSTem.Option.IMASKASM	Disable interrupts while single stepping	12

SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	12
SYStem.Option.LittleEnd	Selection of little endian mode	13
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	13
SYStem.Option.VBR	Vector base address (SH3/4 only)	14
SYStem.RESetOut	CPU reset command	14
CPU specific TrOnchip Commands		15
TrOnchip.RESet	Set on-chip trigger to default state	15
TrOnchip.state	Display on-chip trigger window	15
CPU specific MMU Commands		16
MMU.DUMP	Page wise display of MMU translation table	16
MMU.List	Compact display of MMU translation table	18
MMU.SCAN	Load MMU table from CPU	20
Memory Classes		22

Simulator for TriCore

Simulator for TriCore	(simulator_tricore.pdf)	1
Introduction		5
Simulator Features		5
TRACE32 Simulator License		5
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		6
Quick Start of the Simulator		7
Peripheral Simulation		9
Debugging		10
Troubleshooting		10
Memory Classes		10
Breakpoints		11
Trace		12
FAQ		12
CPU specific SYStem Commands		13
SYStem.CONFIG	Configure debugger according to target topology	13
SYStem.CPU	Select CPU	13
SYStem.LOCK	Tristate the JTAG port	14
SYStem.MemAccess	Select run-time memory access method	15
SYStem.Mode	Establish the communication with the CPU	16
SYStem.Option	CPU specific commands	17
SYStem.Option.DCFREEZE	Do not invalidate cache	17
SYStem.Option.DUALPORT	Implicitly use run-time memory access	17
SYStem.Option.OVERLAY	Enable overlay support	17

SYStem.Option.ETK	Debugging together with ETK from ETAS	18
SYStem.Option.HeartBeat	Bug fix to avoid FPI bus conflict	18
SYStem.Option.ICFLUSH	Flush instruction cache at “Go” or “Step”	18
SYStem.Option.IMASKASM	Disable interrupts while single stepping	19
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	19
SYStem.Option.PERSTOP	Enable global peripheral suspend	19
SYStem.Option.SOFTLONG	Set 32 bit software breakpoints	19
SYStem.RESetOut	CPU reset command	19
SYStem.state	Open SYStem.state window	20
CPU specific TrOnchip Commands		21
TrOnchip	Onchip triggers	21

Simulator for V850

Simulator for V850	(simulator_v850.pdf)	1
Introduction		4
TRACE32 Simulator License		4
Brief Overview of Documents for New Users		4
Demo and Start-up Scripts		5
Quick Start of the Simulator		6
Peripheral Simulation		8
Troubleshooting		9
FAQ		9
CPU specific SYStem Commands		10
SYStem.CPU	CPU type selection	10
SYStem.LOCK	Lock and tristate the debug port	10
SYStem.MemAccess	Select run-time memory access method	10
SYStem.Mode	Establish the communication with the simulator	11
SYStem.CONFIG	Configure debugger according to target topology	12
SYStem.Option.IMASKASM	Mask interrupts during assembler step	12
SYStem.Option.IMASKHLL	Mask interrupts during HLL step	12
SYStem.RESetOut	CPU reset command	13
TrOnchip Commands		14
TrOnchip.state	Display on-chip trigger window	14
TrOnchip.RESet	Set on-chip trigger to default state	14
Memory Classes		15
State Analyzer		16
Keywords for the Display		16

Simulator for XTENSA	(simulator_xtensa.pdf)	1
History		4
Introduction		5
TRACE32 Simulator License		5
Quick Start of the Simulator		6
Peripheral Simulation		8
FAQ		8
CPU specific SYStem Commands		9
SYStem.CONFIG.state	Display target configuration	9
SYStem.CPU	Select the used CPU	9
SYStem.LOCK	Tristate the JTAG port	9
SYStem.MemAccess	Select run-time memory access method	9
SYStem.Mode	Establish the communication with the simulator	10
SYStem.Option.DisMode	Define disassembler mode	10
SYStem.Option.Endianness	Specify the byte ordering	11
SYStem.Option.IMASKASM	Disable interrupts while single stepping	11
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	11
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	12
SYStem.Option.SOFTLONG	Use 32-bit access to set breakpoint	13
SYStem.Option.RUNSTALLMASKASM	Disable "RunStall" while step	13
SYStem.Option.SnoopAddressPC	Program counter snoop address	13
SYStem.Option.SPILLLOC	Temporary memory	13
SYStem.Option.WinRegOption	Windowed register option	14
SYStem.TIE	TIE library files	15
SYStem.TIE.AddCoreLibrary	Add library file	15
SYStem.TIE.CMList	Instructions to display custom registers	15
SYStem.TIE.DELeTe	Remove all library files	16
SYStem.TIE.DISable	Unload and disable TIE instructions	16
SYStem.TIE.ENABLE	Load and enable TIE instructions	16
SYStem.TIE.GENper	Generate peripheral file	17
SYStem.TIE.GETArchOPTions	Detect architectural options from libraries	17
SYStem.TIE.ToolLibraryPath	Specify path for library tools	18
SYStem.TIE.RESet	Reset TIE	18
CPU specific TrOnchip Commands		19
TrOnchip.RESet	Reset on-chip trigger settings	19
TrOnchip.state	Display on-chip trigger window	19
CPU specific MMU Commands		20
MMU.DUMP	Page wise display of MMU translation table	20
MMU.List	Compact display of MMU translation table	22

Simulator for Z80+

Simulator for Z80+	(simulator_z80.pdf)	1
TRACE32 Simulator License		4
Quick Start of the Simulator		5
Peripheral Simulation		7
Troubleshooting		8
FAQ		8
Emulation Modes		9
SYStem.Mode	Establish the communication with the simulator	9
SYStem.CPU	CPU type	10
General SYStem Settings and Restrictions		11
SYStem.Option.BASE	Base address of internal registers	11
Using the MMU for Z180		12
Memory Classes		14

ICD In-Circuit Debugger

ICD Add-Ons

EPROM/FLASH Simulator

EPROM/FLASH Simulator	(eprom_simulator.pdf)	1
Introduction		4
Basics		4
Warning		4
Configuration		5
ICD Configuration for ROM Monitor		5
Mapping		7
Mapping the EPROM Simulator		7
Mapping the EPROM Simulator for BDM/ROM		8
Mapper Commands		11
Data Access		12
Break and Exception Control		13
Break		13

Count		14
Counter		14
Counter Commands		15
eXception		16
eXception.ICEINTPOL	Polarity of ICEINT line	16
eXception.NMIBREAK	Break through NMI	17
eXception.NMIDTR	Break through DTR line	17
eXception.NMIPOL	Polarity selection of NMI signal	17
eXception.NMIRTS	Break through RTS line	17
eXception.RESet	Default settings	17
eXception.RESetDTR	Reset through DTR line	18
eXception.RESetPOL	Polarity of RESET signal	18
eXception.RESetRTS	Reset through RTS line	18
eXception.view	Show exception settings	18
RESET		19
RESet	Initialize simulator	19
SYStem Commands		20
SYStem.Down	Deactivates simulator	20
SYStem.Mode	Selects operation mode	20
SYStem.Up	Activates simulator	21
SYStem.state	Shows operation mode	21
Store Settings		22
AutoSTOre	Autosave of settings	22
ClipSTOre	Store a setting to clipboard	23
STOre	Store a setting	23
Adapters		25
Adapter Configuration		25
Pinout Adapters		29
ESICON Adapter Function		38

TPU Debugger

TPU Debugger	(tpu.pdf)	1
TPU Basics		4
Entering TEST-Mode		5
TPU.BASE	Base address	5
TPU.SCAN	Scannig TPU	5
TPU.view	View TPU channels	6
TPU.Register.ALL	Register operation mode	6
TPU.Register.NEWSTEP	New debugging mode	7
TPU.Register.view	Register display	8

TPU.Register.Set	Register modification	9
TPU.Dump	Memory display	9
TPU.ListEntry	Table display	10
TPU.List	View microcode	11
TPU.Break	Break TPU	11
TPU.Go	Start TPU	12
TPU.SELect	Select TPU for debugging	12
TPU.Step	Single step TPU	13
TPU.RESet	Disable TPU debugger	13

Processor Architecture Manuals

78K

78K0R/RL78 Debugger	(debugger_78k.pdf)	1
Brief Overview of Documents for New Users		4
Demo and Start-up Scripts		4
Warning		5
General Notes/Target Design Requirements/Recommendations		6
General		6
Target Design Requirements		6
Limitations		6
Quick Start		7
Troubleshooting		9
Communication between Debugger and Processor can not be established		9
FAQ		9
78K0R/RL78 Specific Implementations		10
Breakpoints		10
Runtime Measurement		12
Memory Classes		12
CPU specific SYStem Commands		13
SYStem.CONFIG.state	Display target configuration	13
SYStem.CONFIG	Configure debugger according to target topology	14
SYStem.CPU	Select the used CPU	14
SYStem.DebugClock	Set debug clock frequency	15
SYStem.LOCK	Lock and tristate the debug port	15
SYStem.MemAccess	Select run-time memory access method	16
SYStem.Mode	Establish the communication with the target	17
SYStem.state	Display SYStem.state window	17
78K0R/RL78 specific SYStem Commands		18

SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	18
SYStem.Option.KEYCODE	Define 10 byte on-chip security ID	18
SYStem.Option.ResetMASK	Disable internal reset	18
SYStem.Option.SerialFreeze	Stops serial transmissions during break	19
SYStem.Option.TimerFreeze	Stops all internal timers during break	19
CPU specific TrOnchip Commands		20
Debug Connection		21

Andes

Andes Debugger	(debugger_andes.pdf)	1
History		4
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
Warning		6
Quick Start of the JTAG Debugger		7
Troubleshooting		9
Communication Between Debugger and Processor Can Not Be Established		9
FAQ		10
AndesCore Specific Implementations		11
Registers		11
Breakpoints		11
Runtime Measurement		14
Standby Mode		15
Memory Classes		16
Interruption Handling in Hardware		17
AndesCore specific SYStem Commands		24
SYStem.CONFIG	Configure debugger according to target topology	24
SYStem.CPU	Select the used CPU	27
SYStem.JtagClock	Define JTAG frequency	28
SYStem.LOCK	Tristate the JTAG port	29
SYStem.MemAccess	Select run-time memory access method	30
SYStem.Mode	Establish the communication with the target	30
SYStem.Option.ArchVersion	Configure version of architecture	32
SYStem.Option.ArchMcu	Configure MCU architecture	32
SYStem.Option.ArchRdreg	Configure reduced register set	33
SYStem.Option.DIMBR	Define base address of debug instruction memory	33
SYStem.Option.IMASKASM	Disable interrupts while single stepping	33
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	34

SYStem.Option.MMUSPACES	Separate address spaces by space IDs	34
SYStem.Option.SCRATCH	Define address for dummy reads	35
SYStem.Option.TURBO	Speed up memory access	35
SYStem.state	Display SYStem window	36
AndesCore Specific TrOnchip Commands		37
TrOnchip.ContextID	Enable context ID comparison	37
TrOnchip.RESet	Reset on-chip trigger settings	37
TrOnchip.StepVector	Halt on exception entry when single-stepping	37
TrOnchip.state	Display on-chip trigger window	38
CPU specific MMU Commands		39
MMU.DUMP	Page wise display of MMU translation table	39
MMU.List	Compact display of MMU translation table	41
MMU.SCAN	Load MMU table from CPU	43
JTAG Connection		45

APEX

APEX Debugger	(debugger_apex.pdf)	1
Introduction		4
Brief Overview of Documents for New Users		4
Warning		5
Quick Start of the JTAG Debugger		6
Troubleshooting		8
SYStem.Up Errors		8
FAQ		8
APEX Specific Implementations		9
Breakpoints		9
Runtime Measurement		10
Memory Classes		10
Bus Width Mapping		11
CPU specific SYStem Commands		12
SYStem.CONFIG.state	Display target configuration	12
SYStem.CONFIG	Configure debugger according to target topology	14
SYStem.CPU	Select the used CPU	27
SYStem.JtagClock	Define JTAG frequency	28
SYStem.LOCK	Tristate the JTAG port	29
SYStem.MemAccess	Select run-time memory access method	30
SYStem.Mode	Establish the communication with the target	31
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	31
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	32

SYStem.Option.AXICACHEFLAGS	Select AXI-AP CACHE bits	32
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	32
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	33
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	33
SYStem.Option.DAPREMAP	Rearrange DAP memory map	34
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	34
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	35
SYStem.Option.EnReset	Allow the debugger to drive nRESET/nSRST	36
SYStem.Option.IgnoreAttributes	ObeY ELF attributes for breakpoints	36
SYStem.Option.IMASKASM	Disable interrupts while single stepping	36
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	37
SYStem.Option.MEMoryMODEL	Select view of memory classes	38
SYStem.Option.MEMORYHPROT	Select memory-AP HPROT bits	39
SYStem.Option.ResBreak	Halt the core after reset	39
SYStem.Option.TRST	Allow debugger to drive TRST	40
SYStem.Option.WaitReset	Wait with JTAG activities after deasserting reset	40
SYStem.RESetOut	Assert nRESET/nSRST on JTAG connector	41
SYStem.state	Display SYStem.state window	41
APEX Specific TrOnchip Commands		42
TrOnchip.state	Display on-chip trigger window	42
TrOnchip.RESet	Set on-chip trigger to default state	42
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	42
Target Adaption		43
Interface Standards JTAG, Serial Wire Debug, cJTAG		43
Pinout		43

APS

APS Debugger	(debugger_aps.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
Warning		6
Quick Start		7
Troubleshooting		9
FAQ		10
CPU specific SYStem Settings		11
SYStem.CONFIG.state	Display target configuration	11
SYStem.CONFIG	Configure debugger according to target topology	12
SYStem.CPU	Select the used CPU	15
SYStem.JtagClock	Define JTAG clock	16

SYStem.MemAccess	Select run-time memory access method	16
SYStem.Mode	Establish the communication with the target	17
SYStem.LOCK	Lock and tristate the debug port	17
SYStem.Option.IMASKASM	Disable interrupts while single stepping	18
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	19
SYStem.Option.IntelSOC	Slave core is part of Intel® SoC	19
SYStem.Option.MonType	Selects monitor type	20
SYStem.Option.MonBase	Register base address	21
Breakpoints		23
Software breakpoints		23
On-chip breakpoints for instructions		23
Onchip Trace		24
Onchip.Mode	Type of trace records	24
CPU specific TrOnchip Commands		25
Memory Classes		26
JTAG Connector		27
JTAG Connector for ARM-like Designs		27
JTAG Connector for Atom-like Designs		28

ARC

ARC Debugger and Trace	(debugger_arc.pdf)	1
History		6
Introduction		7
Supported ARC Cores		7
Brief Overview of Documents for New Users		7
Demo and Start-up Scripts		9
Warning		10
Troubleshooting		11
SYStem.Up Errors		11
FAQ		11
Quick Start		12
CPU specific SETUP Command		16
SETUP.DIS	Disassembler configuration	16
CPU specific SYStem Commands		17
SYStem.CONFIG.state	Display target configuration	17
SYStem.CONFIG	Configure debugger according to target topology	19
SYStem.CPU	Select CPU type	36
SYStem.JtagClock	Select clock for JTAG communication	37

SYStem.LOCK	Lock and tristate the debug port	39
SYStem.MemAccess	Select run-time memory access method	39
SYStem.Mode	Select target reset mode	41
SYStem.Option	Set a target-specific option	43
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	43
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	43
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits	44
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	45
SYStem.Option.CorePowerDetection	Set methods to detect core power	45
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	47
SYStem.Option.DAPREMAP	Rearrange DAP memory map	48
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	48
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	49
SYStem.Option.DCFLUSH	Invalidate/flush data-cache for modified memory	49
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	49
SYStem.Option.detectOTrace	Disable auto-detection of on-chip trace	50
SYStem.Option.Endianness	Set the target endianness	51
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	51
SYStem.Option.HotBreakPoints	Set breakp. when CPU is running	52
SYStem.Option.ICFLUSH	Invalidate instruction-cache for modified memory	52
SYStem.Option.IMASKASM	Disable interrupts while single stepping	53
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	53
SYStem.Option.IntelSOC	Core is part of Intel® SoC	53
SYStem.Option.LimmBreakPoints	Software breakpoints with extra NOPs	54
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	54
SYStem.Option.OVERLAY	Enable overlay support	55
SYStem.Option.RegNames	Enable trivial names for core registers	56
SYStem.Option.PowerDetection	Choose method to detect the target power	56
SYStem.Option.ResetDetection	Choose method to detect a target reset	57
SYStem.Option.TIMEOUT	Define maximum time for core response	57
SYStem.Option.TRST	Allow debugger to drive TRST	58
SYStem.POWER	Control target power	58
SYStem.state	Show SYStem settings window	58
On-chip Breakpoints/Actionpoints		59
Using On-chip Breakpoints		59
Breakpoints in a ROM Area		59
Limitations		60
TrOnchip.CONVert	Allow extension of address range of breakpoint	61
TrOnchip.VarCONVert	Convert breakpoints on scalar variables	63
TrOnchip.OnchipBP	Number of on-chip breakpoints used by debugger	64
TrOnchip.RESet	Set on-chip trigger to default state	65
TrOnchip.state	Display on-chip trigger window	65
CPU specific MMU Commands		66

MMU.DUMP	Page wise display of MMU translation table	66
MMU.List	Compact display of MMU translation table	68
MMU.SCAN	Load MMU table from CPU	69
MMU.Init	Invalidate TLB entries	70
MMU.Set	Set an MMU TLB entry	70
CPU specific JTAG.CONFIG Commands		71
JTAG.CONFIG	Electrical characteristics of MIPI-60 debug signals	71
JTAG.CONFIG.DRiVer	Set slew rate of JTAG signals	71
JTAG.CONFIG.PowerDownTriState	Automatically tristate outputs	72
JTAG.CONFIG.TckRun	Free-running TCK mode	72
JTAG.CONFIG.TDOEdge	Select TCK edge	72
JTAG.CONFIG.Voltage.HookKThreshhold	Set hook threshold voltages	73
JTAG.CONFIG.Voltage.THreshhold	Set JTAG threshold voltages	73
JTAG.CONFIG.Voltage.REFerence	Voltage level of signals send to target	74
Trace specific NEXUS Commands		75
NEXUS.AuxTM	Enable auxiliary register trace messages	75
NEXUS.BTM	Enable program trace messaging	75
NEXUS.CLOCK	Clock to calculate time out of cycle count information	76
NEXUS.DataSuppress	Suppress data flow on likely FIFO overflow	76
NEXUS.DDR	Enable NEXUS double data rate mode	76
NEXUS.DSM	Enable core debug status messages	77
NEXUS.DTM	Enable data trace messages	77
NEXUS.FILTER	Configure the onchip trace filter resources	78
NEXUS.FILTER.ACompLimit	Trace address filters used by debugger	78
NEXUS.FILTER.DCompLimit	Number of trace data filter used by debugger	78
NEXUS.HISToryTHreshhold	Control the conditional history threshold	79
NEXUS.OFF	Switch the NEXUS trace port off	79
NEXUS.ON	Switch the NEXUS trace port on	79
NEXUS.PortMode	Set NEXUS trace port frequency	80
NEXUS.Register	Display NEXUS trace control registers	80
NEXUS.RegTM	Enable core register trace messages	80
NEXUS.RESet	Reset NEXUS settings	80
NEXUS.RTTBUILD	Define build configuration of used DesignWare trace	81
NEXUS.STALL	Stall program execution when FIFO full	81
NEXUS.state	Display NEXUS port configuration dialog	81
NEXUS.SyncFrame	Control SYNC frame insertion in ATB stream	81
NEXUS.TImeMode	Select method of time measurement	82
NEXUS.TimeStampCLOCK	Specify frequency of the global timestamp	83
NEXUS.TraceID	Set ID for CoreSight ATB stream	84
NEXUS.WTM	Enable watchpoint trace messages	84
Debug Connector Type and Pinout		85
Normal 20-Pin Connector		85

MIPI10 / MIPI20 / MIPI34 Connector	87
Converged MIPI60-Cv2 Connector	87
XDP Connector	87
Trace Connector Type and Pinout	88
Trace Signals	88
Normal Nexus Auxiliary Port (Mictor 38)	89
Dual Eight-bit Nexus Auxiliary Port (Mictor 38)	90
Out Offload and CoreSight TPIU	90

Arm/CORTEX/XSCALE

Arm Debugger	(debugger_arm.pdf)	1
History		9
Warning		10
Introduction		11
Brief Overview of Documents for New Users		11
Demo and Start-up Scripts		12
Quick Start of the JTAG Debugger		13
FAQ		14
Troubleshooting		15
Communication between Debugger and Processor cannot be established		15
Trace Extensions		16
Symmetric Multiprocessing		17
Arm Specific Implementations		18
TrustZone Technology		18
big.LITTLE		21
Breakpoints		23
Access Classes		34
Coprocessors		42
Accessing Memory at Run-time		45
Semihosting		49
Virtual Terminal		53
Large Physical Address Extension (LPAE)		54
Virtualization Extension, Hypervisor		55
Run-time Measurements		55
Trigger		55
Arm specific SYStem Commands		56
SYStem.CLOCK	Inform debugger about core clock	56
SYStem.CONFIG.state	Display target configuration	56
SYStem.CONFIG	Configure debugger according to target topology	57

SYStem.CONFIG.EXTWDTDIS	Disable external watchdog	105
SYStem.CONFIG.SMMU	Internal use	106
SYStem.CPU	Select the used CPU	108
SYStem.JtagClock	Define the frequency of the debug port	108
SYStem.LOCK	Tristate the JTAG port	111
SYStem.MemAccess	Select run-time memory access method	112
SYStem.Mode	Establish the communication with the target	118
SYStem.Option	Special setup	120
SYStem.Option.ABORTFIX	Do not access memory area from 0x0 to 0x1f	120
SYStem.Option.AMBA	Select AMBA bus mode	120
SYStem.Option.ASYNCBREAKFIX	Asynchronous break bugfix	121
SYStem.Option.BUGFIX	Breakpoint bug fix	121
SYStem.Option.BUGFIXV4	Asynch. break bug fix for ARM7TDMI-S REV4	122
SYStem.Option.BigEndian	Define byte order (endianness)	123
SYStem.Option.BOOTMODE	Define boot mode	123
SYStem.Option.CINV	Invalidate the cache after memory modification	124
SYStem.Option.CFLUSH	FLUSH the cache before step/go	124
SYStem.Option.CacheParam	Define external cache	124
SYStem.Option.CorePowerDetection	Set methods to detect core power	124
SYStem.Option.DACRBYPASS	Ignore DACR access permission settings	126
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	127
SYStem.Option.DAP2DBGPWRUPREQ	Force debug power in DAP2	127
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	128
SYStem.Option.DAP2SYSPWRUPREQ	Force system power in DAP2	129
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	130
SYStem.Option.DAPREMAP	Rearrange DAP memory map	130
SYStem.Option.DBGACK	DBGACK active on debugger memory accesses	130
SYStem.Option.DBGNOPWRDWN	DSCR bit 9 will be set in debug mode	131
SYStem.Option.DBGUNLOCK	Unlock debug register via OSLAR	131
SYStem.Option.DCDIRTY	Bugfix for erroneously cleared dirty bits	131
SYStem.Option.DCFREEZE	Disable data cache linefill in debug mode	132
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	132
SYStem.Option.DIAG	Activate more log messages	133
SYStem.Option.DisMode	Define disassembler mode	134
SYStem.Option.DynVector	Dynamic trap vector interpretation	135
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	135
SYStem.Option.ETBFIXMarvell	Read out on-chip trace data	135
SYStem.Option.ETMFIX	Shift data of ETM scan chain by one	136
SYStem.Option.ETMFIXWO	Bugfix for write-only ETM register	136
SYStem.Option.ETMFIX4	Use only every fourth ETM data package	136
SYStem.Option.EXEC	EXEC signal can be used by bustrace	136
SYStem.Option.EXTBYPASS	Switch off the fake TAP mechanism	137
SYStem.Option.FASTBREAKDETECTION	Fast core halt detection	137

SYStem.Option.HRCWOVerRide	Enable override mechanism	137
SYStem.Option.ICEBreakerETMFiXMarvell	Lock on-chip breakpoints	138
SYStem.Option.ICEPiCK	Enable/disable assertions and wait-in-reset	138
SYStem.Option.IMASKASM	Disable interrupts while single stepping	138
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	139
SYStem.Option.INTDiS	Disable all interrupts	139
SYStem.Option.IRQBReAKFiX	Break bugfix by using IRQ	139
SYStem.Option.KEYCODE	Define key code to unsecure processor	140
SYStem.Option.L2Cache	L2 cache used	140
SYStem.Option.L2CacheBase	Define base address of L2 cache register	140
SYStem.Option.LOCKRES	Go to 'Test-Logic Reset' when locked	141
SYStem.Option.MACHINESPACES	Address extension for guest OSes	142
SYStem.Option.MDMAP	Set debug option controlled by NXP MDM-AP	143
SYStem.Option.MemStatusCheck	Check status bits during memory access	144
SYStem.Option.MMUPhysLogMemaccess	Memory access preferences	144
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	145
SYStem.Option.MonitorHoldoffTime	Delay between monitor accesses	146
SYStem.Option.MPUByPASS	Ignore MPU access permission settings	146
SYStem.Option.MultipleSiX	No multiple loads/stores	146
SYStem.Option.NOData	No data connected to the trace	146
SYStem.Option.NOIRCHECK	No JTAG instruction register check	147
SYStem.Option.NoPRCRReset	Do not cause reset by PRCR	147
SYStem.Option.NoRunCheck	No check of the running state	147
SYStem.Option.NoSecureFix	Do not switch to secure mode	148
SYStem.Option.OVERLAY	Enable overlay support	149
SYStem.Option.PALLADIUM	Extend debugger timeout	149
SYStem.Option.PC	Define address for dummy fetches	150
SYStem.Option.ProgramAccessFix	Program memory access bug fix	150
SYStem.Option.PROTECTION	Sends an unsecure sequence to the core	150
SYStem.Option.PWRCHECK	Check power and clock	151
SYStem.Option.PWRCHECKFiX	Check power and clock	151
SYStem.Option.PWRDWN	Allow power-down mode	151
SYStem.Option.PWRDWNRecover	Mode to handle special power recovery	152
SYStem.Option.PWRDWNRecoverTimeOut	Timeout for power recovery	152
SYStem.Option.PWROVR	Specifies power override bit	152
SYStem.Option.ResBreak	Halt the core after reset	153
SYStem.Option.ResetDetection	Choose method to detect a target reset	154
SYStem.Option.RESeTREGiSter	Generic software reset	154
SYStem.Option.RESTARTFiX	Wait after core restart	155
SYStem.Option.RisingTDO	Target outputs TDO on rising edge	155
SYStem.Option.ShowError	Show data abort errors	156
SYStem.Option.SLaVeSOFTReSeT	Allow soft reset of slave cores	156
SYStem.Option.SOFTLONG	Use 32-bit access to set breakpoint	156

SYStem.Option.SOFTQUAD	Use 64-bit access to set breakpoint	157
SYStem.Option.SOFTWORD	Use 16-bit access to set breakpoint	157
SYStem.Option.SPLIT	Access memory depending on CPSR	157
SYStem.Option.StandByTraceDelaytime	Trace activation after reset	158
SYStem.Option.STEPSOFT	Use software breakpoints for ASM stepping	158
SYStem.Option.SYSPWRUPREQ	Force system power	158
SYStem.Option.TIDBGEN	Activate initialization for TI derivatives	159
SYStem.Option.TIETMFX	Bug fix for customer specific ASIC	159
SYStem.Option.TIDEMUXFIX	Bug fix for customer specific ASIC	159
SYStem.Option.TraceStrobe	Deprecated command	159
SYStem.Option.TRST	Allow debugger to drive TRST	160
SYStem.Option.TURBO	Speed up memory access	160
SYStem.Option.WaitIDCODE	IDCODE polling after deasserting reset	161
SYStem.Option.WaitReset	Wait with JTAG activities after deasserting reset	162
SYStem.Option.WATCHDOG	Disable watchdog while debugging	163
SYStem.Option.ZoneSPACES	Enable symbol management for Arm zones	164
SYStem.Option.ZYNQJTAGINDEPENDENT	Configure JTAG cascading	170
SYStem.RESetOut	Assert nRESET/nSRST on JTAG connector	170
SYStem.state	Display SYStem window	171
Arm specific Functions		172
SYStem.Option.HRCWOVerRide()		172
Arm Specific Benchmarking Commands		173
BMC.EXPORT	Export benchmarking events from event bus	173
BMC.EXTEND	Define benchmark counter event	174
BMC.MODE	Define the operating mode of the benchmark counter	175
BMC.<counter>.EVENT	Configure the performance monitor	176
BMC.PRESCALER	Prescale the measured cycles	179
BMC.TARA	Calibrate the benchmark counter	179
Arm Specific TrOnchip Commands		180
TrOnchip.A	Programming the ICE breaker module	180
TrOnchip.A.Value	Define data selector	181
TrOnchip.A.Size	Define access size for data selector	181
TrOnchip.A.CYcle	Define access type	182
TrOnchip.A.Address	Define address selector	183
TrOnchip.A.Trans	Define access mode	184
TrOnchip.A.Extern	Define the use of EXTERN lines	184
TrOnchip.AddressMask	Define an address mask	185
TrOnchip.ContextID	Enable context ID comparison	185
TrOnchip.CONVert	Allow extension of address range of breakpoint	186
TrOnchip.MachineID	Extend on-chip breakpoint/trace filter by machine ID	187
TrOnchip.MatchASID	Extend on-chip breakpoint/trace filter by ASID	188
TrOnchip.MatchMachine	Extend on-chip breakpoint/trace filter by machine	188

TrOnchip.MatchZone	Extend on-chip breakpoint/trace filter by zone	189
TrOnchip.Mode	Configure unit A and B	190
TrOnchip.RESet	Reset on-chip trigger settings	190
TrOnchip.Set	Set bits in the vector catch register	191
TrOnchip.StepVector	Step into exception handler	191
TrOnchip.StepVectorResume	Catch exceptions and resume single step	192
TrOnchip.TEnable	Define address selector for bus trace	193
TrOnchip.TCYcle	Define cycle type for bus trace	194
TrOnchip.VarCONVert	Convert breakpoints on scalar variables	195
TrOnchip.state	Display on-chip trigger window	196
CPU specific MMU Commands		197
MMU.DUMP	Page wise display of MMU translation table	197
MMU.List	Compact display of MMU translation table	201
MMU.SCAN	Load MMU table from CPU	204
CPU specific SMMU Commands		206
SMMU	Hardware system MMU (SMMU)	206
SMMU.ADD	Define a new hardware system MMU	216
SMMU.Clear	Delete an SMMU	218
SMMU.CtxtDescTable	List a context descriptor table	218
SMMU.DumpQueue.<queue>	Dump entries of a queue	219
SMMU.DumpQueue.CMD	Dump cmd queue entries	221
SMMU.DumpQueue.Event	Dump event queue entries	222
SMMU.Register	Peripheral registers of an SMMU	223
SMMU.Register.ContextBank	Display registers of context bank	224
SMMU.Register.Global	Display global registers of SMMU	225
SMMU.Register.MMUregs	Display MMU specific registers	225
SMMU.Register.S1Context	Display stage 1 context descriptor registers	226
SMMU.Register.StreamTblEntry	Display stream table entry registers	226
SMMU.Register.StreamMapRegGrp	Display registers of an SMRG	227
SMMU.RESet	Delete all SMMU definitions	228
SMMU.SSDtable	Display security state determination table	229
SMMU.StreamMapRegGrp	Access to stream map table entries	230
SMMU.StreamMapRegGrp.ContextReg	Display context bank registers	231
SMMU.StreamMapRegGrp.Dump	Page-wise display of SMMU page table	233
SMMU.StreamMapRegGrp.list	List page table entries	235
SMMU.StreamTable	Display a stream table	236
SMMU.StreamTblEntry	Access to a stream table entry	248
SMMU.StreamTblEntry.Dump	Page-wise display of SMMU page table	250
SMMU.StreamTblEntry.list	List page table entries	251
SMMU.StreamTblEntry.Register	Display STE or CD registers	252
Target Adaption		253
Probe Cables		253

Interface Standards JTAG, Serial Wire Debug, cJTAG	253	
Connector Type and Pinout	253	
Armv8 and Armv9 Debugger(debugger_armv8v9.pdf)	1	
History	8	
Warning	10	
Introduction	11	
Brief Overview of Documents for New Users	11	
Demo and Start-up Scripts	12	
Quick Start of the JTAG Debugger	13	
Configure Debugger for SoC Specific Reset Behavior	17	
Troubleshooting	31	
Communication between Debugger and Processor cannot be established	31	
FAQ	32	
Trace Extensions	32	
Quick Start for Multicore Debugging	33	
SMP Debugging - Quick Start	33	
AMP Debugging - Quick Start	37	
Arm Specific Implementations	40	
AArch Mode Support	40	
TrustZone Technology	43	
big.LITTLE	47	
Breakpoints	49	
Access Classes	58	
System Registers (AArch64 Mode)	66	
Coprocessors (AArch32 Mode)	69	
Accessing Memory at Run-time	74	
Semihosting	78	
Virtual Terminal	84	
Large Physical Address Extension (LPAE)	85	
Virtualization Extension, Hypervisor	86	
Debug Field	87	
Run-time Measurements	88	
Trigger	88	
Arm specific SYStem Commands	89	
SYStem.CLOCK	Inform debugger about core clock	89
SYStem.CONFIG.state	Display target configuration	89
SYStem.CONFIG	Configure debugger according to target topology	90
SYStem.CONFIG.BMCSnapshot.Base	Synchronous BMC sampling	138
SYStem.CONFIG.EXTWDTDIS	Disable external watchdog	138
SYStem.CONFIG.GICD	Generic Interrupt Controller Distributor (GIC)	139

SYStem.CONFIG.GICR	Generic Interrupt Controller Redistributor	142
SYStem.CONFIG.GICC	Generic Interrupt Controller physical CPU interface	143
SYStem.CONFIG.GICH	Generic Interrupt Controller virtual interface control	144
SYStem.CONFIG.GICV	Generic Interrupt Controller virtual CPU interface	145
SYStem.CONFIG.SMMU	Internal use	146
SYStem.CPU	Select the used CPU	147
SYStem.JtagClock	Define the frequency of the debug port	148
SYStem.LOCK	Tristate the JTAG port	149
SYStem.MemAccess	Select run-time memory access method	150
SYStem.Mode	Establish the communication with the target	153
SYStem.Option	Special setup	155
SYStem.Option.Address32	Define address format display	155
SYStem.Option.AXI32	Use 32-bit atomic AXI accesses instead of 64-bit	156
SYStem.Option.BreakOS	Allow break during OS-unlock	156
SYStem.Option.CacheStatusCheck	Check status bits during cache access	157
SYStem.Option.CFLUSH	FLUSH the cache before step/go	157
SYStem.Option.CLTAKEY	Set key values for CLTAP operation	158
SYStem.Option.CoreSightRESet	Assert CPU reset via CTRL/STAT	158
SYStem.Option.CTIGate	CTI gate control	158
SYStem.Option.CTITimerStop	Stop system timer when CPU stops	159
SYStem.Option.DACRBYPASS	Ignore DACR access permission settings	159
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	160
SYStem.Option.DAP2DBGPWRUPREQ	Force debug power in DAP2	160
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	161
SYStem.Option.DAPREMAP	Rearrange DAP memory map	162
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	162
SYStem.Option.DAP2SYSPWRUPREQ	Force system power in DAP2	163
SYStem.Option.DBGCLAIM	Debug and PMU claim	164
SYStem.Option.DBGSPR	Use debugger view for SPR access	164
SYStem.Option.DBGUNLOCK	Unlock debug register via OSLAR	165
SYStem.Option.DCacheMaintenance	Data cache maintenance strategy	165
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	166
SYStem.Option.DIAG	Activate more log messages	167
SYStem.Option.DUALPORT	Implicitly use run-time memory access	167
SYStem.Option.DisMode	Define disassembler mode	167
SYStem.Option.EDACR	Define 32-bit value written to EDACR register	168
SYStem.Option.ENFORCECPSWITCH	Try AArch32 for C1x access	168
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	169
SYStem.Option.FunctionalRESet	Custom functional reset	169
SYStem.Option.HRCWOVerRide	Enable override mechanism	169
SYStem.Option.ICacheMaintenance	I-Cache maintenance strategy	170
SYStem.Option.IMASKASM	Disable interrupts while single stepping	170
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	171

SYStem.Option.INTDIS	Disable all interrupts	171
SYStem.Option.IntelSOC	Slave core is part of Intel® SoC	171
SYStem.Option.KEYCODE	Define key code to unsecure processor	172
SYStem.Option.MACHINESPACES	Address extension for guest OSes	172
SYStem.Option.MDMAP	Set debug option controlled by NXP MDM-AP	173
SYStem.Option.MemStatusCheck	Check status bits during memory access	174
SYStem.Option.MMUPhysLogMemaccess	Memory access preferences	175
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	175
SYStem.Option.MPUBYPASS	Ignore MPU access permission settings	176
SYStem.Option.NOMA	Use alternative memory access	177
SYStem.Option.NoPRCRReset	Disable warm reset via PRCR	177
SYStem.Option.OSUnlockCatch	Use the 'OS Unlock Catch' debug event	178
SYStem.Option.OVERLAY	Enable overlay support	178
SYStem.Option.PALLADIUM	Extend debugger timeout	179
SYStem.Option.PWRDWN	Allow power-down mode	179
SYStem.Option.PAN	Overwrite CPSR.PAN setting	179
SYStem.Option.PWRREQ	Request core power	180
SYStem.Option.ResBreak	Halt the core after reset	181
SYStem.Option.ResetDetection	Choose method to detect a target reset	182
SYStem.RESetOut	Assert nRESET/nSRST on JTAG connector	182
SYStem.Option.RESetREGister	Generic software reset	183
SYStem.Option.RisingTDO	Target outputs TDO on rising edge	183
SYStem.Option.SLaVeSOFTRESet	Allow soft reset of slave cores	184
SYStem.Option.SMPMultipleCall	Send start event to each SMP core	184
SYStem.Option.SOFTLONG	Use 32-bit access to set breakpoint	184
SYStem.Option.SOFTQUAD	Use 64-bit access to set breakpoint	185
SYStem.Option.STEPSOFT	Use software breakpoints for ASM stepping	185
SYStem.Option.SOFTWORD	Use 16-bit access to set breakpoint	185
SYStem.Option.TraceFilterOverride	Enable/Disable trace filter override	185
SYStem.Option.TURBO	Disable cache maintenance during memory access	186
SYStem.state	Display SYStem window	186
SYStem.Option.SoftLockUNLOCK	Unlock software lock via EDLAR	187
SYStem.Option.SYSPWRUPREQ	Force system power	187
SYStem.Option.TRST	Allow debugger to drive TRST	187
SYStem.Option.WaitCTIREG	Wait for CTI registers after reset	188
SYStem.Option.WaitDAPPWR	Wait for DAP power after DAP power request	188
SYStem.Option.WaitDBGREG	Wait for core debug registers after reset	189
SYStem.Option.WaitIDCODE	IDCODE polling after deasserting reset	190
SYStem.Option.WaitReset	Wait with JTAG activities after deasserting reset	191
SYStem.Option.ZoneSPACES	Enable symbol management for Arm zones	193
SYStem.Option.ZYNQJTAGINDEPENDENT	Configure JTAG cascading	199
Arm specific Functions		200
STATE.NOCTIACCESS()		200

STATE.NOCPUACCESS()		200
SYStem.Option.HRCWOVerRide()		201
Arm specific Benchmarking Commands		202
BMC.<counter>.CountEL<x>	Select exception level events to be counted	202
BMC.EXPORT	Export benchmarking events from event bus	204
BMC.LongCycle	Configure cycle counter width	205
BMC.PRESCALER	Prescale the measured cycles	205
Arm specific TrOnchip Commands		206
TrOnchip.ContextID	Enable context ID comparison	207
TrOnchip.CONVert	Allow extension of address range of breakpoint	208
TrOnchip.MachineID	Extend on-chip breakpoint/trace filter by machine ID	209
TrOnchip.MatchASID	Extend on-chip breakpoint/trace filter by ASID	210
TrOnchip.MatchMachine	Extend on-chip breakpoint/trace filter by machine	210
TrOnchip.MatchZone	Extend on-chip breakpoint/trace filter by zone	211
TrOnchip.RESERVE	Exclude breakpoint or watchpoint from debugger usage	212
TrOnchip.RESet	Set on-chip trigger to default state	212
TrOnchip.Set	Set bits in the vector catch register	213
TrOnchip.StepVector	Step into exception handler	215
TrOnchip.StepVectorResume	Catch exceptions and resume single step	216
TrOnchip.VarCONVert	Convert breakpoints on scalar variables	217
TrOnchip.state	Display on-chip trigger window	218
Cache Analysis and Maintenance		219
TRACE32 Cache Support by CPU Type		220
CPU specific MMU Commands		223
MMU.DUMP	Page wise display of MMU translation table	223
MMU.List	Compact display of MMU translation table	236
MMU.SCAN	Load MMU table from CPU	238
TRACE32 TLB Support by CPU Type		240
CPU specific SMMU Commands		243
SMMU	Hardware system MMU (SMMU)	243
SMMU.ADD	Define a new hardware system MMU	253
SMMU.Clear	Delete an SMMU	255
SMMU.CtxtDescTable	List a context descriptor table	255
SMMU.DumpQueue.<queue>	Dump entries of a queue	256
SMMU.DumpQueue.CMD	Dump cmd queue entries	258
SMMU.DumpQueue.Event	Dump event queue entries	259
SMMU.Register	Peripheral registers of an SMMU	260
SMMU.Register.ContextBank	Display registers of context bank	261
SMMU.Register.Global	Display global registers of SMMU	262
SMMU.Register.MMUregs	Display MMU specific registers	262
SMMU.Register.S1Context	Display stage 1 context descriptor registers	263
SMMU.Register.StreamTblEntry	Display stream table entry registers	263

SMMU.Register.StreamMapRegGrp	Display registers of an SMRG	264
SMMU.RESet	Delete all SMMU definitions	265
SMMU.SSDtable	Display security state determination table	266
SMMU.StreamMapRegGrp	Access to stream map table entries	267
SMMU.StreamMapRegGrp.ContextReg	Display context bank registers	268
SMMU.StreamMapRegGrp.Dump	Page-wise display of SMMU page table	270
SMMU.StreamMapRegGrp.list	List page table entries	272
SMMU.StreamTable	Display a stream table	273
SMMU.StreamTblEntry	Access to a stream table entry	285
SMMU.StreamTblEntry.Dump	Page-wise display of SMMU page table	287
SMMU.StreamTblEntry.list	List page table entries	288
SMMU.StreamTblEntry.Register	Display STE or CD registers	289
Target Adaption		290
Probe Cables		290
Interface Standards JTAG, Serial Wire Debug, cJTAG		290
Connector Type and Pinout		291
Cortex-M Debugger (debugger_cortexm.pdf)		1
History		6
Warning		7
Introduction		8
Brief Overview of Documents for New Users		8
Demo and Start-up Scripts		9
Products for Debugging and Tracing Cortex-M Cores		10
PowerDebug and Debug Cable		10
µTrace (MicroTrace) (with MIPI20T-HS Whisker)		11
PowerDebug and CombiProbe (with MIPI20T-HS Whisker)		12
PowerDebug and CombiProbe (with CombiProbe MIPI34 Whisker)		13
PowerDebug and PowerTrace (X-License)		14
Quick Start of the JTAG Debugger		15
Troubleshooting		17
Communication between Debugger and Processor Cannot Be Established		17
FAQ		18
Trace Extensions		19
Cortex-M Specific Implementations		20
Breakpoints		20
Access Classes		25
BenchMarkCounter		32
Semihosting		34
Virtual Terminal		35
Runtime Measurement		35

Trigger	36
Micro Trace Buffer (MTB) for Cortex-M0+	36
Cortex-M specific Onchip Commands	37
Onchip.Mode.RAMPRIV	SRAM privilege access 37
Onchip.Mode.SFRWPRIV	Special function register write access 37
Onchip.Mode.TSTARTEN	Enable TSTART signal 37
Onchip.Mode.TSTOPEN	Enable TSTOP signal 37
Onchip.TBADDRESS	Base address of the trace buffer 38
Cortex-M specific SYStem Commands	39
SYStem.CLOCK	Inform debugger about core clock 39
SYStem.CONFIG.state	Display target configuration 39
SYStem.CONFIG	Configure debugger according to target topology 40
SYStem.CONFIG.EXTWDTDIS	Disable external watchdog 89
SYStem.CPU	Select the used CPU 89
SYStem.JtagClock	Define the frequency of the debug port 90
SYStem.LOCK	Tristate the JTAG port 92
SYStem.MemAccess	Select run-time memory access method 93
SYStem.Mode	Establish the communication with the target 96
SYStem.Option	Special setup 98
SYStem.Option.BigEndian	Define byte order (endianness) 98
SYStem.Option.CFLUSHAFTERBREAK	Flush data cache after break 98
SYStem.Option.CLEARHARDFULT	Handle the HFSR[FORCED] bit 98
SYStem.Option.CoreSightRESet	Assert CPU reset via CTRL/STAT 99
SYStem.Option.CORTEXMAHB	AHB-AP type of the Cortex-M 99
SYStem.Option.CypressACQuire	Send acquire sequence 99
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP 100
SYStem.Option.DAP2DBGPWRUPREQ	Force debug power in DAP2 100
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP 101
SYStem.Option.DAP2SYSPWRUPREQ	Force system power in DAP2 102
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check 103
SYStem.Option.DAPREMAP	Rearrange DAP memory map 103
SYStem.Option.DEBUGPORTOptions	Options for debug port handling 103
SYStem.Option.DIAG	Activate more log messages 104
SYStem.Option.DisMode	Define disassembler mode 105
SYStem.Option.DUALPORT	Implicitly use run-time memory access 105
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST) 106
SYStem.Option.FORCESECure	Force secure memory access 106
SYStem.Option.IMASKASM	Disable interrupts while single stepping 106
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 107
SYStem.Option.INTDIS	Disable all interrupts 107
SYStem.Option.IntelSOC	Slave core is part of Intel® SoC 107
SYStem.Option.LOCKRES	Go to 'Test-Logic Reset' when locked 108
SYStem.Option.MDMAP	Set debug option controlled by NXP MDM-AP 108

SYStem.Option.MMUSPACES	Enable space IDs	110
SYStem.Option.NoRunCheck	No check of the running state	111
SYStem.Option.OVERLAY	Enable overlay support	111
SYStem.Option.PALLADIUM	Extend debugger timeout	112
SYStem.Option.PSOCswdACQuire	Debug port acquire for PSOC5	112
SYStem.Option.PWRDWNRecover	Mode to handle special power recovery	112
SYStem.Option.ResBreak	Halt the core after reset	113
SYStem.Option.RESetREGister	Generic software reset	114
SYStem.Option.RisingTDO	Target outputs TDO on rising edge	114
SYStem.Option.SELECTDAP	Select Cortex-M DAP	115
SYStem.Option.SOFTLONG	Use 32-bit access to set breakpoint	115
SYStem.Option.SOFTWORD	Use 16-bit access to set breakpoint	115
SYStem.Option.STEPSOFT	Use software breakpoints for ASM stepping	115
SYStem.Option.SYSPWRUPREQ	Force system power	116
SYStem.Option.SYSRESETREQ	Allow system reset via the AIRC register	116
SYStem.Option.TRST	Allow debugger to drive TRST	116
SYStem.Option.VECTRESET	Allow local reset via the AIRC register	117
SYStem.Option.WaitIDCODE	IDCODE polling after deasserting reset	117
SYStem.Option.WaitReset	Wait with JTAG activities after deasserting reset	118
SYStem.Option.WakeUpACKnowledge	Set acknowledge after wake-up	119
SYStem.RESetOut	Performs a reset	119
SYStem.state	Display SYStem.state window	119
ARM Specific Benchmarking Commands		120
BMC.OFF	Disable benchmark counters	120
BMC.ON	Enable benchmark counters	120
BMC.SELect	Select counter for statistic analysis	120
BMC.Trace	Activate BMC trace	121
ARM specific TrOnchip Commands		122
TrOnchip.state	Display on-chip trigger window	122
TrOnchip.MatchASID	Extend on-chip breakpoint/trace filter by ASID	123
TrOnchip.CONVert	Allow extension of address range of breakpoint	124
TrOnchip.RESERVE	Reserve on-chip breakpoint comparators	125
TrOnchip.RESet	Reset on-chip trigger settings	125
TrOnchip.Set	Set bits in the vector catch register	126
TrOnchip.StepVector	Step into exception handler	127
TrOnchip.StepVectorResume	Catch exceptions and resume single step	127
TrOnchip.VarCONVert	Convert breakpoints on scalar variables	128
JTAG Connection		130
CombiProbe for Cortex-M User's Guide	(combiprobe_cortexm.pdf)	1
TRACE32 Products for Cortex-M		4
PowerDebug and CombiProbe (with MIPI20T-HS Whisker)		5
PowerDebug and CombiProbe (with CombiProbe MIPI34 Whisker)		6

PowerDebug and Debug Cable	7
PowerDebug and PowerTrace (X-License)	8
µTrace (MicroTrace) (with MIPI20T-HS Whisker)	9
Basics	10
Keywords	10
Technical Details of the CombiProbe 2	10
Overview of Cortex-M CoreSight Components	11
Embedded Trace Macrocell (ETM) Overview	11
Data Watchpoint and Trace (DWT) Unit Overview	12
Instrumentation Trace Macrocell (ITM) Overview	12
Trace Port Interface Unit (TPIU) Overview	13
Tool Support for the Cortex-M3	14
20-Pin Debug and Trace Connector	14
Serial Wire Debug Port (SWDP)	15
Lauterbach's CombiProbe	16
Setting up Parallel Trace	17
Configuring the Correct Port Type	17
Connecting to the Target and Configure Trace-related Components	17
Configuring TRACE32 Trace Settings	18
Using the ETM	23
Using the DWT	25
PCSampler	26
Interrupt Trace	28
Tracing Data Accesses	30
Cycle Accurate Trace	35
Merging ETM and DWT Data	36
Performance Analysis with the DWT Counters	38
Serial Wire Debug Port (SWDP) and Serial Wire Viewer (SWV)	40
Software Trace with the ITM	42
Custom Trace DLLs	44
On-the-fly Transfer of ITM and ETM Data	46
Extending the Recording Size	47
Feeding Your Own Applications with Trace Data	48
Real-Time Profiling with the ETM	49
FAQ	50
MicroTrace for Cortex-M User's Guide (microtrace_cortexm.pdf)	1
TRACE32 Products for Cortex-M	4
µTrace (MicroTrace) (with MIPI20T-HS Whisker)	5
PowerDebug and CombiProbe (with MIPI20T-HS Whisker)	6

PowerDebug and CombiProbe (with CombiProbe MIPI34 Whisker)	7
PowerDebug and Debug Cable	8
PowerDebug and PowerTrace (X-License)	9
Basics	10
Keywords	10
CoreSight Components	10
Overview of Cortex-M CoreSight Components	11
Embedded Trace Macrocell (ETM) Overview	11
Data Watchpoint and Trace (DWT) Unit Overview	12
Instrumentation Trace Macrocell (ITM) Overview	12
Trace Port Interface Unit (TPIU) Overview	13
Embedded Trace Buffer	13
Connectors	14
Setting up Parallel Trace	15
Configuring the Correct Port Type	15
Connecting to the Target and Configure Trace-related Components	15
Configuring TRACE32 Trace Settings	16
Using the ETM	21
Using the DWT	23
PCSampler	24
Interrupt Trace	26
Tracing Data Accesses	28
Cycle Accurate Trace	33
Merging ETM and DWT Data	34
Performance Analysis with the DWT Counters	36
Serial Wire Debug Port (SWDP) and Serial Wire Viewer (SWV)	38
Software Trace with the ITM	40
Custom Trace DLLs	42
On-the-fly Transfer of ITM and ETM Data	44
Extending the Recording Size	45
Feeding Your Own Applications with Trace Data	46
Real-Time Profiling with the ETM	47
Discontinued Products	48
μTrace (MicroTrace) (with CombiProbe MIPI34 Whisker)	48
Deprecated Connectors	49
Arm ETM Trace (trace_arm_etm.pdf)	1
History	6
Installation	6

Software Installation	6	
Recommendation for Starting the Software	6	
Recommendation for Power Down	7	
Hardware Installation	7	
Utilization of the ETM	22	
Startup Script	22	
Displaying Trace Results	27	
Programmer's Model of the ETM	30	
ETM Commands	33	
ETM	Embedded Trace Macrocell (ETM)	33
ETM.AbsoluteTimestamp	Absolute cyclecount packets	34
ETM.AddressMunging	Dig endian address munging	34
ETM.ATBTrigger	Use ATB to transfer trace trigger to trace sink	35
ETM.AUXCTLR	Set ETMv4 implementation-specific auxiliary control register	38
ETM.BBC	Branch address broadcast	38
ETM.BBCExclude	Exclude address ranges from branch-broadcasting	39
ETM.BBCInclude	Enable branch-broadcasting for dedicated address ranges	39
ETM.CLEAR	Clear sequencer settings	40
ETM.CLOCK	Set core clock frequency for timing measurements	40
ETM.CORE	Select core for ETM	41
ETM.CPRT	Monitor coprocessor register transfers	41
ETM.COND	Conditional non-branch instructions	42
ETM.ContextID	Select the width of the 'ContextID' register	42
ETM.CycleAccurate	Cycle accurate tracing	43
ETM.CycleCountThreshold	Set granularity for cycle accurate timing info	44
ETM.CycleCountTickEnable	ETMv4 cycle counter overflows	44
ETM.CycleCountTickRate	ETMv4 cycle counter rate	44
ETM.DataSuppress	Suppress data flow to prevent FIFO overflow	45
ETM.DataTrace	Configure data-trace	46
ETM.DataTracePrestore	Show program trace cycle for data trace cycle	48
ETM.DataViewExclude	Suppress data trace for specified address range	49
ETM.DataViewInclude	Restrict broadcast of data accesses to range	50
ETM.DBGRQ	Debug request control	51
ETM.FifoFullExclude	No activation of FIFOFULL in range	51
ETM.FifoFullInclude	FIFOFULL only in range	52
ETM.FifoLevel	Define FIFO level for FIFOFULL	52
ETM.FunnelHoldTime	Define minimum funnel hold time	53
ETM.HalfRate	Halfrate mode	53
ETM.LPOVERRIDE	Prohibit lower power mode	53
ETM.INSTP0	Load and store instructions	54
ETM.MapDecode	Memory map decode control	54
ETM.NoOverflow	Enable ETMv4 feature to prevent target FiFo overflows	55
ETM.ON	Switch ETM on	55

ETM.OFF	Switch ETM off	55
ETM.PortDisable	Force trace-port enable signal to zero	56
ETM.PortDisableOnchip	Disable ETM trace port when ETB is used	57
ETM.PortMode	Select ETM mode	58
ETM.PortRoute	Set up trace hardware	59
ETM.PortSize	Define trace port width	59
ETM.PowerUpRequest	Power-up request for the ETM by the debugger	60
ETM.PseudoDataTrace	Enable pseudo data trace detection	60
ETM.QE	Enable Q elements	60
ETM.QTraceExclude	Prohibit Q trace elements in given address range	62
ETM.QTraceInclude	Allow Q trace elements in given address range	62
ETM.RefClock	Enable STP reference clock	63
ETM.Register	Display the ETM registers	64
ETM.RESet	Reset ETM settings	65
ETM.ReserveContextID	Reserve special values used with context ID	65
ETM.ReturnStack	Enable return stack tracing mode	66
ETM.Set	Precise control of ETM trigger events	67
ETM.SmartTrace	Configure smart trace	75
ETM.STALL	Stall processor to prevent FIFO overflow	75
ETM.state	Display ETM settings	76
ETM.StoppingBreakPoints	Use ETM comparators for breakpoints	77
ETM.SyncPeriod	Set synchronization frequency	80
ETM.TimeMode	Improve ETM/PTM timestamp information	81
ETM.TimeStampCLOCK	Specify frequency of the global timestamp	86
ETM.TimeStamps	Control for global timestamp packets	86
ETM.TimeStampsTrace	Specify data trace correlation method (ETMv4)	87
ETM.Trace	Control generation of trace information	87
ETM.TraceCORE	Core specific default tracing	88
ETM.TraceDataPriority	Define data trace priority	88
ETM.TraceERRor	Force ETM to emit all system error exceptions	89
ETM.TraceExclude	Suppress program trace for specified address range	90
ETM.TraceID	Change the default ID for an ETM trace source	91
ETM.TraceInclude	Restrict program trace to specified address range	91
ETM.TraceNoPCREL	No data trace for accesses relative to program counter	92
ETM.TraceNoSPREL	No data trace for accesses relative to stack pointer	92
ETM.TracePriority	Define priority of ETM	93
ETM.TraceRESet	Forces the ETM to emit all core resets	93
ETM.TRCIDR	Define TRCIDR register values for simulator	94
ETM.VMID	Virtual machine ID tracing	94
Keywords for the Trace Display		95
Examples for Trace Controlling		96
FAQ		98

Diagnosis	99
Error Diagnosis	99
Diagnosis Check List	104
ARM-ETM (LA-7921, LA-7990)	121
ARM-ETM AUTOFOCUS (LA-7991/LA-7992)	122
Support Request	131
Recommendations for Target Board Design	132
Technical Data	134
Operation Voltage	134
Dimensions	135
Adapters	147
Connector Layout	148
Training Arm CoreSight ETM Tracing(training_arm_etm.pdf)	1
ETM Setup	6
ETM Versions	6
Main Setup Windows	7
ETMv1	9
ETMv3	16
PTM (aka. PFT)	27
FLOWERROR	38
Displaying the Trace Contents	41
Source for the Recorded Trace Information	41
Sources of Information for the Trace Display	43
Influencing Factors on the Trace Information	44
States of the Trace	57
The AutoInit Command	58
Basic Display Commands	59
Display Items	63
Find a Specific Record	70
Belated Trace Analysis	72
Trace-based Debugging (CTS)	78
Forward and Backward Debugging	79
CTS Technique	84
Belated Trace-based Debugging	86
HLL Analysis of the Trace Contents	87
Trace Control by Filter and Trigger	91
Context	91
Filters and Trigger by Using the Break.Set Dialog	95
OS-Aware Tracing	114
OS (No Dynamic Memory Management)	114
OS+MMU (Dynamic Memory Management)	124

Specific Write Access vs. Context ID Packet	133
Task Statistics	134
Context ID Comparator	136
Function Run-Times Analysis	138
Software under Analysis (no OS, OS or OS+MMU)	138
Flat vs. Nesting Analysis	138
Flat Analysis	143
Nesting Analysis	166
Trace-based Code Coverage	185
Optimum ETM Configuration (No OS or OS)	185
Optimum ETM Configuration (OS+MMU)	185
Training Cortex-M Tracing (training_cortexm_etm.pdf)	1
History	4
Cortex-M Trace	4
Connectors	7
Basic Trace Configuration	8
Trace Buffer Management	10
MTB Program Flow Trace	13
ETM Program Flow Trace	14
ETM Configuration	14
Trace Capture	16
ETM Stream Mode	17
Displaying the Results	18
Trace Searching	21
Trace Filtering	24
Graphical Navigation	28
Analyzing the Results	30
Trace and Groups	40
Timing	45
Trace Based Code Coverage	48
Trace Based Debugging	49
Off-line Analysis	53
Data Watchpoint and Trace Unit	55
Instrumentation Trace Macrocell	66
Arm ETM Programming Dialog (trace_arm_etm_dialog.pdf)	1
Initialization	3
Initialization of the ETM	3
Programming	4
How to Start	4
Elements in the Dialog Box	4

Definitions	13
Examples	21
RAM Trace Port (trace_rtp.pdf)	1
Overview	3
FAQ	3
Quick Start	4
Commands	6
RTP	Ram trace port (RTP) 6
RTP.CLEAR	Clear tracebuffer 6
RTP.DirectDataMode	Simple trace mode 7
RTP.DirectDataMode.Mode	Direct data mode read/write 7
RTP.DirectDataMode.Width	Trace width in bits 7
RTP.HaltOnOverflow	Halt system on RTP FIFO overflow 8
RTP.Mode	Select the trace mode 8
RTP.OFF	Disables the RTP module 8
RTP.ON	Activates the RTP module 9
RTP.PortSize	Size of RTP data port 9
RTP.PortClock	Configure RTPCLK 9
RTP.RESet	Resets RTP settings 10
RTP.state	Display RTP setup 10
RTP.TraceMode	Complex trace mode 11
RTP.TraceMode.RAM<x>.SECTIon<y>	Configures a trace region 11
RTP.TraceMode.TraceExclude	Invert all trace regions 12
Arm Application Notes	 1
Arm Debug and Trace Interface Specification(app_arm_target_interface.pdf)	1
History	5
Introduction	6
LAUTERBACH Probes	7
Interface Types, Connectors and LAUTERBACH Probes	7
LAUTERBACH Probe Description	11
Debug Interface	16
Introduction	16
Debug Probe Connectors	17
Debug Signals	22
DC Electrical Characteristics	29
Debug Probe Driver/Receiver (IDC20A)	31
Debug Probe Driver/Receiver (HS Whisker and AUTO26)	33
AC Electrical Characteristics	36
JTAG Timing	37
SWD Timing	39

Target Design Considerations	41
Reset Considerations	45
Hot Plugging	46
Trace Interface	47
Parallel Trace Interface	47
Serial Trace Interface	52
PCIe Trace Interface	55
Converter	59
Cross Converting Table	59
Cross Converting Table for Manufacturer Specific Connectors	60
Converter Description	60
Setup of the Debugger for a CoreSight System(app_arm_coresight.pdf)	1
History	4
Introduction	5
Example of a CoreSight System	6
Using this Application Note	8
Your Chip is Available in the SYStem.CPU List	8
Your Chip is NOT Available in the SYStem.CPU List	8
Set up the Debugger Yourselfes for Debugging	8
Set up the Debugger Yourselfes for Tracing	10
Declare Multiple CoreSight Modules of the same Type	10
Configuration Example	11
Required Tool Hardware and Licenses	11
How to use the CoreSight Modules	13
Discover Available CoreSight Components	15
Debug Access Port (DAP)	17
Multiple Test Access Ports in the JTAG Chain	17
Serial Wire Debug Port (SW-DP)	19
Memory Access Ports	21
Real-time Memory Access	24
Core Debug Register Access	25
JTAG Access Port (JTAG-AP)	26
Cross Trigger Interface (CTI), Cross Trigger Matrix (CTM)	27
Performance Monitor Unit (PMU), BenchMark Counter (BMC)	28
Embedded Trace Macrocell (ETM), Program Trace Macrocell (PTM)	29
AMBA AHB Trace Macrocell (HTM)	30
Instrumentation Trace Macrocell (ITM), System Trace Macrocell (STM)	31
Funnel (CSTF), AMBA Trace Bus (ATB)	32

Data Watchpoint and Trace Unit (DWT), Flash Patch and Breakpoint Unit (FPB) ..	33
Embedded Logic Analyzer (ELA)	34
Embedded Trace Buffer (ETB, TMC used as ETB)	35
Embedded Trace FIFO (TMC used as ETF)	36
Embedded Trace Router (TMC used as ETR)	37
Embedded Trace Streamer (TMC used as ETS)	38
REPLICATOR (REP)	39
TRACEPORT	40
Trace Port Interface Unit (TPIU)	41
Serial Wire Viewer (SWV), Serial Wire Output (SWO)	43
High Speed Serial Trace Port (HSSTP)	45
Peripheral Component Interconnect Express (PCIe)	47
Debugging Embedded Cores in Xilinx FPGAs [Zynq] (app_xilinx_zynq.pdf)	1
Introduction	3
Physical Connection Requirements	4
Requirements for Parallel Trace	4
Requirements for Serial HSSTP Trace	7
Trace-Adapter for FMC-featured Target Boards	8
Zynq-7000 Devices	9
Exporting the Zynq-7000 Trace Interface via FixedIO/MIO	10
Exporting the Zynq-7000 Trace Interface via FPGA Fabric/PL: Using a clock divider	12
Exporting the Zynq-7000 Trace Interface via FPGA Fabric/PL: Using DDR I/O registers	15
Performing a Debugger-Based Boot on the Zynq-7000	19
UltraScale+ Devices	20
Exporting the UltraScale+ Trace Interface via FixedIO/MIO	21
Exporting the UltraScale+ Trace Interface via FPGA Fabric/PL	23
Exporting the UltraScale+ Trace Interface via HSSTP (up to 6.25 Gbps)	26
Exporting the UltraScale+ Trace Interface via HSSTP (10 Gbps)	32
Exporting the UltraScale+ Trace Interface via PCIe	38
Using the Example Design for the ZCU102	39
Performing a Debugger-Based Boot on the Zynq UltraScale+	39
Arm Application Note for MXC Chips (app_arm_mxc.pdf)	1
NEXUS Preprocessors	3
Basic NEXUS Handling	5
Settings of the SYStem Window	5
Trigger Settings	10
Further NEXUS Trace Analysis	12

Display of the T-Bit in the Trace.List Window	12
OS Kernel related Trace Analysis	13
Benchmark Counter Analysis using DPU Counters	15

AVR32

AVR32 Debugger and NEXUS Trace (debugger_avr32.pdf)	1
Warning	6
Introduction	7
Brief Overview of Documents for New Users	7
Demo and Start-up Scripts	8
Configuration	9
Debugger	9
Debugger and NEXUS Trace	10
Quick Start	11
Troubleshooting	13
Special Nexus Trace Troubleshooting	14
FAQ	14
AVR Specific Implementations	15
Breakpoints	15
Filter and Trigger for the NEXUS Trace	17
Memory Classes	19
Programming the On-chip FLASH of the AVR32	20
Special Hints, Restrictions, and Known Problems	21
Trace Extension	22
CPU specific SYStem Settings	23
SYStem.CONFIG.state	Display target configuration 23
SYStem.CONFIG	Configure debugger according to target topology 24
SYStem.CPU	Select the used CPU 28
SYStem.JtagClock	Define JTAG clock 28
SYStem.MemAccess	Select run-time memory access method 29
SYStem.Mode	Establish the communication with the target 30
SYStem.LOCK	Lock and tristate the debug port 30
SYStem.Option.IMASKASM	Disable interrupts while single stepping 31
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 31
SYStem.Option.MPU	Disable MPU during memory access 31
SYStem.Option.AUTO	Auto JTAG setting 31
SYStem.EraseChip	Erases the Flash and the EEprom 32
CPU specific TrOnchip Commands	33
TrOnchip.state	Display on-chip trigger window 33

TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	33
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	34
TrOnchip.RESet	Set on-chip trigger to default state	34
TrOnchip.EVTI	Allow the EVTI signal to stop the program execution	34
TrOnchip.EVTO	Output sync signals on EVT0	35
TrOnchip.EXTErnal	Generate a trigger for the trace on high pulse on INx	35
CPU specific Nexus Commands		36
NEXUS.BTM	Branch trace mode	36
NEXUS.DDR	Use the DDR transmission	36
NEXUS.DTM	Data trace mode	37
NEXUS.OFF	Switch the NEXUS trace port off	37
NEXUS.ON	Switch the NEXUS trace port on	37
NEXUS.OTM	Ownership trace messages	38
NEXUS.PinConfig	Override the nexus port pin mapping	38
NEXUS.PortMode	Change the nexus port clock frequency	38
NEXUS.Register	Display NEXUS trace control registers	38
NEXUS.RESet	Reset NEXUS trace port settings	39
NEXUS.Spen<messagetype>	Avoid message overrun	39
NEXUS.SQA	Synchronize trace by using full address	39
NEXUS.state	Display NEXUS port configuration window	39
NEXUS.WTM	Watch trace messages	40
Connectors		41
Debug Connector		41
NEXUS Connector		42

AVR8

AVR8 Debugger	(debugger_avr8.pdf)	1
History		4
Warning		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		7
Configuration		8
System Overview		8
Quick Start		9
Troubleshooting		11
FAQ		12
AVR Specific Implementations		13
Breakpoints		13

Overwriting Fuse and Lock Bits	13
Memory Classes	14
Programming the On-chip FLASH of the megaAVR	15
Special Hints, Restrictions, and Known Problems	15
CPU specific SYStem Settings	16
SYStem.CONFIG.state	Display target configuration 16
SYStem.CONFIG	Configure debugger according to target topology 17
SYStem.CONFIG.DEBUGPORTTYPE	Select debug port type 20
SYStem.CPU	Select the used CPU 21
SYStem.EraseChip	Erases the Flash and the EEprom 21
SYStem.JtagClock	Define JTAG clock 21
SYStem.LOCK	Lock and tristate the debug port 22
SYStem.MemAccess	Select run-time memory access method 22
SYStem.Mode	Establish the communication with the target 23
SYStem.Option.IMASKASM	Disable interrupts while single stepping 23
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 24
CPU specific TrOnchip Commands	25
TrOnchip.state	Display on-chip trigger window 25
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 25
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource 26
TrOnchip.RESet	Set on-chip trigger to default state 26
Connectors	27
Debug Connector	27
Converter 10-pin JTAG to 6-pin SPI for AVR8	27
Converter 10-pin JTAG to 6-pin UPDI for AVR8	28
Converter 10-pin JTAG to 8-pin UPDI for AVR8	28

Beyond

Beyond Debugger and Trace	(debugger_beyond.pdf) 1
History	4
Introduction	5
Brief Overview of Documents for New Users	5
Demo and Start-up Scripts	5
Warning	6
Limitations	7
Quick Start of the JTAG Debugger	8
Troubleshooting	10
Communication between Debugger and Processor can not be established	10
FAQ	10

Beyond Specific Implementations	11
Breakpoints	11
Runtime Measurement	14
Memory Classes	15
Beyond specific SYStem Commands	16
SYStem.CONFIG	Configure debugger according to target topology
SYStem.CONFIG.DebugProtocol	Implemented debug protocol of the CPU
SYStem.CONFIG.MemAccessModule	Select memory access module
SYStem.CPU	Select the used CPU
SYStem.JtagClock	Define JTAG frequency
SYStem.LOCK	Tristate the JTAG port
SYStem.MemAccess	Select run-time memory access method
SYStem.Mode	Establish the communication with the target
SYStem.Option.DBGRQ	Assert DBGRQ line while reset
SYStem.Option.FLOWTRACE	Debug support while FLOWTRACE
SYStem.Option.IMASKASM	Disable interrupts while single stepping
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping
SYStem.Option.LittleEnd	CPU endianness for memory access
SYStem.Option.LPMDDebug	Polling for low-power-mode
SYStem.Option.MMUSPACES	Separate address spaces by space IDs
SYStem.Option.ResetDURation	Reset assertion time
SYStem.Option.TURBO	Speed up memory access
SYStem.Option.WaitReset	Wait with JTAG activities after deasserting reset
SYStem.state	Display SYStem.state window
CPU specific MMU Commands	29
MMU.DUMP	Page wise display of MMU translation table
MMU.List	Compact display of MMU translation table
MMU.SCAN	Load MMU table from CPU
Beyond Specific TrOnchip Commands	34
TrOnchip.RESet	Reset on-chip trigger settings
TrOnchip.StepVector	Halt on exception entry when single-stepping
TrOnchip.state	Display on-chip trigger window
TrOnchip.Set	Trigger on exception
Beyond Specific TERM Commands	37
TERM.METHOD.BufferQUICK	Intrusive buffer based virtual terminal
JTAG Connection	38
Trace Connection	39

Blackfin

Blackfin Debugger	(debugger_blackfin.pdf) 1

Introduction	4
Brief Overview of Documents for New Users	4
Demo and Start-up Scripts	5
Location of Debug Connector	5
Warning	5
Quick Start JTAG	6
Troubleshooting	8
SYStem.Up Errors	8
FAQ	8
Configuration	9
System Overview	9
Blackfin specific SYStem Commands	10
SYStem.CONFIG	10
SYStem.CONFIG.CORE	15
SYStem.CPU	16
SYStem.JtagClock	17
SYStem.LOCK	17
SYStem.MemAccess	18
SYStem.Mode	19
SYStem.Option.IMASKASM	19
SYStem.Option.IMASKHLL	20
Breakpoints	21
Software Breakpoints	21
On-chip Breakpoints	21
Breakpoint in ROM	21
Example for Breakpoints	22
Memory Classes	23
CPU specific TrOnchip Commands	24
JTAG Connector	25

C166 Family

XC2000/XC16x/C166CBC Debugger	(debugger_166cbc.pdf)	1
Introduction		6
ICD/AICD		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		7
Warning		8
Monitor Routine		9

Quick Start	11
Quick Start for Tracing with MCDS On-chip Trace	13
Memory Classes	14
CPU specific SYStem Commands	15
SYStem.CPU	Select the CPU 15
SYStem.JtagClock	Define the JTAG frequency 16
SYStem.MemAccess	Select run-time memory access method 16
SYStem.Mode	Establish the communication with the CPU 17
SYStem.LOCK	Lock and tristate the debug port 18
SYStem.CONFIG.state	Display target configuration 18
SYStem.CONFIG	Configure debugger according to target topology 19
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 24
SYStem.CONFIG.DAP	Define mapping for DAP pins 25
SYStem.CONFIG.DAP.BreakPIN	Define mapping of break pins 25
SYStem.CONFIG.DAP.DAPENable	Enable DAP mode on PORST 25
SYStem.CONFIG.DAP.USERn	Configure and set USER pins 26
SYStem.CONFIG.DEBUGPORTTYPE	Set debug cable interface mode 26
SYStem.Option.DUALPORT	Run-time memory access for all windows 27
SYStem.Option.IDLEFIX	Periodically activate/deactivate JTAG connection 27
SYStem.Option.IMASKASM	Disable interrupts while single stepping 27
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 27
SYStem.Option.MonBase	Define start address of debug monitor 28
SYStem.Option.PERSTOP	Enable global peripheral suspend signal 28
SYStem.Option.PERSTOPFIX	Break CPU via ONCHIP break register 28
SYStem.Option.BRKOUT	Activates BRKOUT signal 29
SYStem.Option.WATCHDOG	Disable or serve watchdog 29
SYStem.Option.TRACEENABLE	Disable traceport 29
SYStem.Option.DebugLevel	Debug level 29
SYStem.Option.BootModelIndex	BootModelIndex 30
SYStem.Option.ICFLUSH	Flush instruction cache 31
SYStem.Option.IDLEDEBUG	Debug in IDLE state 31
SYStem.Option.WaitReset	Delay between PORST and JTAG shifts 31
MCDS Onchip Trace	32
MCDS Onchip Trace Features	32
Supported Features	32
Trace Control	32
Simple Trace Control	32
BenchMarkCounter	36
BMC.CNTx.EVENT	Configure the performance monitor 36
Useful Features	37
Breakpoints	38

Software Breakpoints on Instructions	38
On-chip Breakpoints	38
On-chip Breakpoints in FLASH/ROM	38
Example for Breakpoints	39
TrOnchip Commands	41
TrOnchip.state	Display on-chip trigger window 41
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 41
TrOnchip.RESet	Set on-chip trigger to default state 41
TrOnchip.TEnable	Set filter for the trace 42
TrOnchip.TOFF	Switch the sampling to the trace to OFF 42
TrOnchip.TON	Switch the sampling to the trace to "ON" 42
TrOnchip.TTrigger	Set a trigger for the trace 42
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource 43
TrOnchip.Address	Define address selector 43
TrOnchip.CYcle	Define access type 43
TrOnchip.Data	Define data selector 44
TrOnchip.NoMatch	Define match or nomatch comparison 44
TrOnchip.TaskID	Define task ID comparison 44
Connectors	45
JTAG Connector	45
DAP Connector	46
Troubleshooting	47
SYStem.Up Errors	47
FAQ	47
Technical Data	48
Operation Voltage	48
XC16x Application Notes	
Application Note Debug Cable C166	(c166_app_ocds.pdf) 1
Introduction	4
Debug Cables	5
OCDS Debug Cables	5
Debug Interface Description	11
JTAG Interface	11
DAP Interface	14
Configuring PowerView	16
Selecting the Interface Mode	16
Enabling the DAP Interface on the Chip	16
DAP User Pins	17
Adapters, Converters and Extensions	19

Adapter 16-pin 100 mil to 50 mil	19
Converter 16-pin JTAG to DAP for TriCore/XC2000/XC800	20
Recommended Connectors	21
Standard 2x8 Connector	21
Half-size 2x8 Connector	21
Half-size 2x5 Connector	22
Half-size 2x5 Connector with Keying Pin 7	23

CEVA-Oak/Teak/TeakLite

CEVA-Oak/Teak/TeakLite Debugger and Trace (debugger_oak.pdf)	1
Brief Overview of Documents for New Users	4
Warning	5
Quick Start	6
Troubleshooting	8
FAQ	8
CPU Specific Implementations	9
Breakpoints	9
Software Breakpoints	9
On-chip Breakpoints	9
CPU specific SYStem Settings	10
SYStem.CONFIG.state	Display target configuration 10
SYStem.CONFIG	Configure debugger according to target topology 11
SYStem.CPU	Select the used CPU 36
SYStem.JtagClock	Define JTAG clock 36
SYStem.LOCK	Lock and tristate the debug port 37
SYStem.MemAccess	Select run-time memory access method 37
SYStem.Mode	Establish the communication with the target 38
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits 38
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP 39
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits 39
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits 39
SYStem.Option.BackPC	Keep core running except for debugger access 40
SYStem.Option.BASE	Setup MAILBOX or MMIO base address 40
SYStem.Option.BigEndian	Enable big endian mode 40
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP 41
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check 41
SYStem.Option.DAPREMAP	Rearrange DAP memory map 42
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP 42
SYStem.Option.DEBUGPORTOptions	Options for debug port handling 43
SYStem.Option.EnReset	Allow the debugger to reset the processor 44

SYStem.Option.EXTDBGTRAP	Writing debug trap in external memory	44
SYStem.Option.IMASKASM	Disable interrupts while single stepping	44
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	44
SYStem.Option.MonBase	Set up monitor base address	45
General Restrictions		45
SYStem.Option.MONITOR	Define user-specific debug monitor	45
SYStem.Option.PALLADIUM	Extend debugger timeout	45
SYStem.Option.RisingTDO	Target outputs TDO on rising edge	46
SYStem.Option.RomMon	Disable download of monitor routine	46
SYStem.Option.TKLMON	Use TeakLite monitor for TeakLite-II	46
ETM Commands		47
ETM.AGU32	Data trace mode for AGU	47
ETM.AGU64	Data trace mode for AGU	47
ETM.ISTACK	Interrupt stack operation trace	47
ETM.RWM	Read-write-modify	47
ETM.WrapperFilter	Global breakpoint enable	48
ETM.WrapperSTALL	Enable/disable wrapper stall	48
TrOnchip Commands		49
TrOnchip.RESet	Set on-chip trigger to default state	49
TrOnchip.Set	Set on-chip trigger	49
TrOnchip.Set.BKRE	Trigger on block repeat loop	49
TrOnchip.Set.BRE	Trigger on jump	49
TrOnchip.Set.EXTRE	Trigger on external register read	50
TrOnchip.Set.EXTWE	Trigger on external register write	50
TrOnchip.Set.ILLE	Trigger on illegal instruction access	50
TrOnchip.Set.INTE	Trigger on interrupt	50
TrOnchip.Set.TBFE	Trigger on trace buffer full	50
TrOnchip.state	Display “Trigger-Onchip” dialog	51
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	51
JTAG Connector		52
Memory Classes		53

CEVA-X

CEVA-X Debugger and Trace	(debugger_cevax.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
Warning		6
Quick Start		7
Troubleshooting		9

FAQ	9
CPU Specific Implementations	10
Breakpoints	10
Software Breakpoints	10
On-chip Breakpoints	10
Disassembler	11
CPU specific SYStem Settings	12
SYStem.CONFIG.state	Display target configuration 12
SYStem.CONFIG	Configure debugger according to target topology 14
SYStem.CONFIG.EXTMEM	External program memory 40
SYStem.CPU	Select the used CPU 41
SYStem.JtagClock	Define JTAG clock 41
SYStem.LOCK	Lock and tristate the debug port 41
SYStem.MemAccess	Select run-time memory access method 42
SYStem.Mode	Establish the communication with the target 42
SYStem.Option.IMASKASM	Disable interrupts while single stepping 43
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 44
SYStem.Option.OVERLAY	Enable overlay support 44
SYStem.Option.PB0size	Setup size of internal program memory 45
SYStem.Option.RisingTDO	Target outputs TDO on rising edge 45
SYStem.VCU	Vector Computation Units (VCU) 46
SYStem.VCU.INSTANCES	Number of available VCUs 46
SYStem.VCU.MLD	MLD available or not 46
General Restrictions	46
CEVA-X specific ETM Command	47
ETM.BranchBBC	Control branch BBC mode 47
ETM.IgnoreISyncPredicate	Ignore I-Sync predicates 47
ETM.LoopBBC	Branch broadcast 47
ETM.PredicateAddress	Set predicate address 47
ETM.PredicatePeriod	Predicated counter in ETM wrapper 48
ETM.TimeStampInjectorTraceID	CoreSight ATB ID 48
ETM.WrapperFilter	Global breakpoint enable 48
ETM.WrapperSTALL	Enable/disable wrapper stall 48
TrOnchip Commands	49
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 49
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource 49
TrOnchip.RESet	Set on-chip trigger to default state 50
TrOnchip.Set	Set breakpoint 50
TrOnchip.state	Display “Trigger-Onchip” dialog 50
Ceva Specific Benchmarking Commands	51
BMC.CLOCKS.FORMAT	Cycle counter value format 54

Memory Classes	55
JTAG Connector	56

CPU32 and ColdFire

CPU32/ColdFire Debugger and Trace (debugger_68k.pdf)	1
History	5
Brief Overview of Documents for New Users	5
Demo and Start-up Scripts	6
Warning	6
Quick Start of the BDM Debugger	7
Quick Start of the ROM Monitor	9
Restrictions	12
Troubleshooting	13
FAQ	13
ROM Monitor	14
Monitor Features	14
Monitor Files	14
Address Layout	15
Vector Table	16
Configuration	17
Break without Hardware Interrupt	17
CPU specific Implementations	18
Hardware Breakpoint for MC68360	18
Memory Classes	19
CPU specific SYStem Commands	20
SYStem.BdmClock	Select BDM-clock 20
SYStem.CPU	Select CPU type 21
SYStem.LOCK	Lock and tristate the debug port 21
SYStem.MemAccess	Select run-time memory access method 22
SYStem.Mode	Establish the communication with the CPU 22
SYStem.CONFIG	Configure debugger according to target topology 23
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 28
SYStem.CONFIG.state	Display target configuration 29
SYStem.Option.BASE	Select peripheral base address 29
SYStem.Option.CACHE	Flush instruction cache on MC68349 29
SYStem.Option.HOOK	Compare PC to hook address 29
SYStem.Option.ICFLUSH	Flush of instruction cache before step and go 31
SYStem.Option.IMASKASM	Disable interrupts while single stepping 31

SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	31
SYStem.Option.MMUPhysLogMemaccess	Memory access preferences	31
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	32
SYStem.Option.OTE	Ownership trace	33
SYStem.Option.SLOWRESET	Slow reset enable	33
SYStem.Option.PST	Detect HALT condition of the CPU	33
SYStem.Option.PSTCLKTERM	Termination of the PSTCLK pin	34
SYStem.Option.ResetAction	Debugger behavior when RESET is detected	34
SYStem.Option.StandbyAction	Debugger behavior when power is restored	35
SYStem.Option.TracePULSE	Use PULSE instruction	35
SYStem.Option.TraceWDDATA	Use WDDATA instruction	35
SYStem.RESetOut	Reset target without reset of debug port	35
SYStem.Option.StepFlat	Step without exceptions	36
Trace specific SYStem.Option Commands		37
SYStem.Option.BTB	Change the width of the address information	37
SYStem.Option.DDC	Configure the tracing of data accesses	37
SYStem.Option.TSYNC	Send the PC to the trace port	37
CPU specific TrOnchip Commands		38
TrOnchip.ALIGN	Enable breakpoint alignment	38
TrOnchip.RESet	Set on-chip trigger to default state	38
TrOnchip.state	Display on-chip trigger window	38
TrOnchip.SIZE	Enable break on SIZE lines	39
TrOnchip.TEnable	Set filter for the trace	39
TrOnchip.TOFF	Switch the sampling to the trace to OFF	39
TrOnchip.TON	Switch the sampling to the trace to "ON"	39
TrOnchip.TTtrigger	Set a trigger for the trace	40
CPU specific MMU Commands		41
MMU.DUMP	Page wise display of MMU translation table	41
MMU.List	Compact display of MMU translation table	43
MMU.SCAN	Load MMU table from CPU	45
BDM Connector 68K		47
BDM and Trace Connector ColdFire		47
BDM Connectors for ColdFire V1, V2, V3, V4 and ColdFire+		47
Technical Data BDM 68K		50
Operation Voltage		50
Technical Data BDM ColdFire		51
Operation Voltage		51
Technical Data Trace ColdFire		52
Operation Voltage		52

DSP56K Debugger	(debugger_56000.pdf)	1
Introduction		4
Brief Overview of Documents for New Users		4
Demo and Start-up Scripts		4
Warning		6
Quick Start		7
Troubleshooting		10
SYStem.Up Errors		10
FAQ		10
Configuration		11
On-chip Flash Programming and Debugging on 56F8xxx Derivatives		11
DSP56K Specific Implementations		14
Breakpoints		14
Software Breakpoints		14
On-chip Breakpoints		14
CPU specific SYStem Settings and Restrictions		15
SYStem.CPU	Select the used CPU	15
SYStem.LOCK	Lock and tristate the debug port	15
SYStem.MemAccess	Select run-time memory access method	16
SYStem.Mode	Establish the communication with the target	16
SYStem.CONFIG.state	Display target configuration	18
SYStem.CONFIG	Configure debugger according to target topology	19
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	23
SYStem.Option.COP	Enable WATCHDOG	24
SYStem.Option.DE	Enable DE line	24
SYStem.Option.IMASKASM	Disable interrupts while single stepping	24
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	25
SYStem.Option.SoftBreakFix	Enables "SoftBreakFix" patch	25
SYStem.JtagClock	Define JTAG clock	26
General Restrictions		27
FPU		30
TrOnchip Commands		31
TrOnchip.state	Opens configure panel	31
TrOnchip.A	Trigger cycle	31
TrOnchip.AANDB	Triggers if event occurs on unit A and unit B	32
TrOnchip.AAFTERB	Triggers if event occurs first on unit A and then on unit B	32
TrOnchip.AORB	Triggers if event occurs on unit A or unit B	32
TrOnchip.B	Trigger cycle	32

TrOnchip.BAFTERA	Triggers if event occurs first on unit B and then on unit A	33
TrOnchip.Count	Delay counter	33
TrOnchip.DMA	Trigger on DMA access	33
TrOnchip.Mode	Defines used triggers	33
TrOnchip.OFF	Disable on-chip trigger unit	34
TrOnchip.RESet	Resets settings	34
Floating Point Formats		35
Integer Access Keywords		35
ONCE Connector (56002/56100)		36
JTAG Connector (56300, 56800, 56800E)		37
Memory Classes		39

dsPIC33

dsPIC33 Debugger	(debugger_pic.pdf)	1
History		4
Warning		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		7
Configuration		8
System Overview		8
Quick Start		9
Start a New Debug Session		9
Programming a Productive Application Binary		11
Troubleshooting		13
FAQ		14
dsPIC33 Specific Implementations		15
dsPIC33 Debug Monitor		15
Breakpoints		15
Memory Classes		17
Programming the On-chip FLASH of the dsPIC33		18
Special Hints, Restrictions, and Known Problems		18
CPU specific SYStem Settings		19
SYStem.CLockPrescaler	Select the prescaler for the debug clock	19
SYStem.CONFIG.state	Display target configuration	19
SYStem.CONFIG	Configure debugger according to target topology	20
System.CPU	Select the used CPU	21
SYStem.LOCK	Tristate the debug port	21

SYStem.MemAccess	Select run-time memory access method	22
SYStem.Mode	Establish the communication with the target	23
SYStem.Option	Special setup	24
SYStem.Option.BReakonWDT	Enable break on watchdog time-out	24
SYStem.Option.CLockSWitch	Enable clock group switch	24
SYStem.Option.ENableWDT	Enable watchdog timer	24
SYStem.Option.FastRC	Use FRC as debug port clock	25
SYStem.Option.FreezePer	Freeze peripherals on break or breakpoint	25
SYStem.Option.IMASKASM	Disable interrupts while single stepping	25
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	26
SYStem.Option.PARTitionconfig	Configure the Flash partitions	26
SYStem.Option.PoWeRSaVe	Enable PWRSaV instruction	26
SYStem.state	Display SYStem.state window	27
CPU specific TrOnchip Commands		28
Target Adaption		29
Probe Cables		29
Connector Type and Pinout		29

eSi-RISC

Debugger for eSi-RISC	(debugger_esir.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
Warning		6
Quick Start of the JTAG Debugger		7
Troubleshooting		9
SYStem.Up Errors		9
FAQ		9
eSi-RISC Specific Implementations		10
Access Classes		10
Breakpoints		11
Software Breakpoints		11
On-chip Breakpoints		11
CPU specific SYStem Commands		12
SYStem.CONFIG.state	Display target configuration	12
SYStem.CONFIG	Configure debugger according to target topology	13
SYStem.CPU	Select the used CPU	18
SYStem.JtagClock	Define JTAG frequency	18
SYStem.LOCK	Tristate the JTAG port	18

SYStem.MemAccess	Select run-time memory access method	20
SYStem.Mode	Establish the communication with the target	21
SYStem.Option.IMASKASM	Disable interrupts while single stepping	21
SYStem.Option.GPREG	Configure number of GP registers	22
SYStem.state	Display SYStem.state window	22
CPU specific TrOnchip Commands		23
Target Adaption		24
Connector Type and Pinout		24

eTPU

eTPU Debugger and Trace	(debugger_etpu.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
Warning		7
Target Design Requirement/Recommendations		8
General		8
Quick Start eTPU Debugger		9
Troubleshooting		10
FAQ		10
Configuration		11
System Overview		11
eTPU Debugger Specific Implementations		13
eTPU operating modes		13
Debugging the eTPU		13
Breakpoints and Watchpoints		14
Memory Classes		15
Address Spaces and Addressing Modes		15
CPU specific SYStem Commands		16
SYStem.CONFIG	Configure debugger according to target topology	16
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	16
SYStem.CPU	Select the CPU type	17
SYStem.JtagClock	Select the debug clock frequency	17
SYStem.LOCK	Lock and tristate the debug port	17
SYStem.MemAccess	Select run-time memory access method	18
SYStem.Mode	Select operation mode	19
SYStem.Option.ByteWise	Use byte addressing for eTPU memory space	19
SYStem.Option.DUALPORT	Implicitly use run-time memory access	20
SYStem.Option.HaltTwinEngine	Halt twin engine eTPU	20

CPU specific SYStem Commands	21
SYStem.Option.FreezeCLKS	Freeze eTPU clocks if eTPU halted 21
SYStem.Option.FreezePINS	Freeze pins if eTPU is halted 21
NEXUS specific SYStem Settings	22
NEXUS.BTM	Control for branch trace messages 22
NEXUS.CHAN	Enable CHAN register write trace messages 22
NEXUS.CLIENT<x>.MODE	Set data trace mode of nexus client 22
NEXUS.CLIENT<x>.SELECT	Select a nexus client for data tracing 23
NEXUS.DTM	Control for data trace messages 23
NEXUS.OFF	Switch the NEXUS trace port off 23
NEXUS.ON	Switch the NEXUS trace port on 23
NEXUS.OTM	Enable ownership trace messages 24
NEXUS.PortMode	Define MCKO frequency 24
NEXUS.PortSize	Define the width of MDO 24
NEXUS.PTCE	Program trace enable per channel 25
NEXUS.Register	Display NEXUS trace control registers 25
NEXUS.RESet	Reset NEXUS trace port settings 25
NEXUS.STALL	Stall the program execution 25
NEXUS.state	Display NEXUS port configuration window 26
CPU specific TrOnchip Commands	27
TrOnchip.BusTrigger	Trigger bus on debug event 27
TrOnchip.CBI	Halt on client breakpoint input 27
TrOnchip.CBT	Select client breakpoint timing condition 28
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 28
TrOnchip.EVTI	Use EVTI signal to stop the program execution 29
TrOnchip.EXTErnal	External signals 29
TrOnchip.HTWIN	Halt on twin engine breakpoint 29
TrOnchip.RESet	Reset on-chip trigger settings 30
TrOnchip.SCM	Select channels for that breakpoints are effective 30
TrOnchip.Set	Break on debug event 30
TrOnchip.TEnable	Set filter for the trace 31
TrOnchip.TOFF	Switch the sampling to the trace to OFF 31
TrOnchip.TON	Switch the sampling to the trace to "ON" 32
TrOnchip.TraceTrigger	Trigger trace on debug event 32
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource 33
TrOnchip.state	Display on-chip trigger window 33
Complex Trigger Unit	34
Usage	34
Complex Trigger Examples for eTPU	35
Keywords for the Complex Trigger Unit	36
JTAG Connector	37
Mechanical Description	37

GTM Debugger and Trace	(debugger_gtm.pdf)	1
History		7
Introduction		8
Brief Overview of Documents for New Users		8
Demo and Start-up Scripts		9
GTM Debugger and Trace		9
Warning		10
Target Design Requirement/Recommendations		11
General		11
For MPC57xx		11
Quick Start GTM Debugger		12
AURIX Architecture - Quick Start		13
MPC57xx/SPC58xx/SPC57xx Architecture - Quick Start		14
RH850 Architecture - Quick Start		15
S32 and SR6 Architecture - Quick Start		16
Troubleshooting		17
FAQ		17
Configuration		18
System Overview		18
GTM Operating Modes		18
Debugging the GTM		18
Breakpoints and Watchpoints		20
Access Classes		21
Address Spaces and Addressing Modes		21
CPU specific SYStem Commands		22
SYStem.CONFIG.state	Display target configuration	22
SYStem.CONFIG	Configure debugger according to target topology	23
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	23
SYStem.CONFIG.DEBUGPORTTYPE	Set debug cable interface mode	24
SYStem.CONFIG.MCSModule	Select the MCS module	24
SYStem.CONFIG.PortSHaRing	Control sharing of debug port with other tool	24
SYStem.CPU	Select the CPU type	25
SYStem.JtagClock	Select the debug clock frequency	26
SYStem.LOCK	Lock and tristate the debug port	26
SYStem.MemAccess	Select run-time memory access method	26
SYStem.Mode	Select operation mode	27
SYStem.Option.DUALPORT	Implicitly use run-time memory access	28
SYStem.Option.IMASKASM	Disable interrupts while single stepping	28

SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	28
SYStem.CONFIG.DEBUGPORT	Select target interface	29
NEXUS Commands		30
NEXUS.ARU	Control for ARU trace messages	30
NEXUS.ARUAccessX	ARU debugging address	30
NEXUS.FTM	Control for fetch trace messages	30
NEXUS.FTCE	Fetch trace enable per channel	31
NEXUS.DPLL	DPLL data trace messages	31
NEXUS.DPLLMemory	RAM module selection	32
NEXUS.DTM	Control for data trace messages	32
NEXUS.DTC	Data trace channel select	32
NEXUS.DTCE	Data trace enable per channel	33
NEXUS.OFF	Switch the NEXUS trace port off	33
NEXUS.ON	Switch the NEXUS trace port on	33
NEXUS.RESet	Reset NEXUS trace port settings	34
NEXUS.RefClock	Enable Aurora reference clock	34
NEXUS.PortMode	Set NEXUS trace port frequency	34
NEXUS.PortSize	Set trace port width	35
NEXUS.state	Display Nexus configuration window	35
NEXUS.SYNC	Address-sync trace messaging enable	35
NEXUS.TimeStamps	Control for timestamp trace messages	36
NEXUS.TimeStampSel	Select TBU channel for timestamp trace messages	36
General TrOnchip Commands		37
TrOnchip.state	Display onchip trigger window	37
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	38
TrOnchip.RESet	Reset on-chip trigger settings	38
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	38
TriCore specific TrOnchip Commands		39
TrOnchip.ARU	ARU settings	39
TrOnchip.ARU.ACCESS	ARU debugging address	39
TrOnchip.MCS	MCS setting	40
TrOnchip.MCS.Channel	Select the MCS channel	40
TrOnchip.MCS.Module	Select the MCS module	40
TrOnchip.OTGBx	OTGB0 and OTGB1 settings	41
TrOnchip.OTGBx.SELect	Select trace source	41
TrOnchip.OTGBx.LowBMType	Select IOS module for low byte	42
TrOnchip.OTGBx.HighBMType	Select IOS module for high byte	43
TrOnchip.OTGBx.LowBMInst	Low byte module instance	43
TrOnchip.OTGBx.HighBMInst	High byte module instance	43
TrOnchip.OTGBx.SENSitivNeg	Bit sensitive trace selection	44
TrOnchip.OTGBx.SENSitivPos	Bit sensitive trace selection	44
TrOnchip.OTGB2	OTGB2 setting	45

TrOnchip.OTGB2.SELect	Select trace source	45
TrOnchip.OTGBM0	OTGBM0 setting	46
TrOnchip.OTGBM0.SELect	Select trace source	46
TrOnchip.OTGBM1	OTGBM1 setting	47
TrOnchip.OTGBM1.SELect	Select trace source	47
CPU specific TrOnchip Commands		48
TrOnchip.ARUx.Address	ARU address compare	48
TrOnchip.ARUx.DataHigh	ARU data low value compare	48
TrOnchip.ARUx.DataLow	ARU data low value compare	48
TrOnchip.ARUx.HALT	ARU access halt enable	49
TrOnchip.ARUx.Watchpoint	ARU access watchpoint enable	49
TrOnchip.ATOMWPCx	ATOM watchpoint settings	50
TrOnchip.ATOMWPCx.Channel	ATOM channel selection	50
TrOnchip.ATOMWPCx.HALT	ATOM halt enable	50
TrOnchip.ATOMWPCx.Module	ATOM sub-module selection	50
TrOnchip.ATOMWPCx.TIMING	ATOM watchpoint enable	51
TrOnchip.ATOMWPCx.Transition	ATOM channel slope selection	51
TrOnchip.ATOMWPCx.Watchpoint	ATOM watchpoint enable	51
TrOnchip.DPLLWPC1	DPLL watchpoint settings	52
TrOnchip.DPLLWPC1.Event	DPLL source selection	52
TrOnchip.DPLLWPC1.HALT	DPLL TASI/SASI halt enable	52
TrOnchip.DPLLWPC1.Transition	DPLL TASI/SASI slope selection	52
TrOnchip.DPLLWPC1.Watchpoint	DPLL TASI/SASI watchpoint enable	52
TrOnchip.DPLLWPC2	DPLL RAM watchpoint settings	53
TrOnchip.DPLLWPC2.Address	DPLL RAM address compare	53
TrOnchip.DPLLWPC2.ACCESS	DPLL RAM read/write control	53
TrOnchip.DPLLWPC2.Data	DPLL RAM data compare	53
TrOnchip.DPLLWPC2.HALT	DPLL RAM access halt enable	53
TrOnchip.DPLLWPC2.Module	DPLL RAM module selection	54
TrOnchip.DPLLWPC2.Watchpoint	DPLL RAM access watchpoint enable	54
TrOnchip.EVTOx	Select EVTOx output	54
TrOnchip.SPEx	SPEx	55
TrOnchip.SPEx.DIR	SPEx DIR watchpoint settings	55
TrOnchip.SPEx.DIR.HALT	SPEx DIR halt enable	55
TrOnchip.SPEx.DIR.TIMING	SPEx DIR watchpoint enable	55
TrOnchip.SPEx.DIR.Transition	SPEx DIR slope selection	55
TrOnchip.SPEx.DIR.Watchpoint	SPEx DIR watchpoint enable	56
TrOnchip.SPEx.NIPD	SPEx NIPD watchpoint settings	57
TrOnchip.SPEx.NIPD.HALT	SPEx NIPD halt enable	57
TrOnchip.SPEx.NIPD.TIMING	SPEx NIPD watchpoint enable	57
TrOnchip.SPEx.NIPD.Transition	SPEx NIPD slope selection	57
TrOnchip.SPEx.NIPD.Watchpoint	SPEx NIPD watchpoint enable	58
TrOnchip.TBU	TBU watchpoint settings	59

TrOnchip.TBUx.Data	TBU data value compare	59
TrOnchip.TBUx.HALT	TBU access halt enable	59
TrOnchip.TBUx.Watchpoint	TBU access watchpoint enable	59
TrOnchip.TBU0.SELect	TBU0 type selection	60
TrOnchip.TIMWPC	TIM watchpoint settings	61
TrOnchip.TIMWPCx.Channel	TIM channel selection	61
TrOnchip.TIMWPCx.HALT	TIM halt enable	61
TrOnchip.TIMWPCx.Module	TIM sub-module selection	61
TrOnchip.TIMWPCx.TIMING	TIM watchpoint enable	62
TrOnchip.TIMWPCx.Transition	TIM channel slope selection	62
TrOnchip.TIMWPCx.Watchpoint	TIM watchpoint enable	62
TrOnchip.TOMWPC	TOM watchpoint settings	63
TrOnchip.TOMWPCx.Channel	TOM channel selection	63
TrOnchip.TOMWPCx.HALT	TOM halt enable	63
TrOnchip.TOMWPCx.Module	TOM sub-module selection	63
TrOnchip.TOMWPCx.TIMING	TOM watchpoint enable	64
TrOnchip.TOMWPCx.Transition	TOM channel slope selection	64
TrOnchip.TOMWPCx.Watchpoint	TOM watchpoint enable	64
TrOnchip.WPCE	Breakpoint enable per channel	65
Arm specific TrOnchip Commands		66
TrOnchip.TIOINWPCx	TIO input watchpoint settings	66
TrOnchip.TIOINWPCx.Channel	TIO input channel selection	66
TrOnchip.TIOINWPCx.HALT	TIO input halt enable	66
TrOnchip.TIOINWPCx.Module	TIO input sub-module selection	66
TrOnchip.TIOINWPCx.TIMING	TIO input watchpoint enable	67
TrOnchip.TIOINWPCx.Transition	TIO input channel slope selection	67
TrOnchip.TIOINWPCx.Watchpoint	TIO input watchpoint enable	67
TrOnchip.TIOOUTWPCx	TIO output watchpoint settings	68
TrOnchip.TIOOUTWPCx.Channel	TIO output channel selection	68
TrOnchip.TIOOUTWPCx.HALT	TIO output halt enable	68
TrOnchip.TIOOUTWPCx.Module	TIO output sub-module selection	68
TrOnchip.TIOOUTWPCx.Polarity	TIO output polarity selection	69
TrOnchip.TIOOUTWPCx.TIMING	TIO output watchpoint enable	69
TrOnchip.TIOOUTWPCx.Transition	TIO output channel slope selection	69
TrOnchip.TIOOUTWPCx.Watchpoint	TIO output watchpoint enable	69
RH850 specific TrOnchip Commands		70
TrOnchip.BreakChannel	Select the channel for breakpoints	70
TrOnchip.ATOMSlotx	Select the ATOM module for trace	70
TrOnchip.TIMSlotx	Select the TIM module for trace	70
JTAG Connector		71
Mechanical Description		71

H8S/23x9 Debugger	(debugger_h8s.pdf)	1
Introduction		4
Brief Overview of Documents for New Users		4
Demo and Start-up Scripts		5
Warning		6
Application Note		7
Location of Debug Connector		7
Reset Line		7
Enable JTAG Mode		7
Quick Start JTAG		8
Troubleshooting		10
SYStem.Up Errors		10
FAQ		10
Configuration		11
System Overview		11
System Commands		12
SYStem.CPU	CPU type selection	12
SYStem.JtagClock	JTAG clock selection	12
SYStem.Option.Advanced	Advanced addressing mode	13
SYStem.Option.BrkVector	Breakpoint trap	13
SYStem.Option.IMASKASM	Interrupt disable on ASM	13
SYStem.Option.IMASKHLL	Interrupt disable on HLL	13
SYStem.Option.KEYCODE	Keycode	14
SYStem.Option.SLOWRESET	Slow reset	14
SYStem.MemAccess	Select run-time memory access method	15
SYStem.Mode	System mode selection	15
Multicore Debugging		17
SYStem.LOCK	JTAG lock	17
SYStem.CONFIG	Configure debugger according to target topology	18
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	22
SYStem.CONFIG.state	Display target configuration	23
Breakpoints		24
Software Breakpoints		24
On-chip Breakpoints		24
Breakpoint in ROM		24
Example for Breakpoints		24
TrOnchip Commands		26
TrOnchip.state	Display on-chip trigger window	26

TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	26
TrOnchip.DMA	Trigger on DMA cycle	26
TrOnchip.DTC	Trigger on DTC cycle	27
TrOnchip.SIZE	Trigger on byte, word, long memory accesses	27
TrOnchip.RESet	Set on-chip trigger to default state	27
TrOnchip.SEQ	Sequential breakpoints	28
Memory Classes		29
Trace		30
FIFO Trace		30
Runtime Measurement		31
JTAG Connector		32

Hexagon

Hexagon Debugger	(debugger_hexagon.pdf)	1
History		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		6
Warning		7
Quick Start of the ICD Debugger for Hexagon		8
1. Hexagon Conceptual Basics		8
2. Prepare the Start		9
3. Select the Clock for the JTAG Communication		9
4. Configure the Debugger According to the Needs of the Application		9
5. Enter Debug Mode		10
6. Load the Application		10
7. View the Source Code		10
8. Write a Start-up Script		10
Debugger Basics		12
Memory Classes and Memory Access		12
Stack Display		12
Hexagon Security		12
Virtual Hardware Threads		13
On-chip Breakpoints		14
Restrictions		15
Troubleshooting		16
SYStem.Up Errors		16
FAQ		16

CPU specific SYStem Settings	17
SYStem.CONFIG.state	Display target configuration 17
SYStem.CONFIG	Configure debugger according to target topology 19
SYStem.CONFIG.MSA	Enable translation by system MMU 46
SYStem.CPU	Select CPU type 46
SYStem.JtagClock	Select clock for JTAG communication 47
SYStem.LOCK	Tristate the JTAG port 47
SYStem.MemAccess	Select run-time memory access method 48
SYStem.Option.MMUSPACES	Separate address spaces by space IDs 48
SYStem.Option.MACHINESPACES	Address extension for guest OSES 49
SYStem.Mode	Select target reset mode 51
SYStem.Option	CPU specific commands 52
SYStem.Option.BUGFIX	Workaround for single-stepping an RTE instruction 52
SYStem.Option.DCFREEZE	Do not invalidate cache 52
SYStem.Option.CLADE	Enable debugger support for CLADE 53
SYStem.Option.CLADEDICT	Load CLADE dictionary from file 53
SYStem.Option.CLADEPARAM	Define the CLADE address parameters 54
SYStem.Option.CLADEREAD	Use hardware for reading compressed RAM 54
SYStem.Option.ICFLUSH	Flush instruction cache at “Go” or “Step” 54
SYStem.Option.IMASKASM	Disable interrupts while single stepping 55
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 55
SYStem.Option.ISDBSoftBreakFix	Workaround for prefetch issue 56
SYStem.Option.PC	PC parking position 56
SYStem.RESetOut	Reset target without reset of debug port 57
SYStem.Option.REVIsion	Define default value for REV register 57
SYStem.Option.SRST	Reset via SRST line 57
SYStem.StuffInstruction.Assemble	Execute assembly stuff instruction 58
SYStem.StuffInstruction.OpcodE	Execute opcode stuff instruction 58
SYStem.Option.TCMBase	Base address of the TCM 58
SYStem.Option.TLBINV	Invalidate TLB while MMU is off 59
SYStem.state	Display SYStem.state window 59
CPU specific MMU Commands	60
MMU.DUMP	Page wise display of MMU translation table 60
MMU.List	Compact display of MMU translation table 62
MMU.MAP	Translations from ELF file memory load map 63
MMU.MAP.dump	Display addresses from ELF file memory load map 63
MMU.MAP.SCAN	Load MMU table from ELF file 63
MMU.SCAN	Load MMU table from CPU 64
MMU.Set	Write to MMU on processor 66
MMU.TLB	Scan or dump MMU TLB entries 66
MMU.VTLB	Scan or dump VTLB entries 67
CPU specific BenchMarkCounter Commands	68
BMC.COUNTER<n>	Specify event count 68

BMC.CyclePeriod	Specify export rate	69
BMC.OFF	Disable benchmark counters	69
BMC.ON	Enable benchmark counters	69
BMC.SELect	Select counter for statistic analysis	69
BMC.SPDM	Specify profiler control	70
BMC.SPLIT	Specify export rate	71
TrOnchip Commands		72
TrOnchip.ContextID	Extend on-chip breakpoint/trace filter by TID	72
TrOnchip.MatchASID	Extend on-chip breakpoint/trace filter by ASID	72
TrOnchip.RESet	Reset “TrOnchip” settings	73
TrOnchip.StepException	Single stepping of exceptions and interrupts	73
TrOnchip.state	Display on-chip trigger window	73
JTAG Connector		74
Mechanical Description of the 20-pin Debug Cable		74
Electrical Description of the 20-pin Debug Cable		75
Hexagon-ETM Trace	(trace_hexagon_etm.pdf)	1
Controlling the ETM		4
ETM.state/Breakpoints		5
Commands		12
ETM	Embedded Trace Macrocell	12
ETM.ATBSIZE	ATB bus size	12
ETM.BBC	Broadcast all branches	12
ETM.CLEAR	Clear trace and sequencer settings	12
ETM.CLOCK	Set core clock frequency for timing measurements	13
ETM.ContextID	Provide TID in synchronisation packages	14
ETM.CycleAccurate	Cycle accurate tracing	15
ETM.CycleAccurateShift	Shift cycle accurate timestamps	15
ETM.CycleCoarse	Use coarse cycle accurate mode	15
ETM.CallReturnOnly	Trace only functions calls and returns	16
ETM.DataTrace	Define broadcast of load/store address tracing	16
ETM.DataTraceSelect	Select with units or cluster are traced	16
ETM.DISableClockOff	Compress trace information	17
ETM.FillPort	Compress trace information	17
ETM.ForceAtom	Force atom	18
ETM.ForceLdStKillAtom	Force atom	18
ETM.GlobalSyncs	GSYNC control	18
ETM.GSyncPeriod	Specify GSYNC period	18
ETM.GSyncPeriodDisable	GSYNC period disable	19
ETM.GSyncWithUserPkt	GSYNC with USERPKT	19
ETM.IgnoreAtom	Ignore E/N atoms	19
ETM.IgnoreSyncOverflow	Handle ignore synch-overflow bit	20
ETM.LoopTrace	Control for loop back branch packets	20

ETM.OFF	Switch ETM off	20
ETM.ON	Switch ETM on	21
ETM.PortMode	Select ETM mode	21
ETM.PortSize	Define trace port width	21
ETM.Register	Display the ETM register	22
ETM.RESet	Reset ETM settings	23
ETM.Set	Set ETM registers	24
ETM.STALL	Stall control	33
ETM.SynchPeriod	Specify ISYNC period	34
ETM.TestBusBeforeSleep	TestBus before sleep	34
ETM.TestBusSampleGroup	TestBus sample group	34
ETM.TestBusSweepingMode	TestBus sweeping mode	34
ETM.TestBusTrace	TestBus tracing control	35
ETM.TestBusTraceMUX	Select bus for 'TestBus' tracing	35
ETM.TestBusTracePeriod	Specify test bus sampling time	35
ETM.TestBusTriggerMode	TestBus trigger mode	35
ETM.TlmeMode	Improve ETM timestamp information	36
ETM.TlmeSyncs	Time syncs	36
ETM.Trace	Broadcasting of instruction flow	36
ETM.TraceASID	Broadcast instruction trace of specified ASID	36
ETM.TraceID	Set ETM trace ID manually	37
ETM.TracePriority	Define priority of ETM messages	37
ETM.TraceTID	Broadcast instruction trace of specified software thread	37
ETM.TraceTNUM	Broadcast instruction trace of specified hardware thread	38
ETM.TSyncWithUserPkt	TSYNC with user packet	38
ETM.state	Display ETM setup	38
ETM.UserPktDisable	Disable user packets	39
Keyword for the Trace Display		40
Trace Commands for SMP Debugging		41
Benchmark Counters		45

IPU

IPU Debugger	(debugger_ipu.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Warning		6
Quick Start of the JTAG Debugger		7
Troubleshooting		9
SYStem.Up Errors		9
FAQ		9

IPU Specific Implementations	10
IPUS and IPUV Core Debugging in Heterogeneous SMP System	10
IPU Specific Peripheral Files	11
Breakpoints	12
Memory Access Classes	13
CPU specific SYStem Commands	14
SYStem.CONFIG.state	Display target configuration 14
SYStem.CONFIG	Configure debugger according to target topology 15
SYStem.CPU	Select the used CPU 29
SYStem.JtagClock	Define JTAG frequency 29
SYStem.LOCK	Tristate the JTAG port 31
SYStem.MemAccess	Select run-time memory access method 32
SYStem.Mode	Establish the communication with the target 33
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits 33
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP 33
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits 34
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits 35
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check 35
SYStem.Option.DAPREMAP	Rearrange DAP memory map 35
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP 36
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP 36
SYStem.Option.DEBUGPORTOptions	Options for debug port handling 37
SYStem.state	Display SYStem.state window 38
IPU Specific TrOnchip Commands	39
TrOnchip.Set.FINISH	Set 'Break on Finish' on-chip breakpoint 39
TrOnchip.Set.POS	Set on-chip trigger for total pixel position 39
TrOnchip.Set.XPOS	Set on-chip trigger for horizontal pixel position 39
TrOnchip.Set.YPOS	Set on-chip trigger for vertical pixel position 40
TrOnchip.RESet	Set on-chip trigger to default state 40
TrOnchip.state	Display on-chip trigger window 40

M32R

M32R Debugger and Trace	(debugger_m32r.pdf) 1
Introduction	5
Brief Overview of Documents for New Users	5
Demo and Start-up Scripts	5
Warning	7
Quick Start	8
Troubleshooting	11
SYStem.Up Errors	11

Memory Access Errors	11
FAQ	11
CPU specific SYStem Settings and Restrictions	12
SYStem.CONFIG	Configure debugger according to target topology 12
SYStem.CPU	Select target CPU 12
SYStem.JtagClock	Define JTAG clock 13
SYStem.LOCK	Lock and tristate the debug port 13
SYStem.MemAccess	Select run-time memory access method 14
SYStem.Mode	Establish the communication with the target 15
SYStem.Option	Display SYStem window 15
SYStem.Option.DBI	Enables program break via debug interrupt 15
SYStem.Option.IMASKASM	Disable interrupts while single stepping 16
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 16
SYStem.Option.KEYCODE	Code protection 17
SYStem.Option.TriState	Allow debugger to drive JTAG and reset 18
SYStem.state	Display SYStem.state window 18
Trace specific Commands	19
SYStem.Option.BTM	Enables program trace messages 19
SYStem.Option.DTM	Enables data trace messages 19
SYStem.Option.STALL	Trace message overrun control 19
SYStem.Option.TRCLK	Trace output clock ratio 20
SYStem.Option.TRDATA	Trace port width 20
TrOnchip	21
TrOnchip.RESet	Resets all TO settings 21
TrOnchip.state	Opens configuration panel 22
Security Levels of the M32R Family	23
Security Level	23
Flash Erase if Device is secured	24
General Restrictions and Hints	25
Floating Point Formats	26
Integer Access Keywords	26
JTAG Connection	27
Mechanical Description of the 10-pin Debug Cable	27
Electrical Description of the 10-pin Debug Cable	27
Mechanical Description of the 20-pin Trace Connector	28
Memory Classes	30

M8051EW

M8051EW Debugger	(debugger_m8051ew.pdf) 1
Introduction	5

Warning	6
Quick Start	7
Troubleshooting	9
SYStem.Up Errors	9
KEIL OMF-51 and OMF2	10
Breakpoints	11
Debugging with Low Target Frequencies	13
Mapping Memory	14
FAQ	14
Configuration	15
CPU specific SYStem Settings and Restrictions	16
SYStem.state	Open SYStem.state window 16
SYStem.CONFIG.state	Display target configuration 16
SYStem.CONFIG	Configure debugger according to target topology 17
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 21
SYStem.CPU	Select CPU 22
SYStem.JtagClock	Define JTAG clock 23
SYStem.LOCK	Lock and tristate the debug port 23
SYStem.MemAccess	Select run-time memory access method 24
SYStem.Mode	Establish communication with the target 24
SYStem.Option.IMASKASM	Disable interrupts while single stepping 26
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 26
SYStem.Option.IntelSOC	Slave core is part of Intel® SoC 26
SYStem.Option.LittleEnd	Selection of little endian mode 27
SYStem.Option.PATCHBP	Use patch unit for on-chip breakpoints 27
SYStem.Option.PRDELAY	Set delay time after RESET 28
SYStem.Option.ResBreak	Request break after reset 29
SYStem.Option.TRAPEN	Enable TRAP_EN flag in EOR 29
Memory Classes	30
SYMBOL Commands	31
Special Function Register (SFR) symbols	31
TrOnchip Commands	33
TrOnchip.state	Display on-chip trigger window 33
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 33
TrOnchip.RESet	Set on-chip trigger to default state 33
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource 34
JTAG Connectors	35
Target Board Connectors	35
LAUTERBACH Adapters	39

R8051XC Debugger	(debugger_r8051xc.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Warning		6
Quick Start		7
Troubleshooting		9
SYStem.Up Errors		9
KEIL OMF-51 and OMF2		10
Debugging with Low Target Frequencies		10
Mapping Memory		11
FAQ		11
Configuration		12
CPU specific SYStem Settings and Restrictions		13
SYStem.state	Open system window	13
SYStem.CONFIG.state	Display target configuration	13
SYStem.CONFIG	Configure debugger according to target topology	14
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	18
SYStem.CPU	Select CPU	19
SYStem.JtagClock	Define JTAG clock	20
SYStem.LOCK	Lock and tristate the debug port	20
SYStem.MemAccess	Select run-time memory access method	21
SYStem.Mode	Establish communication with the target	21
SYStem.Option.IMASKASM	Disable interrupts while single stepping	23
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	23
SYStem.Option.LittleEndian	Treat memory as little endian	23
SYStem.Option.DPTREXT	Selects the address of DPS register	23
SYStem.Option.PRDELAY	Set delay time after RESET	24
Memory Classes		24
SYMBOL Commands		25
Special Function Register (SFR) symbols		25
TrOnchip Commands		27
TrOnchip.state	Display on-chip trigger window	27
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	27
TrOnchip.RESet	Set on-chip trigger to default state	27
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	28
JTAG Connectors		29
LAUTERBACH Adapters		29

MAC71xx/72xx NEXUS Debugger and Trace	(nexus_mac.pdf)	1
Brief Overview of Documents for New Users		5
Warning		6
Quick Start of the JTAG Debugger		7
Troubleshooting		9
Communication Between Debugger and Processor cannot be established		9
FAQ		9
Trace Extension		9
ARM specific Implementations		10
Breakpoints		10
Trigger		14
Virtual Terminal		14
Semihosting		14
Runtime Measurement		15
Coprocessors		15
Memory Classes		16
Programming the On-chip FLASH of the MAC71/72xx		17
ARM specific SYStem Commands		18
SYStem.CONFIG.state	Display target configuration	18
SYStem.CONFIG	Configure debugger according to target topology	18
SYStem.CPU	Select the used CPU	20
SYStem.JtagClock	Define JTAG frequency	21
SYStem.LOCK	Tristate the JTAG port	22
SYStem.MemAccess	Select run-time memory access method	23
SYStem.Mode	Establish the communication with the target	24
SYStem.Option.ABORTFIX	Do not access 0x0-0x1f	25
SYStem.Option.BUGFIX	Breakpoint bug fix for ARM7TDMI-S REV2	25
SYStem.Option.BigEndian	Define byte order (endianness)	26
SYStem.Option.CFLUSH	FLUSH the cache before step/go	26
SYStem.Option.DBGACK	DBGACK active on debugger memory accesses	26
SYStem.Option.DisMode	Define disassembler mode	27
SYStem.Option.EnReset	Allow the debugger to drive nRESET/nSRST	27
SYStem.Option.IMASKASM	Disable interrupts while single stepping	28
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	28
SYStem.Option.OVERLAY	Enable overlay support	28
SYStem.Option.INTDIS	Disable all interrupts	29
SYStem.Option.LOCKRES	Go to 'Test-Logic Reset' when locked	29
SYStem.Option.NOIRCHECK	No JTAG instruction register check	29

SYStem.Option.PC	Define address for dummy fetches	30
SYStem.Option.ResBreak	Halt the core after reset	31
SYStem.Option.RisingTDO	Target outputs TDO on rising edge	31
SYStem.Option.ShowError	Show data abort errors	32
SYStem.Option.SOFTLONG	Use 32-bit access to set breakpoint	32
SYStem.Option.SOFTWARE	Use 16-bit access to set breakpoint	32
SYStem.Option.SPLIT	Access memory depending on CPSR	32
SYStem.Option.STEPSOFT	Use software breakpoints for ASM stepping	33
SYStem.Option.TRST	Allow debugger to drive TRST	33
SYStem.Option.TURBO	Speed up memory access	33
SYStem.Option.WaitReset	Wait with JTAG activities after deasserting reset	34
SYStem.RESetOut	Assert nRESET/nSRST on JTAG connector	34
ARM specific NEXUS Commands		35
NEXUS.BTM	Control for branch trace messages	35
NEXUS.ThumbBTM	Control for branch trace messages	35
NEXUS.OTM	Control for ownership trace messages	35
NEXUS.WTM	Control for watchpoint messages	36
NEXUS.DTM	Control for data trace messages	36
NEXUS.PortMode	Set NEXUS trace port frequency	36
NEXUS.PortSize	Set trace port width	36
NEXUS.UBA	Specify user base address	37
NEXUS.STALL	Stall the program execution	37
ARM specific TrOnchip Commands		38
TrOnchip.RESet	Reset on-chip trigger settings	38
TrOnchip.CONVert	Extend the breakpoint range	38
TrOnchip.Mode	Configure unit A and B	39
TrOnchip.A	Programming the ICE breaker module	39
TrOnchip.A.Value	Define data selector	39
TrOnchip.A.Size	Define access size for data selector	40
TrOnchip.A.CYcle	Define access type	40
TrOnchip.A.Address	Define address selector	41
TrOnchip.A.Trans	Define access mode	41
TrOnchip.A.Extern	Define the use of EXTERN lines	42
TrOnchip.state	Display on-chip trigger window	42
Filter and Trigger for the NEXUS Trace		43
Filter and Trigger provided by the Processor		43
Nexus specific TrOnchip Commands		44
TrOnchip.EVTI	Allow the EVTI signal to stop the program execution	44
TrOnchip.EXTErnal	Generate a trigger for the trace on high pulse on INx	45
JTAG Connection		46
Mechanical Description of the 20-pin Debug Cable		46
Electrical Description of the 20-pin Debug Cable		47

Mechanical Description of the 14-pin Debug Cable	48
Electrical Description of the 14-pin Debug Cable	48
Technical Data	49
Pinout MICTOR	49
Mechanical Dimension	50
Adapter	50
Operation Voltage	50

MCS08

MCS08 Debugger (debugger_hc08.pdf)	1
Brief Overview of Documents for New Users	5
Warning	6
Troubleshooting	7
SYStem.Up Errors	7
FAQ	7
CPU Specific Implementations	8
Breakpoints	8
Software Breakpoints	8
On-chip Breakpoints	8
Quick Start of the ICD Debugger for HC9S08	9
1. Prepare the Start	9
2. Select the Clock for the BDM Communication	10
3. Configure the Debugger According to the Needs of the Application	10
4. Map the EPROM Simulator if Available (optional)	10
5. Tell the Debugger Where it should use On-chip Breakpoints (optional)	10
6. Enter Debug Mode	11
7. Load the Program	11
8. Initialize Program Counter and Stackpointer	12
9. View the Source Code	12
CPU specific SYStem Settings and Restrictions	14
Restrictions	14
SYStem.BdmClock	Select clock for BDM communication 14
Special Functions	15
SYStem.CONFIG	Configure debugger according to target topology 15
SYStem.CPU	Select CPU type 16
SYStem.LOCK	Lock and tristate the debug port 16
SYStem.MemAccess	Select run-time memory access method 17
SYStem.Mode	Select target reset mode 17
SYStem.Option.IMASKASM	Disable interrupts while single stepping 19
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 19

Hardware Breakpoints	20
Program Breakpoints	20
Read and Write Breakpoints	20
Data Breakpoints	21
Onchip Commands	22
Onchip.Mode.EventTrace	Start recording after trigger event 22
Onchip.Mode.FlowTrace	Flow trace mode 22
Onchip.Mode.LoopTrace	Inhibit redundant entries 22
TrOnchip Commands	23
TrOnchip.Mode	Select trace and trigger mode 23
TrOnchip.state	Open the control window for the on-chip trigger resources 24
TrOnchip.RESet	Reset the on-chip trigger resources 24
Memory Classes	25
FLASH EEPROM Management	26
Secure and Unsecure	27
BDM Connector ICD-MCS08	28

MCS12

MCS12 Debugger	(debugger_hc12.pdf) 1
Brief Overview of Documents for New Users	5
Warning	6
Troubleshooting	7
SYStem.Up Errors	7
FAQ	7
CPU Specific Implementations	8
Breakpoints	8
Software Breakpoints	8
On-chip Breakpoints	8
Quick Start of the ICD Debugger for HC12	9
1. Prepare the Start	9
2. Select the Clock for the BDM Communication	10
3. Configure the Debugger according to the Needs of the Application	10
4. Map the EPROM Simulator if Available (optional)	11
5. Tell the Debugger Where it should use On-chip Breakpoints (optional)	11
6. Enter Debug Mode	11
7. Load the Program	12
8. Initialize Program Counter and Stackpointer	12
9. View the Source Code	13

CPU specific SYStem Settings and Restrictions	15
Restrictions	15
SYStem.BdmClock	Select clock for BDM communication 16
SYStem.CONFIG	Configure debugger according to target topology 16
SYStem.CPU	Select CPU type 17
SYStem.LOCK	Lock and tristate the debug port 17
SYStem.MemAccess	Select run-time memory access method 17
SYStem.Mode	Select target reset mode 18
SYStem.Option.BASE	Base address of internal registers 19
SYStem.Option.CLKSW	Force BDM to work on CPU bus frequency 19
SYStem.Option.DUALPORT	All memory accesses are done hidden 20
SYStem.Option.GLOBAL	Memory accesses are done global 20
SYStem.Option.MonBase	Monitor relocation 21
SYStem.Option.SOFTWARE	Breakpoints are set by word access 21
SYStem.Option.VFP	Voltage for FLASH programming 22
SYStem.Option.WATCHDOG	COP support 23
Hardware Breakpoints	24
Program Breakpoints	24
Read and Write Breakpoints	24
Data Breakpoints	25
Onchip Commands	26
Onchip.Mode.DetailTrace	Detailed trace recording mode 26
Onchip.Mode.EventTrace	Start recording after trigger event 26
Onchip.Mode.FlowTrace	Flow trace mode 26
Onchip.Mode.LoopTrace	Inhibit redundant entries 26
Onchip.Mode.CPU	Select CPU as onchip trace source 27
Onchip.Mode.XGATE	Select XGATE as onchip trace source 27
Onchip.Mode.BOTH	Select CPU and XGATE as onchip trace source 27
Onchip.Mode.TimeStamp	Enable onchip timestamps 27
TrOnchip Commands	28
TrOnchip.state	Display on-chip trigger window 28
TrOnchip.Mode	Select trace and trigger mode 29
TrOnchip.RESets	Set on-chip trigger to default state 30
TrOnchip.XBreak	Enable crossbreak between S12 core and XGATE 30
TrOnchip.RESERVE	Deprive debugger of address comparator use 30
Memory Classes	31
Debugging with active PLL	32
Debugging with active Watchdog	33
FLASH EEPROM Management	34
FLASH EEPROM on S12X Derivatives	34
EEPROM Management	35

Banked Applications	37
Background and Compatibility Information	37
SYStem.Option.PAGING	Banked applications 37
SYStem.Option.RAMHM	Alternate RAM mapping 38
SYStem.Option.ROMHM	ROM in second half of map 38
SYStem.Option.TRANS	Transparent mode 38
Local and Global Memory Map on S12X Targets	41
Using the MMU for HC12DA/DG/DT128	42
SYStem.Option.MEMEXP	Memory expansion 43
SYStem.Option.ROMTST	FLASH EEPROM test mode 43
Using the MMU for HC12A4/F8	44
BDM Connector	48
BDM Connector ICD - MCS12, ICD-S12X	48
BDM Connector ICD - HC12	48

Meta

Meta Debugger	(debugger_meta.pdf) 1
Introduction	5
Brief Overview of Documents for New Users	5
Demo and Start-up Scripts	6
Warning	6
Quick Start of the Debugger	8
Troubleshooting	10
Communication between Debugger and Processor can not be established	10
FAQ	10
Meta specific Implementations	11
Meta Configuration	11
Access Classes	12
Breakpoints	13
SYStem.CONFIG	Configure debugger according to target topology 15
SYStem.CONFIG.state	Display target configuration 21
SYStem.CPU	Select the used CPU 21
SYStem.JtagClock	Define JTAG frequency 21
SYStem.LOCK	Lock and tristate the debug port 22
SYStem.MemAccess	Select run-time memory access method 23
SYStem.Mode	Establish communication with target 24
SYStem.Option.IMASKASM	Disable interrupts while single stepping 24
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 26
SYStem.Option.MINIM	Map execution- to storage address range 26
SYStem.state	Display SYStem.state window 26

TrOnchip.RESet	Set on-chip trigger to default state	27
TrOnchip.state	Display on-chip trigger window	27
Target Adaption		28
Interface Standards JTAG, Serial Wire Debug, cJTAG		28
Connector Type and Pinout		28

Mico32

Mico32 Debugger	(debugger_mico32.pdf)	1
History		4
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		6
Warning		7
Quick Start of the Debugger		8
Troubleshooting		10
Communication between Debugger and Processor can not be established		10
FAQ		10
Mico32 specific Implementations		11
Mico32 Configuration		11
Mico32 Debug Monitor		11
Access Classes		12
Breakpoints		14
Mico32 specific Event for the ON and GLOBALON Command		16
Mico32 specific SETUP Commands		17
SETUP.DIS	Disassembler configuration	17
Mico32 specific SYStem Commands		19
SYStem.CONFIG.state	Display target configuration	19
SYStem.CONFIG	Configure debugger according to target topology	20
SYStem.CPU	Select the used CPU	26
SYStem.JtagClock	Define JTAG frequency	26
SYStem.LOCK	Lock and tristate the debug port	27
SYStem.MemAccess	Select run-time memory access method	28
SYStem.Mode	Establish communication with target	29
SYStem.Option.AllowDirectIWAcess	Allow direct instruction bus access	29
SYStem.Option.CacheCoherentACCESS	Second level cache settings	30
SYStem.Option.CorePowerDetection	Special core power detection	30
SYStem.Option.DUALPORT	Update all memory displays during runtime	31
SYStem.Option.IMASKASM	Disable interrupts while single stepping	31
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	32

SYStem.Option.STEPSOFT	Use alternative method for ASM single step	32
SYStem.state	Display SYStem.state window	32
Mico32 specific TrOnchip Commands		33
TrOnchip.RESet	Set on-chip trigger to default state	33
TrOnchip.state	Display on-chip trigger window	33
TrOnchip.StepVector	Halt on exception entry when single-stepping	33
Target Adaption		34
Probe Cables		34
Interface Standards JTAG		34
Connector Type and Pinout		34

MicroBlaze

MicroBlaze Debugger and Trace	(debugger_microblaze.pdf)	1
History		5
General Note		6
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		7
MicroBlaze Debug and Trace Features Supported by TRACE32		8
ESD Protection		9
Quick Start of the Debugger		10
Quick-Start of the Real-Time Trace		13
Compiling Software with Debug Information		14
Troubleshooting		15
SYStem.Up Errors		15
FAQ		15
Displaying MicroBlaze Core Configuration		16
CPU specific Implementations		17
Memory Accesses Causing Bus Errors		17
Breakpoints		18
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	20
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	20
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits	20
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	21
SYStem.Option.BrkHandler	Control writing of software break handler	21
SYStem.Option.BrkVector	Configures an alternative breakvector	22
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	23
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	23

SYStem.Option.DAPREMAP	Rearrange DAP memory map	24
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	24
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	25
SYStem.Option.IMASKASM	Interrupt disable on ASM	26
SYStem.Option.IMASKHLL	Interrupt disable on HLL	26
SYStem.Option.LittleEndian	Select little endian mode	26
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	26
SYStem.Option.ResetMode	Select the reset mode	27
SYStem.Option.DUALPORT	Use real-time access by default	28
SYStem.Option.MDMSINGLELMB	Use MDM LMB master 0 for all cores	28
TERM.METHOD.MDMUART	Terminal configuration	28
Memory Classes		30
Register Names		30
CPU specific SYStem Commands		31
SYStem.CPU	Select the used CPU	31
SYStem.JtagClock	Selects the frequency for the debug interface	32
SYStem.LOCK	Lock and tristate the debug port	32
SYStem.MemAccess	Select run-time memory access method	33
SYStem.Mode	Select operation mode	34
SYStem.CONFIG	Configure debugger according to target topology	35
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	46
SYStem.CONFIG.state	Display target configuration	47
SYStem.CONFIG.MDM.Base	Select MDM base address	47
SYStem.CONFIG.MDM.DebugPort	Set core to debug	47
SYStem.CONFIG.MDM.RESet	Reset MDM configuration	48
SYStem.CONFIG.MDM.view	Display MDM configuration	48
SYStem.CONFIG.MDM.UserInst	Set default user BSCAN port	48
TrOnchip Commands		49
TrOnchip.state	Display on-chip trigger window	49
TrOnchip.RESet	Set on-chip trigger to default state	49
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	49
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	50
CPU specific MMU Commands		51
MMU.DUMP	Page wise display of MMU translation table	51
MMU.List	Compact display of MMU translation table	52
MMU.SCAN	Load MMU table from CPU	54
Real-Time Trace		56
SYStem.Option.DTM	Control data trace messages	56
SYStem.Option.QUICKSTOP	Control trace of software breakpoints	56
Configuring your FPGA		57
JTAG Connector		58

Application Notes for MicroBlaze	
Connecting to MicroBlaze Targets for Debug and Trace(app_microblaze.pdf)	1
Connecting the TRACE32 Debugger to the Target	3
Selecting a MicroBlaze Core in the Target	4
Pre-Calculated Multicore Settings for Common Eval Boards	5
Detecting multicore settings	6
Manually Calculating Multicore Settings for Microblaze Cores (one or more FPGAs)	7
Example: Calculating Microblaze Multicore Settings for ML310	9
Modifying Xilinx ML605 for Direct JTAG Access (app_ml605.pdf)	1
Introduction	3
Requirements	3
Description of the Modification	4
Testing the Modified ML605 Board	6
Using ML605 without USB-JTAG Bridge	7
Pinout of the JTAG to Xilinx ML605 Adaptor	8
Miscellaneous Information	9

MIPS

MIPS Debugger and Trace(debugger_mips.pdf)	1
Introduction	5
Brief Overview of Documents for New Users	5
Demo and Start-up Scripts	5
WARNING	6
Quick Start of the EJTAG Debugger	7
Troubleshooting	8
SYStem.Up Errors	8
FAQ	9
CPU specific Implementations	10
Breakpoints	10
Trigger	12
Runtime Measurement	12
Register	12
Memory Classes	13
SPR Memory Overlay	14

MIPS specific SYStem Commands	16
SYStem.CONFIG	Configure debugger according to target topology 16
SYStem.CPU	Select the used CPU 28
SYStem.DETECT.CORENUMBER	Detect core number 29
SYStem.JtagClock	Define JTAG clock 30
SYStem.LOCK	Tristate the JTAG port 31
SYStem.MemAccess	Select run-time memory access method 31
SYStem.Mode	Establish the communication with the target 33
SYStem.Option.Address32	Define address format display 34
SYStem.Option.DCFREEZE	Freeze data cache 34
SYStem.Option.DCREAD	Use DCACHE for data read 35
SYStem.Option.DisMode	Define disassembler mode 35
SYStem.Option.Endianness	Define endianness of target memory 37
SYStem.Option.EnReset	Control target system reset 37
SYStem.Option.EnTRST	Control TAP reset 37
SYStem.Option.HoldReset	Set system reset hold time 38
SYStem.Option.FlowTrace	Define operating mode of RISC TRACE 38
SYStem.Option.FREEZE	Freeze system timer in stop mode 38
SYStem.Option.ICFLUSH	Flush of instruction cache during step and go 39
SYStem.Option.ICREAD	Use ICACHE for program read 39
SYStem.Option.IMASKASM	Disable interrupts while ASM single stepping 39
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 40
SYStem.Option.KEYCODE	Define key code to unsecure processor 40
SYStem.Option.MCBreaksynch	Select break synchronization method 40
SYStem.Option.MMUPhysLogMemaccess	Memory access preferences 41
SYStem.Option.MMUSPACES	Separate address spaces by space IDs 41
SYStem.Option.MonBase	Base address for monitor download routine 43
SYStem.Option.OVERLAY	Enable overlay support 43
SYStem.Option.PROTECTION	Sends an unsecure sequence to the core 44
SYStem.Option.ResBreak	Halt the core after reset 44
SYStem.Option.STEPOCHIP	Use onchip breakpoints for ASM stepping 44
SYStem.Option.STEPSOFT	Use software breakpoints for ASM stepping 45
SYStem.Option.TURBO	Enable fast download 45
SYStem.Option.UnProtect	Unprotect memory addresses 45
SYStem.Option.WaitReset	Set system reset wait time 46
SYStem.Option.WATCHDOG	Disable hardware watchdogs 46
SYStem.RESetOut	Assert nRESET/nSRST on JTAG connector 47
On-chip Breakpoints	48
TrOnchip.AddressMask	Define an address mask 48
TrOnchip.CORERESSET	Halt at reset vector after core reset 48
TrOnchip.RESet	Set on-chip trigger to default state 48
TrOnchip.StepVector	Halt on exception vector during step 48
TrOnchip.UseWatch	Use watchpoints 49

TrOnchip.state	Display on-chip trigger window	49
CPU specific MMU Commands		50
MMU.DUMP	Page wise display of MMU translation table	50
MMU.FORMAT	Define MMU table structure	51
MMU.List	Compact display of MMU translation table	55
MMU.SCAN	Load MMU table from CPU	56
MMU.Set	Set MMU registers	57
MMU.TLB.Set	Set MMU registers	58
MMU.TLBSET	Set MMU registers	58
TCB		59
TCB Control		59
Configuring your FPGA		61
Using JTAG for FPGA configuration		61
EJTAG Connector		62
Mechanical Description of the 14-pin EJTAG Connector		62
Electrical Description of the 14-pin EJTAG Connector		63
Mechanical Description of the 24-pin EJTAG Connector		64
Electrical Description of the 24-pin EJTAG Connector		65
Recommended JTAG Circuit on Target		66
Technical Data Debugger		67
Operation Voltage		67
Mechanical Dimensions		67
Technical Data Trace		69
Operation Voltage		69
Mechanical Dimensions		70

MSP430

MSP430 Debugger	(debugger_msp430.pdf)	1
History		4
Introduction		4
Brief Overview of Documents for New Users		4
Demo and Start-up Scripts		5
Warning		5
General Notes/Target Design Requirements/Recommendations		6
General		6
Target Design Requirements		6
Limitations		6
Contacting Support		7

Quick Start	8
Troubleshooting	11
Communication between Debugger and Processor can not be established	11
FAQ	11
MSP430 Specific Implementations	12
Breakpoints	12
Cycle Counter	15
Runtime Measurement	15
Memory Classes	16
State Storage	16
Trigger Sequencer	16
CPU specific SYStem Commands	17
SYStem.state	Display SYStem.state window 17
SYStem.CONFIG	Configure debugger according to target topology 18
SYStem.CPU	Select the used CPU 19
SYStem.JtagClock	Set jtag clock frequency 19
SYStem.LOCK	Lock and tristate the debug port 19
SYStem.MemAccess	Select run-time memory access method 20
SYStem.Mode	Establish the communication with the target 22
SYStem.Option	Configure debugger behavior 23
SYStem.Option.IMASKASM	Disable interrupts for assembler single steps 23
SYStem.Option.IMASKHLL	Disable interrupts for HLL single steps 23
SYStem.Option.LPMX5	Enable LPMx5 support 23
SYStem.Option.TURBO	Speed up memory access 24
SYStem.Option.TCKTOTEST	Configure clock output pins 24
MSP430 Specific TrOnchip Commands	25
TrOnchip.CONVert	Extend the breakpoint range 25
TrOnchip.RESet	Set on-chip trigger to default state 25
TrOnchip.state	Display on-chip trigger window 26
Low Power Mode Debugging	27
Avoid Loss of Device	27
Supported Low Power Modes	27
Debug Connection	29

M-Core

M-Core Debugger	(debugger_mcore.pdf) 1
Introduction	5
Brief Overview of Documents for New Users	5
Demo and Start-up Scripts	5

Warning	7
Quick Start JTAG/ONCE	8
Breakpoints	10
Software Breakpoints	10
On-chip Breakpoints	10
Breakpoint in ROM	11
Example for Breakpoints	11
Troubleshooting	12
SYStem.Up Errors	12
Memory Access Errors	12
FAQ	12
Configuration	13
Runtime Measurement	13
Memory Classes	13
Memory Coherency	13
M-Core specific SYStem Commands for the Debugger	14
SYStem.CONFIG	14
SYStem.CONFIG	Configure debugger according to target topology 14
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 19
SYStem.CPU	Selects the CPU 20
SYStem.JtagClock	Sets JTAG clock frequency 20
SYStem.LOCK	Tristate the JTAG port 20
SYStem.MemAccess	Select run-time memory access method 21
SYStem.Mode	Establish the communication with the target 22
SYStem.Option.DE	Stop CPU via debug enable line 23
SYStem.Option.DUALPORT	Update all memory displays during runtime 23
SYStem.Option.IMASKASM	Disable interrupts while single stepping 23
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 24
SYStem.Option.PC	Not supported command 24
SYStem.Option.TRST	Use TRST line to reset the TAP controller 24
Trigger On-chip Commands	25
TrOnchip.CYcle	Define access type 25
TrOnchip.A.Address	Define address selector 26
TrOnchip.EXTErnal	Generate a trigger for trace on high pulse on in0 or in1 26
TrOnchip.Mode	Configure unit A and B 27
TrOnchip.RESet	Set on-chip trigger to default state 28
TrOnchip.state	Display on-chip trigger window 28
JTAG Connector	29
Technical Data	30
Operation Voltage	30

NIOS II Debugger and Trace	(debugger_nios.pdf)	1
History		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		6
Warning		7
Troubleshooting		8
SYStem.Up Errors		8
Trace Errors		8
FAQ		8
Quick Start of the ICD Debugger for Nios II		9
1. Prepare the Start		9
2. Configure your FPGA with a Nios II Core (optional)		9
3. Select the Clock for the JTAG Communication		9
4. Configure the Debugger According to the Needs of the Application		10
5. Tell the Debugger where it should use On-chip Breakpoints (optional)		10
6. Enter Debug Mode		10
7. Load the Program		11
8. Initialize Program Counter and Stackpointer		11
9. View the Source Code		11
CPU specific SYStem Settings and Restrictions		13
Restrictions		13
SYStem.CONFIG	Configure multi-core debugger	14
SYStem.CONFIG.CORE	Select core in FPGA	17
SYStem.CONFIG.state	Show multi-core settings	19
SYStem.CONFIG.CPUID	Tell the debugger to which CPU it should connect	20
SYStem.CPU	Select CPU type	20
SYStem.CONFIG.JtagUartNR	Specify JTAG UART component number	20
SYStem.DETECT.ScanCpulDS	Scan which CPU IDs exist in FPGA design	21
SYStem.JtagClock	Select clock for JTAG communication	22
SYStem.LOCK	Lock and tristate the debug port	22
SYStem.MemAccess	Select run-time memory access method	23
SYStem.Mode	Select target reset mode	23
SYStem.Option.BTM	Enable/disable branch trace	24
SYStem.Option.CFGCLK	Set clock frequency for configuration	24
SYStem.Option.DCFLUSH	Flush data cache before “Go”	24
SYStem.Option.DBGALL	Enable/disable debug mode for all cores	25
SYStem.Option.LocalRESet	Assert a local JTAG reset at SYStem.Up	25
SYStem.Option.DTM	Select kind of data trace	26

SYStem.Option.Endianness	Select endianness of core	26
SYStem.Option.FSS	Enable/disable FS2 compatibility mode	27
SYStem.Option.FPH	Enable the disassembly of floating point instructions	27
SYStem.Option.ICFLUSH	Flush instruction cache before “Go”	27
SYStem.Option.IMASKASM	Disable interrupts while single stepping	28
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	28
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	28
SYStem.Option.PIDWidth	Specify size of PID field in the TLB	29
SYStem.Option.QUARTUS	Workaround for QUARTUS II version 13.0	30
SYStem.Option.TOFF	Enable/disable tracetrigger input	30
SYStem.Option.SYNC	Specify frequency of SYNC messages	30
Configuring your FPGA		31
JTAG.LOADRBF	Configure FPGA with RBF file	31
JTAG Uart Support		33
On-chip Breakpoints		34
Program Breakpoints		34
Read and Write Breakpoints		34
Data Breakpoints		35
Trace Control Breakpoints		35
CPU specific MMU Commands		36
MMU.DUMP	Page wise display of MMU translation table	36
MMU.List	Compact display of MMU translation table	38
MMU.SCAN	Load MMU table from CPU	39
TrOnchip Commands		41
TrOnchip.state	Display on-chip trigger window	41
TrOnchip.RESet	Set on-chip trigger to default state	41
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	41
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	42
Memory Classes		43
BDM Connector ICD-NIOS II		44
NIOS II Trace Connector		45
NIOS II Application Note		
NIOS II Instantiating the Off-chip Trace Logic	(app_nios.pdf)	1
Disable Automatic PLL Instantiation		4
Using Extra Registers for Better Timing		5
Multiplexing Trace Outputs of Multiple CPUs		6

PPC400/PPC440 Debugger and Trace (debugger_ppc400.pdf)	1
History	5
Introduction	5
Brief Overview of Documents for New Users	6
Demo and Start-up Scripts	6
Warning	7
Target Design Requirement/Recommendations	8
General	8
Quick Start JTAG	9
Troubleshooting	11
SYStem.Up Errors	11
FAQ	11
Configuration	12
System Overview	12
ICD Trace Extension for PPC400 (ICT)	13
General Fact for PPC403 RiscTrace Use	13
Debugging and Trace Mode	13
What does the PPC403 Trace Mode provide?	13
Used Options for RiscTrace	14
CPU specific Implementations	15
General Restrictions	15
Breakpoints	15
Memory Classes	18
CPU specific SYStem Commands	19
SYStem.BdmClock	Set JTAG clock frequency 19
SYStem.CPU	Select the used CPU 19
SYStem.LOCK	Lock and tristate the debug port 19
SYStem.MemAccess	Select run-time memory access method 20
SYStem.Mode	Select operation mode 20
SYStem.CONFIG.state	Display target configuration 21
SYStem.CONFIG	Configure debugger according to target topology 22
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 26
CPU specific SYStem Commands	27
SYStem.Option.CLOCKX2	Selects the clock for the real-time trace 27
SYStem.Option.DCFREEZE	Freeze contents of cache while debugging 27
SYStem.Option.DCREAD	Read from data cache 28
SYStem.Option.DMALOW	Switch DMA to low priority 28

SYStem.Option.DataTrace	Enable data trace via branch table method	28
SYStem.Option.FREEZERUN	Stop timer in user mode	28
SYStem.Option.FREEZE	Stop timer in debug mode	29
SYStem.Option.FlowTrace	Prepare CPU for real-time trace	29
SYStem.Option.FOLDING	Execute more instructions per cycle	29
SYStem.Option.HOOK	Compare PC to hook address	29
SYStem.Option.ICFLUSH	Invalidate instruction cache	31
SYStem.Option.ICREAD	Read from instruction cache	31
SYStem.Option.IMASKASM	Disable interrupts while single stepping	31
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	31
SYStem.Option.ISOCM	Configure first address of ISOCM	32
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	32
SYStem.Option.NoDebugStop	Disable JTAG stop on debug events	33
SYStem.Option.NoJtagHalt	Disable HALT line	33
SYStem.Option.NOTRAP	Use alternative instruction to enter debug mode	34
SYStem.Option.OVERLAY	Enable overlay support	34
SYStem.Option.ResetMode	Selects the reset mode	35
SYStem.Option.SLOWRESET	Activate SLOWRESET	35
SYStem.Option.STEPSOFT	Use alternative method for ASM single step	35
SYStem.Option.TURBO	Skip additional checks/waits	36
CPU specific TrOnchip Commands		37
TrOnchip.state	Setup window	37
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	38
TrOnchip.DISable	Disable NEXUS trace register control	38
TrOnchip.ENABLE	Use CPU internal trigger logic	38
TrOnchip.RESet	Set on-chip trigger to default state	39
TrOnchip.Set	Trigger sources	39
TrOnchip.TEnable	Set filter for the trace	39
TrOnchip.TOFF	Switch the sampling to the trace to OFF	39
TrOnchip.TON	Switch the sampling to the trace to 'ON'	40
TrOnchip.TTrigger	Set a trigger for the trace	40
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	40
TrOnchip.SYNCHRONOUS	Switches mode for data breakpoints	40
CPU specific MMU Commands		42
MMU.DUMP	Page wise display of MMU translation table	42
MMU.List	Compact display of MMU translation table	44
MMU.SCAN	Load MMU table from CPU	46
MMU.FORMAT	Define MMU table structure	48
MMU.Set.TLB	Create a TLB entry on the TARGET	48
MMU.TLBINIT	Reset TLB	51
MMU.TLBRESET	Reset TLB	51
Debug Connector		52

Mechanical Description	52
Trace Connectors	53
Mictor Connector 38 pin (Version B) for PPC440	53
Mictor Connector 38 pin (Version B) for PPC405	54
Connector 20 pin (Version A) for PPC405 (obsolete)	54
Mictor Connector 38 pin (Version B) for PPC403	55
Connector 20 pin (Version A) for PPC403	55
Application Note for PPC400/PPC440	
Debugging Embedded Cores in Xilinx FPGAs [PPC4xx](app_xilinx_ppc400.pdf)	1
Introduction	3
TRACE32 Software Requirements	3
Related Documents	4
Supported TRACE32 JTAG Cables	5
Physical Connection of the TRACE32 Debugger	6
JTAG Connection via 16-pin PPC Connector	6
JTAG Connection via Configuration JTAG (Xilinx 14 pin connector)	6
Connecting JTAG and Trace Preprocessor	7
Trace Connections via Expansion Headers	7
Supported JTAG Topologies	7
Setting the SYStem.CPU option	9
Multicore Settings for Xilinx FPGAs	10
1st Topology: Separate JTAG Interfaces for FPGA and each PPC Core	12
2nd Topology: Separate FPGA JTAG/ joint PPC JTAG for all PPC Cores	13
3rd Topology: Joint JTAG Interface for the FPGA and all PPC400/PPC440 Cores	15
Design Considerations for Debugging and Tracing	17
Debugging Embedded PPC405 Cores	17
Tracing Embedded PPC405 Cores	17
Debugging Embedded PPC440 Cores	18
Tracing Embedded PPC440 Cores using TRACE32	18
Frequently Asked Questions	19
Virtex2Pro, Virtex4FX, Virtex5FXT: TRACE32 does not display ISOCM memories	19
Flow errors tracing PPC cores on Xilinx ML310 eval board	19
Electrical Interface	19

PQ/MPC500

MPC5xx/8xx Debugger and Trace	(debugger_ppc.pdf)	1
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		6

Warning	7
Quick Start	8
Target Design Requirement/Recommendations	10
General	10
RESET Configuration	11
BDM Termination	12
General Restrictions	13
Troubleshooting	14
FAQ	14
Configuration	15
Breakpoints	17
Software Breakpoints	17
On-chip Breakpoints	17
On-chip Breakpoints on InstructionsROM or FLASH	18
On-chip Breakpoints on Read or Write Accesses	18
Example for Breakpoints	18
Simultaneous FLASH Programming for MPC555	19
Memory Classes	20
Memory Coherency MPC8xx	20
Trace Extension	21
MPC555/MPC553 Pin Multiplexing	21
Troubleshooting MPC500/MPC800 RISC Trace	22
Used Options for RiscTrace	22
General SYStem Commands	23
SYStem.BdmClock	Define the BDM clock speed 23
SYStem.CONFIG	Configure debugger according to target topology 23
SYStem.CPU	Select CPU type 23
SYStem.MemAccess	Select run-time memory access method 24
SYStem.Mode	Establish the communication with the CPU 24
CPU specific SYStem Commands	26
SYStem.Option.BASE	Set base address for on-chip peripherals 26
SYStem.LOADVOC	Load vocabulary for code compression 26
SYStem.Option.BRKNOMSK	Allow program stop in a non-recoverable state 26
SYStem.Option.CCOMP	Enable code compression 27
SYStem.Option.CLEARBE	Clear MSR[BE] on step/go 27
SYStem.Option.CLOCKX2	Select clock for real-time trace 27
SYStem.Option.CSxxx	CS setting for program flow trace 28
SYStem.Option.DCFREEZE	Freeze contents of cache while debugging 28
SYStem.Option.DCREAD	Use DCACHE for data read 29
SYStem.Option.DUALPORT	Run-time memory access for all windows 29

SYStem.Option.FAILSAVE	Special error handling for debug port	29
SYStem.Option.FREEZE	Stop timer in debug mode	30
SYStem.Option.FreezePin	Use alternative signal on the BDM connector	30
SYStem.Option.IBUS	Configure the show cycles for the I-BUS	30
SYStem.Option.ICFLUSH	Flush branch target cache before program start	31
SYStem.Option.ICREAD	Use ICACHE for program read	32
SYStem.Option.IMASKASM	Disable interrupts while single stepping	32
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	32
SYStem.Option.LittleEnd	Selection of little endian mode	33
SYStem.Option.MMUSPACES	Enable space IDs	33
SYStem.Option.NODATA	The external data bus is not connected to trace	34
SYStem.Option.NOTRAP	Use alternative instruction to enter debug mode	34
SYStem.Option.OVERLAY	Enable overlay support	35
SYStem.Option.PPCLittleEnd	Control for PPC little endian	35
SYStem.Option.SCRATCH	Scratch for FPU access	36
SYStem.Option.SIUMCR	SIUMCR setting for the trace	36
SYStem.Option.SLOWLOAD	Alternative data load algorithm	36
SYStem.Option.SLOWRESET	Activate SLOWRESET	36
SYStem.Option.STEPSOFT	Use alternative method for ASM single step	37
SYStem.Option.VECTORS	Define ranges for not-standard interrupt vectors	37
SYStem.Option.VFLS	Use VFLS pins for run/stop detection	37
SYStem.Option.WATCHDOG	Enable software watchdog after SYStem.Up	37
SYStem.state	Display SYStem window	38
CPU specific MMU commands		39
MMU.DUMP	Page wise display of MMU translation table	39
MMU.List	Compact display of MMU translation table	41
MMU.SCAN	Load MMU table from CPU	42
MMU.Set	Set an MMU TLB entry	44
CPU specific TrOnchip Commands		45
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	45
TrOnchip.DISable	Disable NEXUS trace register control	45
TrOnchip.ENable	Enable NEXUS trace register control	45
TrOnchip.G/H	Define data selector	46
TrOnchip.IWx	I-Bus watchpoint	46
TrOnchip.IWx.Count	Event counter for I-Bus watchpoint	46
TrOnchip.IWx.Ibus	Instructions address for I-Bus watchpoint	47
TrOnchip.IWx.Watch	Activate I-Bus watchpoint pin	47
TrOnchip.LWx	L-Bus watchpoint	47
TrOnchip.LW0.Count	Event counter for L-Bus watchpoint	47
TrOnchip.LW0.CYcle	Cycle type for L-Bus watchpoint	48
TrOnchip.LW0.Data	Data selector for L-Bus watchpoint	48
TrOnchip.LW0.Ibus	Instructions address for I-Bus watchpoint	48
TrOnchip.LW0.Lbus	Instructions address for L-Bus watchpoint	49

TrOnchip.LW0.Watch	Activate L-Bus watchpoint pin	49
TrOnchip.RESet	Reset on-chip trigger unit	49
TrOnchip.Set	Stop program execution at specified exception	50
TrOnchip.TEnable	Set filter for the trace	51
TrOnchip.TOFF	Switch the sampling to the trace to OFF	51
TrOnchip.TON	Switch the sampling to the trace to ON	51
TrOnchip.TTrigger	Set a trigger for the trace	52
TrOnchip.VarCONVert	Adjust HLL breakpoint in on-chip resource	52
TrOnchip.state	Display on-chip trigger window	52
BDM Connector		53
10 pin BDM Connector MPC500/MPC800		53
Software Trace as a Flow Trace		54
MPC56x NEXUS Debugger and Trace (nexus_mpc5xx.pdf)		1
Brief Overview of Documents for New Users		6
Warning		7
Quick Start		8
Target Design Requirement/Recommendations		10
General		10
Correct Start-up Sequence for the NEXUS Debugger		10
Special Warning for MPC561 and MPC563		11
AXIOM Evaluation Board		11
Troubleshooting		12
FAQ		12
Configuration		13
Breakpoints		15
Software Breakpoints		15
On-chip Breakpoints		15
On-chip Breakpoints on InstructionsROM or FLASH		16
On-chip Breakpoints on Read or Write Accesses		16
Example for Breakpoints		16
Simultaneous FLASH Programming for MPC56x		17
Memory Classes		18
General SYStem Commands		19
SYStem.CONFIG	Configure debugger according to target topology	19
SYStem.CPU	Select CPU type	19
SYStem.LOCK	Lock and tristate the debug port	19
SYStem.MemAccess	Select run-time memory access method	19
SYStem.Mode	Establish the communication with the CPU	21
CPU specific SYStem Commands		23

SYStem.LOADVOC	Load vocabulary for code compression	23
SYStem.Option.CCOMP	Enable code compression	23
SYStem.Option.CLEARBE	Clear MSR[BE] on step/go	23
SYStem.Option.DCFREEZE	Freeze contents of cache while debugging	23
SYStem.Option.DUALPORT	Run-time memory access for all windows	24
SYStem.Option.EXTVECTORS	Workaround for revision C silicon	24
SYStem.Option.ETASFIX	Workaround for MPC555 RiscTrace	24
SYStem.Option.FAILSAVE	Special error handling for debug port	24
SYStem.Option.FREEZE	Stop timer in debug mode	25
SYStem.Option.HighMemory	Switch on high memory	25
SYStem.Option.ICFLUSH	Flush branch target cache before program start	25
SYStem.Option.IMASKASM	Disable interrupts while single stepping	25
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	26
SYStem.Option.LittleEnd	Selection of little endian mode	26
SYStem.Option.NOTRAP	Use alternative instruction to enter debug mode	26
SYStem.Option.OVERLAY	Enable overlay support	27
SYStem.Option.PPCLittleEnd	Control for PPC little endian	27
SYStem.Option.SCRATCH	Scratch for FPU access	28
SYStem.Option.SLOWRESET	Activate SLOWRESET	28
SYStem.Option.TriState	Control for NEXUS lines	28
SYStem.Option.VECTORS	Define ranges for not-standard interrupt vectors	28
SYStem.Option.WATCHDOG	Enable software watchdog after SYStem.Up	29
SYStem.state	Display SYStem window	29
CPU specific NEXUS Commands		30
NEXUS.BTM	Program trace messaging enable	30
NEXUS.DTM	Enable data trace messaging	30
NEXUS.OFF	Switch the NEXUS trace port off	31
NEXUS.ON	Switch the NEXUS trace port on	31
NEXUS.OTM	Ownership trace messaging enable	31
NEXUS.PortSize	Set trace port width	31
NEXUS.PTM	Enable an enhanced method of program trace	32
NEXUS.PTSM	Enable program trace sync mode	32
NEXUS.QFM	Configure queue flush mode	32
NEXUS.Register	Display NEXUS trace control registers	32
NEXUS.RESet	Reset NEXUS trace port settings	33
NEXUS.state	Display NEXUS port configuration window	33
CPU specific TrOnchip Commands		34
TrOnchip.BusTrigger	Generate a trigger for the internal trigger bus	34
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	35
TrOnchip.DISable	Disable NEXUS trace register control	36
TrOnchip.ENable	Enable NEXUS trace register control	37
TrOnchip.EVTI	Allow the EVTI signal to stop the program execution	37
TrOnchip.EVTO	Use EVTO signal for runtime measurement	38

TrOnchip.EXTernal	Enable trace trigger input of NEXUS adapter	38
TrOnchip.G/H	Define data selector	39
TrOnchip.IWx	I-Bus watchpoint	39
TrOnchip.IWx.Count	Event counter for I-Bus watchpoint	39
TrOnchip.IWx.Ibus	Instructions address for I-Bus watchpoint	40
TrOnchip.IWx.Watch	Activate I-Bus watchpoint pin	40
TrOnchip.LWx	L-Bus watchpoint	40
TrOnchip.LW0.Count	Event counter for L-Bus watchpoint	40
TrOnchip.LW0.CYcle	Cycle type for L-Bus watchpoint	41
TrOnchip.LW0.Data	Data selector for L-Bus watchpoint	41
TrOnchip.LW0.Ibus	Instructions address for I-Bus watchpoint	41
TrOnchip.LW0.Lbus	Instructions address for L-Bus watchpoint	42
TrOnchip.LW0.Watch	Activate L-Bus watchpoint pin	42
TrOnchip.RESet	Reset on-chip trigger unit	42
TrOnchip.Set	Stop program execution at specified exception	43
TrOnchip.TEnable	Set filter for the trace	44
TrOnchip.TOFF	Switch the sampling to the trace to OFF	44
TrOnchip.TON	Switch the sampling to the trace to ON	44
TrOnchip.TTrigger	Set a trigger for the trace	45
TrOnchip.VarCONVert	Adjust HLL breakpoint in on-chip resource	45
TrOnchip.state	Display on-chip trigger window	45
Technical Data		47
Mechanical Dimension		47
Operation Voltage		47

PQII, MPC5200, MPC603/7xx, MPC74xx

PPC600 Family Debugger	(debugger_ppc600.pdf)	1
History		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		7
Warning		8
Signal Level		8
ESD Protection		8
Target Design Requirement/Recommendations		9
General		9
Quick Start		10
Troubleshooting		12
Problems with Memory Access		13
FAQ		13

Configuration	14
System Overview	14
PowerPC 600 Family Specific Implementations	15
Breakpoints	15
Access Classes	21
Cache	22
Little Endian Operation	24
CPU specific SYStem Commands	25
SYStem.BdmClock	Set JTAG frequency 25
SYStem.CPU	Select the CPU type 25
SYStem.LOCK	Lock and tristate the debug port 26
SYStem.MemAccess	Select run-time memory access method 26
SYStem.Mode	Select operation mode 27
SYStem.CONFIG.state	Display target configuration 28
SYStem.CONFIG	Configure debugger according to target topology 29
SYStem.CONFIG.CHKSTPIN	Control pin 8 of debug connector 33
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 34
SYStem.CONFIG.DriverStrength	Configure driver strength of TCK pin 35
SYStem.CONFIG.QACK	Control QACK pin 35
CPU specific System Commands	36
SYStem.Option.BASE	Set base address for on-chip peripherals 36
SYStem.Option.BUS32	Use 32-Bit data-bus mode 37
SYStem.Option.CONFIG	Select RCW configuration 37
SYStem.Option.DCREAD	Read from data cache 38
SYStem.Option.DUALPORT	Implicitly use run-time memory access 38
SYStem.Option.FREEZE	Freeze timebase when core halted 39
SYStem.Option.HoldReset	Set reset hold time 40
SYStem.Option.HOOK	Compare PC to hook address 40
SYStem.Option.HRCWOverRide	Override HRCW on SYStem.Up 41
SYStem.Option.ICFLUSH	Invalidate instruction cache before go/step 41
SYStem.Option.ICREAD	Read from instruction cache 42
SYStem.Option.IMASKASM	Disable interrupts while single stepping 42
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 42
SYStem.Option.IP	Set MSR_IP value for breakpoints / SYStem.Up 43
SYStem.Option.LittleEnd	True little endian mode 43
SYStem.Option.MemProtect	Enable memory access safeguard 43
SYStem.Option.MemSpeed	Configure memory access timing 44
SYStem.Option.MMUSPACES	Separate address spaces by space IDs 44
SYStem.Option.NoDebugStop	Disable JTAG stop on debug events 45
SYStem.Option.NOTRAP	Use alternative software breakpoint instruction 46
SYStem.Option.OVERLAY	Enable overlay support 47
SYStem.Option.PARITY	Generate parity on memory access 47

SYStem.Option.PINTDebug	Program interrupt debugging	48
SYStem.Option.PPCLittleEnd	PPC little endian mode	48
SYStem.Option.PTE	Evaluate PTE table for address translation	49
SYStem.Option.RESetBehavior	Set behavior when target reset detected	49
SYStem.Option.ResetMode	Select reset mode for SYStem.Up	50
SYStem.Option.SLOWRESET	Relaxed reset timing	50
SYStem.Option.STEPSOFT	Use alternative method for ASM single step	51
SYStem.Option.WaitReset	Set reset wait time	52
SYStem.Option.WATCHDOG	Leave software watchdog enabled	53
CPU specific MMU Commands		54
MMU.DUMP	Page wise display of MMU translation table	54
MMU.List	Compact display of MMU translation table	56
MMU.SCAN	Load MMU table from CPU	58
MMU.Set	Write MMU TLB entries to CPU	60
CPU specific BenchMarkCounter Commands		61
BMC.<counter>.FREEZE	Freeze counter in certain core states	61
BMC.FREEZE	Freeze counters while core halted	62
CPU specific TrOnchip Commands		63
TrOnchip.DISable	Disable debug register control	63
TrOnchip.ENABLE	Enable debug register control	63
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	63
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	64
TrOnchip.RESet	Reset on-chip trigger settings	64
TrOnchip.state	Display on-chip trigger window	64
TrOnchip.TEnable	Set filter for the trace	64
TrOnchip.TOFF	Switch the sampling to the trace to OFF	65
TrOnchip.TON	Switch the sampling to the trace to "ON"	65
TrOnchip.TTtrigger	Set a trigger for the trace	65
Mechanical Description		66
JTAG/COP Connector PPC603e/700/MPC8200		66
Technical Data		67

PQIII

PQIII Debugger	(debugger_ppcpq3.pdf)	1
History		5
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		6
Warning		7

Target Design Requirement/Recommendations	8
General	8
Quick Start	9
Troubleshooting	10
SYStem.Up Errors	10
FAQ	11
Configuration	12
System Overview	12
PowerPC MPC85XX/QorIQ specific Implementations	13
Breakpoints	13
Access Classes	17
Cache	18
Debugging Information	21
Programming Flash on MPC85XX / QorIQ P10XX/P20XX, PSC93XX	22
On-chip Trace on MPC85XX/QorIQ	23
PowerPC MPC85XX/QorIQ specific SYStem Commands	25
SYStem.BdmClock	Set BDM clock frequency 25
SYStem.CONFIG.state	Display target configuration 25
SYStem.CONFIG	Configure debugger according to target topology 26
SYStem.CONFIG.CHKSTPIN	Control pin 8 of debug connector 29
SYStem.CONFIG.DriverStrength	Configure driver strength of TCK pin 30
SYStem.CONFIG.QACK	Control QACK pin 30
SYStem.CPU	Select the target processor 31
SYStem.LOCK	Lock and tristate the debug port 31
SYStem.MemAccess	Select run-time memory access method 32
SYStem.Mode	Select operation mode 33
CPU specific SYStem.Option Commands	34
SYStem.Option.CINTDebug	Enable debugging of critical interrupts 34
SYStem.Option.CoreStandBy	On-the-fly breakpoint setup 34
SYStem.Option.DCFREEZE	Prevent data cache line load/flush in debug mode 34
SYStem.Option.DCREAD	Read from data cache 35
SYStem.Option.DUALPORT	Implicitly use run-time memory access 35
SYStem.Option.FREEZE	Freeze system timers on debug events 36
SYStem.Option.HOOK	Compare PC to hook address 36
SYStem.Option.ICFLUSH	Invalidate instruction cache before go and step 36
SYStem.Option.ICREAD	Read from instruction cache 37
SYStem.Option.IMASKASM	Disable interrupts while single stepping 37
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 37
SYStem.Option.MMUSPACES	Separate address spaces by space IDs 38
SYStem.Option.NoDebugStop	Disable JTAG stop on debug events 38
SYStem.Option.NOTRAP	Use alternative software breakpoint instruction 39

SYStem.Option.OVERLAY	Enable overlay support	40
SYStem.Option.PERSTOP	Stop on-chip peripherals in debug mode	40
SYStem.Option.RESetBehavior	Set behavior when target reset detected	41
SYStem.Option.SLOWRESET	Relaxed reset timing	41
SYStem.Option.STEPSOFT	Use alternative method for ASM single step	41
CPU specific MMU Commands		43
MMU.DUMP	Page wise display of MMU translation table	43
MMU.List	Compact display of MMU translation table	45
MMU.SCAN	Load MMU table from CPU	47
MMU.Set	Set an MMU TLB entry	49
CPU specific BenchMarkCounter Commands		50
BMC.FREEZE	Freeze counters while core halted	50
BMC.<counter>.FREEZE	Freeze counter in certain core states	50
BMC.<counter>.SIZE	No function	51
CPU specific TrOnchip Commands		52
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	52
TrOnchip.DISable	Disable NEXUS trace register control	52
TrOnchip.ENABLE	Enable NEXUS trace register control	53
TrOnchip.RESet	Reset on-chip trigger settings	53
TrOnchip.Set	Enable special on-chip breakpoints	54
TrOnchip.VarCONVert	Adjust HLL breakpoint in on-chip resource	55
TrOnchip.state	View on-chip trigger setup window	56
MPC85XX/QorIQ Specific On-chip Trace Settings		57
Onchip.Mode.IFSel	Select interface to be traced	57
JTAG Connector		58
Mechanical Description		58

QorIQ

QorIQ Debugger and NEXUS Trace	(debugger_ppcqorIQ.pdf)	1
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		7
Warning		8
Target Design Recommendations		9
General		9
Quick Start		10
Troubleshooting		11
SYStem.Up Errors		11
FAQ		12

Tool Configuration		13
TRACE32 Debugger		13
TRACE32 Debugger and Trace with Serial Preprocessor		14
TRACE32 Debugger and Trace with PowerTrace Serial		16
PowerPC QorIQ specific Implementations		20
Breakpoints		20
Access Classes		24
Cache		26
Debugging Information		29
Trace Information		40
CPU specific SYStem Commands		48
SYStem.BdmClock	Set debug clock frequency	48
SYStem.CONFIG.state	Display target configuration	49
SYStem.CONFIG	Configure debugger according to target topology	50
SYStem.CONFIG.CHKSTPIN	Control pin 8 of debug connector	53
SYStem.CONFIG.DriverStrength	Configure driver strength of TCK pin	54
SYStem.CONFIG.QACK	Control QACK pin	54
SYStem.CPU	Select the CPU type	55
SYStem.LOCK	Lock and tristate the debug port	55
SYStem.MemAccess	Select run-time memory access method	55
SYStem.Mode	Select operation mode	57
CPU specific SYStem.Option Commands		58
SYStem.Option.Address32	Define address format display	58
SYStem.Option.DCFREEZE	Data cache state frozen while core halted	58
SYStem.Option.DCREAD	Read from data cache	59
SYStem.Option.DUALPORT	Implicitly use run-time memory access	60
SYStem.Option.FREEZE	Freeze system timers on debug events	60
SYStem.Option.HOOK	Compare PC to hook address	60
SYStem.Option.HRCWOVerRide	Override RCW during SYStem.Up	61
SYStem.Option.ICFLUSH	Invalidate instruction cache before go and step	61
SYStem.Option.ICREAD	Read from instruction cache	61
SYStem.Option.IMASKASM	Disable interrupts while single stepping	62
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	62
SYStem.Option.MACHINESPACES	Address extension for guest OSes	62
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	63
SYStem.Option.NoDebugStop	Disable JTAG stop on debug events	65
SYStem.Option.OVERLAY	Enable overlay support	66
SYStem.Option.RESetBehavior	Set behavior when target reset detected	67
SYStem.Option.SLOWRESET	Relaxed reset timing	67
SYStem.Option.STEPSOFT	Use alternative method for ASM single step	68
SYStem.Option.TranslationSPACE	Identify user and hypervisor modes	68
SYStem.Option.ZoneSPACES	Enable symbol management for zones	69

CPU specific MMU Commands		73
MMU.DUMP	Page wise display of MMU translation table	73
MMU.FORMAT	Define MMU table structure	76
MMU.List	Compact display of MMU translation table	81
MMU.SCAN	Load MMU table from CPU	83
MMU.Set	Set an MMU TLB entry	85
CPU specific BenchMarkCounter Commands		86
BMC.FREEZE	Freeze counters while core halted	86
BMC.Trace	Trace performance monitor events	86
BMC.<counter>.FREEZE	Freeze counter in certain core states	87
CPU specific TrOnchip Commands		88
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	88
TrOnchip.RESet	Reset on-chip trigger settings	89
TrOnchip.Set	Enable special on-chip breakpoints	89
TrOnchip.VarCONVert	Adjust HLL breakpoint in on-chip resource	90
TrOnchip.state	View on-chip trigger setup window	91
Nexus and Trace specific commands		92
DDRTrace.List	List DDR trace contents	92
DQMTrace.List	List DQM trace contents	92
NEXUS.BTM	Enable program trace messaging	93
NEXUS.CoreENable	Core specific trace configuration	93
NEXUS.DDRConfig.ADDRessfilter	Filter Nexus DDR messages	94
NEXUS.DDRConfig.Controller	Configure Nexus DDR message type	94
NEXUS.DQM	Enable data acquisition messaging	95
NEXUS.LaneMapping	Logical to physical lane mapping	96
NEXUS.LaneMapping.APPLY	Apply logical to physical lane mapping	96
NEXUS.LaneMapping.SetLane	Configure logical to physical lane mapping	96
NEXUS.OCeaNport.Mode	Configure Nexus OCeaN message type	97
NEXUS.OCeaNport<index>.TraceSElect	Select Nexus OCeaN trace type	98
NEXUS.OFF	Switch the Nexus trace port off	98
NEXUS.ON	Switch the Nexus trace port on	99
NEXUS.OTM	Enable ownership trace messaging	99
NEXUS.PortMode	Set Nexus trace port frequency	100
NEXUS.PortSize	Set trace port width	100
NEXUS.POTD	Disable periodic ownership trace	101
NEXUS.PTCM	Enable program trace correlation messages	101
NEXUS.PTFGS	Program trace mark	101
NEXUS.PTFPMM	Program trace mark	102
NEXUS.PTFPR	Program trace mark	102
NEXUS.PTMARK	Program trace mark	103
NEXUS.RefClock	Enable Aurora reference clock	103
NEXUS.Register	Display NEXUS trace control registers	103

NEXUS.RESet	Reset Nexus trace port settings	104
NEXUS.SerDesCFG	Enable SerDes PLL control register manipulation	104
NEXUS.SerDesCFG.FRATE	Select frequency of SerDes PLL VCO	104
NEXUS.SerDesCFG.REFCLK	Select frequency of SerDes reference clock	105
NEXUS.Spen<messagetype>	Enable message suppression	105
NEXUS.STALL	Stall the program execution when FIFO level is reached	106
NEXUS.state	Display Nexus port configuration window	107
NEXUS.SuppRTHReshold	Set fill level for message suppression	107
NEXUS.TimeStamps	Append target timestamps to Nexus messages	108
NEXUS.USEPORT	Define used PCIe controller for PCIe trace	108
NEXUS.WTM	Enable watchpoint messaging	108
OCeaNTrace.List	List OCeaN trace contents	109
Onchip specific Commands		110
Onchip.TBARange	Configure on-chip trace base address range	110
Filters and Triggers for the Nexus Trace		111
JTAG Connector		113
Mechanical Description		113

Qorivva MPC5xxx/SPC5xx

Qorivva MPC5xxx/SPC5xx Debugger and NEXUS Trace	 (debugger_mpc5500.pdf)	1
History		8
Introduction		9
Available Tools		9
Software Installation		12
Hardware Installation		13
ESD Protection Considerations		16
Demo and Start-up Scripts		16
Debug Cable / Nexus Adapter Versions and Detection		17
Brief Overview of Documents for New Users		18
Target Design Requirement/Recommendations		20
General (ICD Debugger)		20
Quick Start		21
Run Program from On-chip SRAM		21
Run Program from FLASH		23
Connect to Running Program (hot plug-in)		24
FAQ		25
Debugging		26
Breakpoints		26
Memory Access		31

Cache Debugging Support	34	
Support for Peripheral Modules	37	
Debugging and Tracing Through Reset	39	
Multicore Debugging	41	
Watchdog Timer Support	44	
Censorship Unlock	46	
Non-secure boot (S32R294)	50	
Troubleshooting Debug	51	
Tracing	52	
e200 PCFIFO On-chip Trace	52	
MPC57XX/SPC57X/SPC58X NEXUS On-chip Trace (trace-to-memory)	53	
External Trace Ports (Parallel NEXUS/Aurora NEXUS)	54	
Tracing the Program Flow	55	
Tracing of Data (read/write) Transactions	56	
Tracing of Context Switches	57	
Trace Based Run-time Measurement / Timestamping	58	
Trace Filtering and Triggering with Debug Events	60	
Tracing Peripheral Modules / Bus Masters	65	
Trace Filtering and Triggering Features Provided by TRACE32	66	
Troubleshooting Trace	66	
FLASH Programming Support	68	
FLASH Programming Scripts	68	
Requirements due to FLASH ECC Protection	70	
Programming the RCHW or Boot Header	71	
Programming the Shadow Row	71	
Programming Serial Boot Password and Censorship Word	73	
TEST / UTEST / OTP FLASH Programming	74	
Brownout Depletion Recovery	76	
Troubleshooting FLASH	76	
Command Reference: SYStem Commands	78	
SYStem.BdmClock	Set BDM clock frequency	78
SYStem.CONFIG.state	Display target configuration	79
SYStem.CONFIG	Configure debugger according to target topology	80
SYStem.CONFIG.DEBUGPORTTYPE	Set debug cable interface mode	85
SYStem.CONFIG.EXTWDTDIS	Disable external watchdog	86
SYStem.CONFIG.PortSHaRing	Control sharing of debug port with other tool	87
SYStem.CPU	Select the target processor	87
SYStem.LOCK	Lock and tristate the debug port	88
SYStem.MemAccess	Select run-time memory access method	88
SYStem.Mode	Select operation mode	90
Command Reference: SYStem.Option Commands	91	
SYStem.Option.BISTRUN	Debug with BIST enabled	91

SYStem.Option.CoreStandBy	On-the-fly breakpoint and trace setup	91
SYStem.Option.DCFREEZE	Data cache state frozen while core halted	91
SYStem.Option.DCREAD	Read from data cache	92
SYStem.Option.DISableResetEscalation	Control reset escalation disabling	92
SYStem.Option.DISableShortSequence	Short reset sequence handling	93
SYStem.Option.DisMode	Disassembler operation mode	93
SYStem.Option.DUALPORT	Implicitly use run-time memory access	94
SYStem.Option.FASTACCESS	Special operation mode for fast run control	95
SYStem.Option.FREEZE	Freeze system timers on debug events	95
SYStem.Option.HoldReset	Set reset hold time	96
SYStem.Option.ICFLUSH	Invalidate instruction cache before go and step	96
SYStem.Option.ICREAD	Read from instruction cache	96
SYStem.Option.IMASKASM	Disable interrupts while single stepping	97
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	97
SYStem.Option.KEYCODE	Inhibit censorship protection	97
SYStem.Option.LPMDDebug	Enable low power mode debug handshake	99
SYStem.Option.LockStepDebug	Enable lock-step core register access	100
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	100
SYStem.Option.NexusMemoryCoherency	Coherent NEXUS mem-access	101
SYStem.Option.NoDebugStop	Disable JTAG stop on debug events	102
SYStem.Option.NoJtagRdy	Do not evaluate JTAG_RDY signal	102
SYStem.Option.NOTRAP	Use brkpt instruction for software breakpoints	103
SYStem.Option.OVERLAY	Enable overlay support	104
SYStem.Option.PC	Set fetch address debug actions	104
SYStem.Option.RESetBehavior	Set behavior when target reset detected	105
SYStem.Option.ResBreak	Halt the core while reset asserted	105
SYStem.Option.ResetDetection	Configure reset detection method	106
SYStem.Option.ResetMode	Select reset mode for SYStem.Up	107
SYStem.Option.SLOWRESET	Relaxed reset timing	107
SYStem.Option.STEPSOFT	Use alternative method for ASM single step	108
SYStem.Option.TDOSElect	Select TDO source of lock step core pair	108
SYStem.Option.VECTORS	Specify interrupt vector table address	108
SYStem.Option.WaitBoomRom	Wait for BootROM completion	109
SYStem.Option.WaitReset	Set reset wait time	109
SYStem.Option.WATCHDOG	Debug with software watchdog timer	111
Command Reference: MMU Commands		113
MMU.DUMP	Page wise display of MMU translation table	113
MMU.List	Compact display of MMU translation table	115
MMU.SCAN	Load MMU table from CPU	117
MMU.Set	Set an MMU TLB entry	119
Command Reference: BenchMarkCounter		120
BMC.<counter>.ATOB	Enable event triggered counter start and stop	120
BMC.<counter>.FREEZE	Freeze counter in certain core states	123

BMC.FREEZE	Freeze counters while core halted	124
BMC.Trace	Trace performance monitor events	124
Command Reference: TrOnchip		125
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	125
TrOnchip.EDBRAC0	Assign debug events to target software	126
TrOnchip.EVTEN	Enable EVTI and EVTO pins	127
TrOnchip.RESet	Reset on-chip trigger settings	128
TrOnchip.Set	Enable special on-chip breakpoints	128
TrOnchip.VarCONVert	Set single address breakpoint for scalar	129
TrOnchip.state	View on-chip trigger setup window	130
Command Reference: Onchip		131
Onchip.TBARange	Set on-chip trace buffer address range	131
Command Reference: NEXUS		132
NEXUS.BTM	Enable program trace messaging	132
NEXUS.CLIENT<x>.BUSSEL	Set NXMC target RAM	132
NEXUS.CLIENT<x>.MODE	Set data trace mode of nexus client	132
NEXUS.CLIENT<x>.SELECT	Select a nexus client for data tracing	133
NEXUS.CLIENT3.SPTACQMASTER	Trace individual SPT masters	133
NEXUS.CoreENable	Enable core tracing for dedicated cores in SMP	133
NEXUS.DDR	Enable NEXUS double data rate mode	134
NEXUS.DMADTM	Enable DMA data trace messaging	134
NEXUS.DTM	Enable data trace messaging	135
NEXUS.DTMARK	Data trace mark	135
NEXUS.DTMWhileHalted	Data trace messaging while core halted	136
NEXUS.DQM	Enable data acquisition messaging	136
NEXUS.FRAYDTM	Enable FlexRay data trace messaging	136
NEXUS.HTM	Enable branch history messaging	137
NEXUS.OFF	Switch the NEXUS trace port off	137
NEXUS.ON	Switch the NEXUS trace port on	138
NEXUS.OTM	Enable ownership trace messaging	139
NEXUS.PCRCONFIG	Configure NEXUS PCR for tracing	139
NEXUS.PINCR	Define DCI PINCR register value	140
NEXUS.PortMode	Set NEXUS trace port frequency	140
NEXUS.PortSize	Set trace port width	141
NEXUS.POTD	Periodic ownership trace disable	141
NEXUS.PTCM	Enable program trace correlation messages	142
NEXUS.PTMARK	Program trace mark	142
NEXUS.RefClock	Enable Aurora reference clock	143
NEXUS.Register	Display NEXUS trace control registers	143
NEXUS.RESet	Reset NEXUS trace port settings	143
NEXUS.RFMHISTBUGFIX	Double RFM workaround	143
NEXUS.SmartTrace	Enable smart trace analysis	144

NEXUS.Spen<messagetype>	Enable message suppression	144
NEXUS.STALL	Stall the program execution when FIFO full	144
NEXUS.state	Display NEXUS port configuration window	145
NEXUS.SupprTHReshold	Set fill level for message suppression	145
NEXUS.TimeStamps	Enable on-chip timestamp generation	145
NEXUS.WTM	Enable watchpoint messaging	146
Nexus specific TrOnchip Commands		147
TrOnchip.Alpha	Set special breakpoint function	147
TrOnchip.Beta	Set special breakpoint function	147
TrOnchip.Charly	Set special breakpoint function	148
TrOnchip.Delta	Set special breakpoint function	148
TrOnchip.DISable	Disable NEXUS trace register control	148
TrOnchip.Echo	Set special breakpoint function	148
TrOnchip.ENable	Enable NEXUS trace register control	149
TrOnchip.EVTI	Allow the EVTI signal to stop the program execution	149
TrOnchip.EVTO	Use EVTO signal for runtime measurement	149
TrOnchip.EXTernal	Enable trace trigger input of NEXUS adapter	150
TrOnchip.Out0	Select OUT0 pin signal source	150
TrOnchip.Out1	Select OUT1 pin signal source	151
TrOnchip.TOOLIO2	Select TOOLIO2 pin signal source	152
TrOnchip.TRaceControl	Trace control with special debug events	153
Debug and Trace Connectors		154
14-pin JTAG/OnCE Connector (JTAG)		154
AUTO26 Connector (JTAG)		154
10-pin ECU14 Connector (with converter LA-3843)		155
38-pin Mictor Connector (NEXUS parallel)		155
50-pin SAMTEC ERF8 Connector (NEXUS parallel)		156
51-pin GlenAir / ROBUST Connector (NEXUS parallel)		157
34-pin SAMTEC ERF8 Connector (Aurora NEXUS)		158
Mechanical Dimensions		159
Technical Data		168
Operation Voltage		168
Operation Frequency		168
Application Note for Nexus MPC5xxx		
Complex Trigger Unit for Nexus MPC5xxx	(app_ctu_mpc5xxx.pdf)	1
History		4
Introduction		5
Program Structure		7
Conditions		8
Declaration Reference		10

ADDRESS	Address selectors	10
EVENTCOUNTER	Event counter	11
FLAGS	Flags	13
HWME	Hardware message events	14
OTME	Ownership trace message events	14
TIMECOUNTER	Time counter	15
TRIG	External triggers	17
Instruction Reference		19
BREAK	Analyzer stop	19
Bus	Bus trigger	19
CONTinue	Sequential level switching	20
Counter	Counter control	20
Flag	Flag control	23
GOTO	Level switching	24
Mark	Recording markers	24
Out	Output control	25
Sample	Recording control	26
Trigger	Trigger control	28
CTU Programming Examples		30
Data Trace Message based events		30
Example: Trace trigger on data value		31
Example: Program break on data value		31
Watchpoint hit message based events		32
Example: Runtime measurement with markers		34
Example: Program break based function runtime		37
Using external signals with the CTU		38
Example: Record single message on rising edge of trigger input		39
Example: Program break based on pulse interval of IN input		40
Appendix: Complex Trigger Unit Keyword Reference		41

RH850

RH850 Debugger and Trace(debugger_rh850.pdf)	1
History	7
Introduction	8
Available Tools	8
Software Installation	10
Related Documents	10
Demo and Start-up Scripts	11
Brief Overview of Documents for New Users	11
Warning	12

Useful Tips	13	
Application Starts Running at SYStem.Up	13	
Greenhills Compiler	14	
Stop Timers and Peripherals during application-break	14	
Location of Debug Connector	14	
Reset Line	14	
Debugging the STOP and DeepSTOP Mode	15	
Configuration	16	
System Overview	16	
Single Core Debugging - Quick Start	17	
Debug from Reset	17	
Connect to Running Program (Hot Plug-In)	19	
Troubleshooting	20	
SYStem.Up Errors	20	
FAQ	20	
Debugging	21	
RH850 Debug Interface Modes	21	
Breakpoints	24	
Access Classes	26	
Support for Peripheral Modules	29	
Runtime Measurement	29	
Multicore Debugging	30	
FLASH Programming Support	34	
Tracing	36	
SFT Trace via LPD4	36	
NEXUS On-chip Trace	36	
External Trace Ports (Parallel NEXUS/Aurora NEXUS)	36	
Tracing the Program Flow	37	
Tracing of Data (read/write) Transactions	38	
Trace Filtering and Triggering with Debug Events	39	
Tracing Peripheral Modules / Bus Masters	43	
SFT Software Trace	44	
SFT Software Trace to On-chip Trace	44	
SFT Software Trace via LPD4 debug port	45	
CPU specific SYStem Commands	46	
SYStem.BAUDRATE	Baudrate setting	46
SYStem.CONFIG.state	Display target configuration	46
SYStem.CONFIG	Configure debugger according to target topology	47
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	51
SYStem.CONFIG.DEBUGPORT	Select target interface	52

SYStem.CONFIG.DEBUGPORTTYPE	Select debug port type	52
SYStem.CONFIG.EXTWDTDIS	Disable external watchdog	53
SYStem.CONFIG.PortSHaRing	Control sharing of debug port with other tool	54
SYStem.CORECLOCK	Core clock frequency	54
SYStem.CPU	CPU type selection	55
SYStem.JtagClock	JTAG clock selection	55
SYStem.LOCK	Lock and tristate the debug port	55
SYStem.MemAccess	Select run-time memory access method	56
SYStem.Mode	System mode selection	57
SYStem.OSCCLOCK	Oscillator clock frequency	58
SYStem.RESetOut	Reset target without reset of debug port	58
CPU specific SYStem.Option Commands		59
SYStem.Option.CFU	CalibrationFunctionUnit support	59
SYStem.Option.DOWNMODE	Behavior of SYStem.Mode Down	59
SYStem.Option.DUALPORT	Implicitly use run-time memory access	60
SYStem.Option.FLMD0	FLMD0 pin default level	60
SYStem.Option.HoldReset	Set reset hold time	60
SYStem.Option.ICUS	ICU-S enable	62
SYStem.Option.IDSET	Program KeyCodes to CPU option bytes	63
SYStem.Option.IMASKASM	Interrupt disable	63
SYStem.Option.IMASKHLL	Interrupt disable	63
SYStem.Option.KEYCODE	Keycode (G3Kx cores only)	64
SYStem.Option.MACHINESPACES	Address extension for guest OSes	64
SYStem.Option.OCDID	OnChipDebugID setting	65
SYStem.Option.CFID	CodeFlashID setting	65
SYStem.Option.DFID	DataFlashID setting	65
SYStem.Option.OPTIONBYTE	Option-byte setting	66
SYStem.Option.OPTIONBYTE8	Option-byte setting	66
SYStem.Option.CIDA	Customer-ID A setting	66
SYStem.Option.CIDB	Customer-ID B setting	67
SYStem.Option.CIDC	Customer-ID C setting	67
SYStem.Option.PERSTOP	Disable CPU peripherals if stopped	68
SYStem.Option.RESetBehavior	Set behavior when target reset detected	69
SYStem.Option.ResetDetection	Configure reset detection method	70
SYStem.Option.RDYLINE	RDY pin available	70
SYStem.Option.SLOWRESET	Timeout for ResetRiseTime	71
SYStem.Option.WaitReset	Set reset wait time	71
SYStem.Option (Exception Lines Enable)		72
SYStem.Option.CPINT	CPINT line enable	72
SYStem.Option.REQest	Request line enable	72
SYStem.Option.RESET	Reset line enable	72
SYStem.Option.STOP	Stop line enable	73
SYStem.Option.WAIT	Wait line enable	73

CPU specific BenchMarkCounter Commands	74
BMC.<counter>.ATOB	Enable event triggered counter start and stop 75
BMC.<counter>.EVENT	Configure the performance monitor 76
BMC.<counter>.TRIGMODE	BMC trigger mode 77
BMC.<counter>.TRIGVAL	BMC trigger value 77
CPU specific TrOnchip Commands	78
TrOnchip.CONVert	Allow extension of address range of breakpoint 78
TrOnchip.EVTEN	Enable 'EVTO-' trigger input (Aurora trace only) 79
TrOnchip.RESet	Set on-chip trigger to default state 79
TrOnchip.SIZE	Trigger on byte, word, long memory accesses 80
TrOnchip.state	Display on-chip trigger window 80
TrOnchip.VarCONVert	Convert breakpoints on scalar variables 81
Command Reference: NEXUS	83
NEXUS.BTM	Program trace messaging enable 83
NEXUS.CoreENable	Core specific trace configuration 83
NEXUS.CLIENT<x>.MODE	Set data trace mode of nexus client 83
NEXUS.CLIENT<x>.SELECT	Select a nexus client for data tracing 84
NEXUS.DTM	Data trace messaging enable 84
NEXUS.OFF	Disable NEXUS register access 85
NEXUS.ON	Switch the NEXUS trace port on 85
NEXUS.PortMode	Set NEXUS trace port frequency 85
NEXUS.PortSize	Set trace port width 86
NEXUS.RESet	Reset NEXUS trace port settings 86
NEXUS.SFT	Software trace messaging enable 86
NEXUS.SUSpend	Stall the program execution when FIFO full 86
NEXUS.SYNC	Address-sync trace messaging enable 87
NEXUS.SyncPeriod	Set period of timestamp sync messages 87
NEXUS.state	Display NEXUS port configuration window 87
NEXUS.TimeStamps	On-chip timestamp generation enable 88
Nexus specific TrOnchip Commands	89
TrOnchip.Alpha	Set special breakpoint function 89
TrOnchip.Beta	Set special breakpoint function 90
TrOnchip.Charly	Set special breakpoint function 90
TrOnchip.Delta	Set special breakpoint function 91
TrOnchip.Echo	Set special breakpoint function 91
CPU specific Functions	92
CPU.BASEFAMILY()	CPU family 92
CPU.DEVICEID()	Value of the device-id 92
CPU.SUBFAMILY()	CPU subfamily 92
SYStem.BAUDRATE()	Value of baudrate 93
SYStem.CORECLOCK()	Core clock frequency 93
SYStem.OSCCLOCK()	Oscillator clock frequency 93

SYStem.CFID()	Values of CodeFlashID	93
SYStem.DFID()	Values of DataFlashID	94
SYStem.OCDID()	Values of OnChipDebugID	94
SYStem.OPBT()	Values of Option-bytes	94
SYStem.OPBT8()	Values of Option-bytes	95
SYStem.RESETDETECTION()	Reset detection method	95
Debug Connector		96
Debug Connector 14 pin 100mil		96
Debug Connector 26		97
Trace Connectors and Adapters		98
Adapter for RH850 (LA-3561)		98
Parallel NEXUS Connector (Debug and Trace)		100
Aurora NEXUS SAMTEC 34-pin (Debug and Trace)		101
Aurora NEXUS SAMTEC 40-pin (Trace only)		102
Aurora NEXUS SAMTEC 46-pin (Debug and Trace)		103
RH850 Application Notes		
Application Note Benchmark Counter RH850	(app_rh850_bmc.pdf)	1
Introduction		3
Intended Audience		3
Prerequisites		3
Related Documents		3
Measuring Runtimes		4
Measuring a Single Function		4
Breaking when a Function Takes Longer than a Specified Time		7
Measuring Multiple Functions		9
Measuring a Task or Thread		12
Multi-Core Considerations		14

RISC-V

RISC-V Debugger	(debugger_riscv.pdf)	1
History		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Script		7
List of Abbreviations and Definitions		8
Warning		9
Quick Start of the JTAG Debugger		10
Quick Start for Debug Module Configuration		15

Debug Module Access via JTAG-DTM	16
Debug Module Access via Debug Bus	17
Quick Start for Multicore Debugging	19
SMP Debugging	19
SMP Debugging - Selective	20
Homogeneous SMP/AMP Debugging	21
Heterogeneous SMP/AMP Debugging	22
Troubleshooting	23
Communication between Debugger and Processor cannot be established	23
FAQ	24
RISC-V Specific Implementations	25
Debug Specification for External Debug Support	25
Access Classes	26
Breakpoints	32
Floating-Point Extensions	35
Hardware Performance Monitor	36
Hart State: Unavailable	37
Semihosting	38
Vector Extension	39
CPU specific SETUP Command	40
SETUP.DIS	Disassembler configuration 40
CPU specific SYStem Commands	42
SYStem.CONFIG.state	Display target configuration 42
SYStem.CONFIG	Configure debugger according to target topology 43
SYStem.CONFIG.HART.INDEX	Set hart index 62
SYStem.CPU	Select the CPU to be debugged 63
SYStem.JtagClock	Define JTAG frequency 64
SYStem.LOCK	Tristate the JTAG port 65
SYStem.MemAccess	Select run-time memory access method 66
SYStem.MemAccessStop	Memory access while stopped 68
SYStem.Mode	Establish the communication with the target 69
SYStem.Option	Special setup 71
SYStem.Option.Address32	Define address format display 71
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits 71
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP 72
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits 72
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits 73
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP 74
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check 74
SYStem.Option.DAPREMAP	Rearrange DAP memory map 75
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP 75

SYStem.Option.DEBUGPORTOptions	Options for debug port handling	76
SYStem.Option.DMACTiveRESet	Allow debugger to reset DM via dmactive	77
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	77
SYStem.Option.HARVARD	Use Harvard memory model	78
SYStem.Option.HoldReset	Set reset duration time	78
SYStem.Option.IMASKASM	Disable interrupts while single stepping	79
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	79
SYStem.Option.IsaEXTension	Manually configure support for ISA extensions	80
SYStem.Option.KeepAlive	Keep hart available for debugger	80
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	81
SYStem.Option.ResetDetection	Choose method to detect a target reset	82
SYStem.Option.ResetMode	Select reset method	82
SYStem.Option.SOFTLONG	Use 32-bit access to set SW breakpoints	85
SYStem.Option.SYSDownACTion	Define action during SYStem.Down	86
SYStem.Option.TRST	Allow debugger to drive TRST	86
SYStem.Option.WaitReset	Set reset wait time	87
SYStem.Option.ZoneSPACES	Enable symbol management for zones	88
SYStem.state	Display SYStem.state window	89
CPU specific FPU Command		90
FPU.Set	Write to FPU register	90
CPU specific MMU Commands		91
MMU.DUMP	Page wise display of MMU translation table	91
MMU.List	Compact display of MMU translation table	93
MMU.SCAN	Load MMU table from CPU	94
CPU specific TrOnchip Commands		96
Target Adaption		97
Connector Type and Pinout		97

RX Debugger

RX Debugger	(debugger_rx.pdf)	1
Introduction		4
Brief Overview of Documents for New Users		4
Demo and Start-up Scripts		5
Warning		6
Application Note		7
Location of Debug Connector		7
Reset Line		7
Enable Debug Mode		8
Enable AUD Trace lines		8
Quick Start JTAG		9

Troubleshooting	11
SYStem.Up Errors	11
Trace Errors	12
FAQ	12
Configuration	13
System Overview	13
CPU specific SYStem Settings	14
SYStem.CONFIG	Configure debugger according to target topology
SYStem.CONFIG.CORE	Assign core to TRACE32 instance
SYStem.CONFIG.state	Display target configuration
SYStem.CPU	CPU type selection
SYStem.JtagClock	JTAG clock selection
SYStem.LOCK	JTAG lock
SYStem.MemAccess	Select run-time memory access method
SYStem.Mode	System mode selection
SYStem.Option.BigEndian	Define byte order (endianness)
SYStem.Option.IMASKASM	Interrupt disable
SYStem.Option.IMASKHLL	Interrupt disable
SYStem.Option.KEYCODE	Keycode
Breakpoints	23
Software Breakpoints	23
On-chip Breakpoints	23
Breakpoint in ROM	24
Example for Breakpoints	24
TrOnchip Commands	25
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource
TrOnchip.RESet	Set on-chip trigger to default state
TrOnchip.SEQ	Sequential breakpoints
TrOnchip.state	Display on-chip trigger window
Memory Classes	27
Trace	28
AUD-Trace	28
SYStem.Option.AUDBT	AUD branch trace enable
SYStem.Option.AUDDT	AUD data trace enable
SYStem.Option.AUDRTT	AUD real time trace enable
SYStem.Option.AUDClock	AUD clock select
On-chip Trace	30
Onchip.Mode.ProgramTrace	Program flow trace enable
Onchip.Mode.DataTrace	Data trace enable
On-chip Performance Analysis	31

Runtime Measurement	32
JTAG Connector	33
AUD Trace Connector	34

SDMA

SDMA Debugger	(debugger_sdma.pdf)	1
Warning		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		7
Configuration		9
System Overview		9
Quick Start of the Debugger		10
Troubleshooting		13
Communication between Debugger and Processor can not be established		13
FAQ		14
SDMA specific Implementations		15
Memory Classes		15
Breakpoints		15
On-chip Trace		16
Special Hints, Restrictions, and Known Problems		16
SDMA specific SYStem Commands		17
SYStem.CONFIG.state	Display target configuration	17
SYStem.CONFIG	Configure debugger according to target topology	18
SYStem.CPU	Select the used CPU	29
SYStem.JtagClock	Define the frequency of the debug port	30
SYStem.LOCK	Lock and tristate the debug port	30
SYStem.MemAccess	Select run-time memory access method	31
SYStem.Mode	Establish the communication with the target	32
SYStem.Option	Special setup	34
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	35
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	35
SYStem.Option.DAPREMAP	Rearrange DAP memory map	36
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	36
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	37
SYStem.Option.DUALPORT	Implicitly use run-time memory access	38
SYStem.Option.IMASKASM	Disable interrupts while single stepping	38
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	38
CPU specific TrOnchip Commands		39

Target Adaption	40
Probe Cables	40
Connector Type and Pinout	40

StarCore

StarCore Debugger and Trace	(debugger_starcore.pdf)	1
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		6
Warning		8
Quick Start		9
Troubleshooting		12
SYStem.Up Errors		12
Memory Access Errors		13
NEXUS Flow Errors and FIFO Overflow Messages		14
FAQ		15
Configuration		16
CPU specific SYStem Settings and Restrictions		17
SYStem.CLOCK	Setup core clock	17
SYStem.CONFIG.state	Display target configuration	17
SYStem.CONFIG	Configure debugger according to target topology	18
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	22
SYStem.CPU	Select the used CPU	23
SYStem.LOCK	Lock and tristate the debug port	23
SYStem.MemAccess	Select run-time memory access method	23
SYStem.Mode	Establish the communication with the target	25
SYStem.Option.BASE	Sets the SUI base address	25
SYStem.Option.DCFLUSH	Data cache flush before step/run	26
SYStem.Option.DTM	Enables data trace messages	26
SYStem.Option.EnReset	Allow the debugger to drive nRESET/nSRST	26
SYStem.Option.EnTrst	Allow debugger to drive TRST	27
SYStem.Option.HalfRate	Enable Nexus DDR mode	27
SYStem.Option.ICFLUSH	Instruction cache flush before step/run	28
SYStem.Option.IMASKASM	Disable interrupts while single stepping	28
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	28
SYStem.Option.IPLDI	Sets interrupt mask strategy	29
SYStem.Option.LittleEnd	Switches between endian modes	29
SYStem.Option.MCKO	Nexus output clock ratio	29
SYStem.Option.MPU	MPU disabled	30
SYStem.Option.Nexus	Nexus port width	30

SYStem.Option.OCEBASE	Base address for OnCE registers	30
SYStem.Option.OCECORE	OnCE selection	31
SYStem.Option.OVC	Trace message overrun control	31
SYStem.Option.PTM	Enables program trace messages	31
SYStem.Option.SAMPLE	Adjust NEXUS sample point	32
SYStem.Option.SLOWPOLL	Change timing of JTAG during runtime	32
SYStem.Option.SLOWRESET	Expand reset time for additional reset module	32
SYStem.Option.VBA	Set up VBA value for analysis	33
SYStem.Option.WaitReset	Halt the core after reset	34
SYStem.Option.WATCHDOG	Enable WATCHDOG	35
SYStem.JtagClock	Define JTAG clock	35
CPU specific MMU Commands		39
MMU.DUMP	Page wise display of MMU translation table	39
MMU.List	Compact display of MMU translation table	39
MMU.SCAN	Load MMU table from CPU	40
BenchMarkCounter		41
TrOnchip		42
TrOnchip	Control of on-chip resources	45
TrOnchip.CONVert	Automatically convert range to single address	46
TrOnchip.REGister	Shows custom on-chip trigger registers	46
TrOnchip.RESet	Set on-chip trigger to default state	46
TrOnchip.VarCONVert	Automatically convert range to single address	46
TrOnchip.state	Opens configure panel	47
On-chip Trace		48
Onchip.Mode	Select mode to control trace buffer and contents	48
Onchip.VTBA	Set the destination address of the onchip trace	49
General Restrictions		50
Floating Point Formats		51
Integer Access Keywords		51
File I/O Support		52
Metrowerks MSLIO Support		52
JTAG Connection		53
Mechanical Description of the 20-pin Debug Cable		53
Electrical Description of the 20-pin Debug Cable		54
JTAG Connector 14-pin		55
Memory Classes		57
StarCore Application Note		
StarCore Application Note for MXC Chips	(starcore_mxc_app.pdf)	1
NEXUS Preprocessors		3

Basic NEXUS Handling	4
Settings of the SYStem Window	4
Trigger Settings	9
Further NEXUS Trace Analysis	11
Display of the T-Bit in the Trace.List Window	11
OS Kernel related Trace Analysis	12
Benchmark Counter Analysis using DPU Counters	14

STM8

STM8 Debugger	(debugger_stm8.pdf)	1
Warning		4
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Script		5
Configuration		6
System Overview		6
Quick Start		7
Troubleshooting		9
FAQ		10
STM8 specific SYStem Settings		11
SYStem.CPU	Select the used CPU	11
SYStem.MemAccess	Select run-time memory access method	11
SYStem.Mode	Establish the communication with the target	12
SYStem.LOCK	Lock and tristate the debug port	12
SYStem.Option.IMASKASM	Disable interrupts while single stepping	13
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	13
CPU specific TrOnChip Commands		14
Breakpoints		15
Software breakpoints		15
On-chip breakpoints for instructions		15
On-chip breakpoints for data		15
Memory Classes		16
Target Adaption		17
Connector Type and Pinout		17

STRED

STRED Debugger and Trace	(debugger_stred.pdf)	1

Introduction	5
Brief Overview of Documents for New Users	5
Demo and Start-up Scripts	6
Warning	7
Quick Start	8
Troubleshooting	9
SYStem.Up Errors	9
FAQ	9
Configuration	10
System Overview	10
Debugging	12
Breakpoints	12
Runtime Access	14
Access Classes	15
Debug Code	16
Tracing	17
CPU specific SYStem Commands	19
SYStem.CONFIG	19
Configure debugger according to target topology	
SYStem.CONFIG.state	41
Display target configuration	
SYStem.CPU	42
Select the used CPU	
SYStem.JtagClock	43
Define JTAG frequency	
SYStem.LOCK	44
Lock and tristate the debug port	
SYStem.MemAccess	45
Select run-time memory access method	
SYStem.Mode	46
Establish the communication with the target	
SYStem.Option.AHBHPROT	47
Select AHB-AP HPROT bits	
SYStem.Option.AXIACEEnable	47
ACE enable flag of the AXI-AP	
SYStem.Option.AXICACHEFLAGS	47
Configure AXI-AP cache bits	
SYStem.Option.AXIHPROT	48
Select AXI-AP HPROT bits	
SYStem.Option.DAPDBGPWRUPREQ	49
Force debug power in DAP	
SYStem.Option.DAPNOIRCHECK	49
No DAP instruction register check	
SYStem.Option.DAPREMAP	50
Rearrange DAP memory map	
SYStem.Option.DAPSYSPWRUPREQ	50
Force system power in DAP	
SYStem.Option.DbgBase	51
Set base address of debug code	
SYStem.Option.DbgOvwr	51
Allow debug code overwrite	
SYStem.Option.DbgTrap	51
Allow trap handler address overwrite	
SYStem.Option.DEBUGPORTOptions	52
Options for debug port handling	
SYStem.Option.EnReset	53
Allow the debugger to drive nRESET (nSRST)	
SYStem.Option.EnTRST	53
Control TAP reset	
SYStem.Option.IMASKASM	53
Disable interrupts while single stepping	
SYStem.Option.IMASKHLL	54
Disable interrupts while HLL single stepping	

CPU specific Benchmarking Commands	55
BMC.<counter>.MODE	Configure counter mode 55
CPU specific TrOnchip Commands	56
TrOnchip.state	Display on-chip trigger window 56
TrOnchip.RESet	Set on-chip trigger to default state 56
RTM - Trace Source Configuration	57
RTM	Trace source RTM 57
RTM.CLEAR	Reset RTM registers to default values 58
RTM.CoreENable	Select specific cores for RTM trace 58
RTM.DataTrace	Configure data-trace 59
RTM.OFF	Switch RTM off 59
RTM.ON	Switch RTM on 60
RTM.Register	Display RTM registers 60
RTM.RESet	Reset RTM settings 61
RTM.state	Display RTM settings 61
RTM.Trace	Disable RTM configuration by the debugger 61
RTM.TraceID	Set RTM trace ID range 62
Target Adaption	63
Target Adaption for ARM	63

SuperH

SH2, SH3 and SH4 Debugger	(debugger_sh4.pdf) 1
History	5
Introduction	5
Brief Overview of Documents for New Users	6
Demo and Start-up Scripts	6
Warning	7
Application Note	8
Location of Debug Connector	8
Reset Line	8
Enable JTAG Mode SH2	9
Enable JTAG Mode SH3	9
SH7710/12 Solution Engine	9
Enable AUD Trace lines of SH7760	9
Memory Mapping of SH7615/ SH7616 BusControlRegisters	9
Enable 8-bit AUD Trace Interface of SH4-202	10
Quick Start JTAG	11
Troubleshooting	13
SYStem.Up Errors	13

Trace Errors	14
FAQ	14
Configuration	15
System Overview	15
CPU specific SYStem Settings	16
SYStem.CONFIG.state	Display target configuration 16
SYStem.CONFIG	Configure debugger according to target topology 17
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 21
SYStem.CPU	CPU type selection 22
SYStem.JtagClock	JTAG clock selection 22
SYStem.LOCK	JTAG lock 23
SYStem.MemAccess	Select run-time memory access method 24
SYStem.Mode	System mode selection 24
SYStem.Option.EnReset	Allow the debugger to drive nRESET 25
SYStem.Option.HOOK	Compare PC to hook address 25
SYStem.Option.IMASKASM	Interrupt disable 26
SYStem.Option.IMASKHLL	Interrupt disable 26
SYStem.Option.JtagWait	JTAG wait enable 26
SYStem.Option.KEYCODE	Keycode SH7144/45 26
SYStem.Option.MMUSPACES	Separate address spaces by space IDs 27
SYStem.Option.NoRunCheck	No check of the running state 28
SYStem.Option.SLOWRESET	Slow reset enable 28
SYStem.Option.SOFTLONG	Use LONG access for softbreak patching 28
SYStem.Option.SOFTSLOT	Prevent softbreak in slot-instruction 29
SYStem.Option.STEPSOFT	Use software breakpoints for ASM stepping 29
SYStem.Option.LittleEnd	Selection of little endian mode 29
SYStem.RESetOut	Reset target without reset of debug port 29
SYStem.Option.VBR	Vector base address (SH3/4 only) 30
Multicore Debugging	30
Breakpoints	31
Software Breakpoints	31
On-chip Breakpoints	31
On-chip Breakpoints SH7047, SH7144, SH7145	32
On-chip Breakpoints SH72513	32
Breakpoint in ROM	33
Example for Breakpoints	33
CPU specific BenchMarkCounter Commands	34
BMC.<counter>.ATOB	Advise counter to count within AB-range 34
CPU specific TrOnchip Commands	35
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 35
TrOnchip.IOB	I/O breakpoints (SH4, ST40) 35

TrOnchip.LDTLB	LDTLB breakpoints	35
TrOnchip.A.IBUS	I-bus breakpoints (SH2A)	36
TrOnchip.RESet	Set on-chip trigger to default state	36
TrOnchip.RPE	Reset sequential trigger on reset point	36
TrOnchip.SEQ	Sequential breakpoints (SH4, ST40)	37
TrOnchip.SIZE	Trigger on byte, word, long memory accesses	37
TrOnchip.state	Display on-chip trigger window	37
CPU specific MMU Commands		38
MMU.DUMP	Page wise display of MMU translation table	38
MMU.List	Compact display of MMU translation table	40
MMU.SCAN	Load MMU table from CPU	42
Memory Classes and Cache Handling		44
Memory Classes (SH2)		44
Memory Classes (SH3, SH4, ST40)		44
Cache Handling(SH3, SH4, ST40)		45
SYStem Commands		46
SYStem.Option.ICFLUSH	Cache invalidation option	46
SYStem.Option.DCFREEZE	Freeze data cache contents	46
SYStem.Option.DCCOPYBACK	Cache copy back	46
SYStem.Option.ICREAD	Cache read option	46
SYStem.Option.DCREAD	Cache read option	47
Trace		48
FIFO Trace (SH2A, SH3, SH4, ST40)		48
SYStem.Option.FIFO	FIFO trace configuration	48
LOGGER Trace (SH4, ST40, SH7705)		49
AUD-Trace (SH2A, SH4, ST40)		50
SYStem.Option.AUDBT	AUD branch trace enable	51
SYStem.Option.AUDDT	AUD data trace enable	51
SYStem.Option.AUDRTT	AUD real time trace enable	51
SYStem.Option.AUDClock	AUD clock select	51
SYStem.Option.AUD8	AUD 8-bit enable	52
AUD-Trace (SH3)		53
SYStem.Option.AUDRTT	AUD real time trace enable	53
SYStem.Option.AUDClock	AUD clock select	53
On-chip Trace SH2A		54
Onchip.Mode.MBusTrace	Mbus trace enable	54
Onchip.Mode.IBusTrace	Ibus trace enable	55
Onchip.Mode.ProgramTrace	Program flow trace enable	55
Onchip.Mode.DataReadTrace	Data read trace enable	55
Onchip.Mode.DataWriteTrace	Data write trace enable	56
On-chip Performance Analysis (SH4, ST40)		57
TrOnchip.PMCTRx	Performance counter configuration	57

Runtime Measurement	59
JTAG Connector	60

TI DSPs

C2000 Debugger	(debugger_c2000.pdf)	1
History		4
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
Converter from GEL to PRACTICE		6
Warning		7
DSP specific Implementations		8
Trigger		8
Breakpoints		8
Memory Classes		9
DSP specific SYStem Commands		10
SYStem.CONFIG.state	Display target configuration	10
SYStem.CONFIG	Configure debugger according to target topology	11
SYStem.CONFIG.ERAD	Embedded real-time analysis and diagnostic module	45
SYStem.CPU	Select the used CPU	45
SYStem.JtagClock	Define JTAG frequency	46
SYStem.LOCK	Tristate the JTAG port	47
SYStem.MemAccess	Select run-time memory access method	48
SYStem.Mode	Establish the communication with the target	48
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	49
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	49
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits	50
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	50
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	51
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	51
SYStem.Option.DAPREMAP	Rearrange DAP memory map	52
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	52
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	53
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	54
SYStem.Option.EnTRST	Control TAP reset	54
SYStem.Option.ExecutionMode	Sets the CPU execution mode	54
SYStem.Option.IMASKASM	Disable interrupts while single stepping	55
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	55
SYStem.Option.KEYCODE	Define key code to unsecure processor	55
SYStem.Option.TargetServer	Use target server from Texas Instruments	56

SYStem.RESetOut	Reset target without reset of debug port	56
TrOnchip Commands		57
TrOnchip.state	Display on-chip trigger window	57
TrOnchip.RESet	Set on-chip trigger to default state	57
ERAD Commands		58
ERAD	Embedded real-time analysis and diagnostic module	58
ERAD.OFF	Turn ERAD features off	58
ERAD.ON	Turn ERAD features on	58
JTAG Connection		59
Mechanical Description of the 20-pin Debug Cable		59
Electrical Description of the 20-pin Debug Cable		60
Mechanical Description of the TI Connector		61
FAQ		61
Operation Voltage		62
C5000 Debugger	(debugger_c5500.pdf)	1
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		6
Converter from GEL to PRACTICE		7
Warning		7
DSP specific Implementations		8
Trigger		8
Breakpoints		8
Memory Classes		9
DSP specific SYStem Commands		10
SYStem.Option.IMASKASM	Disable interrupts while single stepping	10
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	10
SYStem.CPU	Select the used CPU	10
SYStem.JtagClock	Define JTAG frequency	11
SYStem.MemAccess	Select run-time memory access method	12
SYStem.Mode	Establish the communication with the target	13
SYStem.CONFIG.state	Display target configuration	14
SYStem.CONFIG	Configure debugger according to target topology	16
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	49
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	49
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits	49
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	50
SYStem.Option.ByteMode	Define byte mode	50
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	51
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	51

SYStem.Option.DAPREMAP	Rearrange DAP memory map	52
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	52
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	53
SYStem.Option.DUALPORT	Implicitly use run-time memory access	54
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	54
SYStem.LOCK	Tristate the JTAG port	54
SYStem.Option.EnTRST	Control TAP reset	55
SYStem.Option.INTDIS	Disable all interrupts	55
SYStem.Option.MUHP	High-priority memory access	55
SYStem.Option.OVERLAY	Enable overlay support	56
SYStem.Option.PWRDWN	Allow power-down mode	56
SYStem.Option.TargetServer	Use target server from TI	57
SYStem.Option.TURBO	Use DMA for write accesses	57
SYStem.RESetOut	Reset the DSP	57
SYStem.Option.CToolsDecoder	Use TI's trace decoder software	58
SYStem.Option.CtoolsNoSync	CToolsNoSync	58
CPU specific BenchMarkCounter Commands		59
BMC.<counter>.ATOB	Advise counter to count within AB-range	59
BMC.<counter>.EVENT	Assign event to counter	60
TrOnchip Commands		61
TrOnchip.state	Display on-chip trigger window	61
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	61
C55X specific TrOnchip Commands		62
TrOnchip.ATOB	Activate on-chip breakpoints in AB-range	62
TrOnchip.BMCTR	Configure the benchmark counter	62
TrOnchip.CLOCK	Set the clock for the benchmark counter	66
TrOnchip.CoefficientAccess	AET trigger optimization	66
TrOnchip.DualAccess	AET trigger optimization	66
TrOnchip.PROfile	Display the benchmark data	66
TrOnchip.RESet	Set on-chip trigger to default state	67
Tracing		68
Controlling the Trace Capture		68
Trace Breakpoints		68
JTAG Connection		69
Mechanical Description of the 20-pin Debug Cable		69
Electrical Description of the 20-pin Debug Cable		70
Mechanical Description of the 14-pin Debug Cable		71
Electrical Description of the 14-pin Debug Cable		71
Mechanical Description of the TI Connector		72
FAQ		72
Operation Voltage		73

C6000 Debugger and Trace	(debugger_c6000.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
Converter from GEL to PRACTICE		6
Warning		7
DSP specific Implementations		8
Trigger		8
Breakpoints		8
Memory Classes		10
DSP specific SYStem Commands		11
SYStem.CONFIG.state	Display target configuration	11
SYStem.CONFIG	Configure debugger according to target topology	12
SYStem.CPU	Select the used CPU	46
SYStem.JtagClock	Define the frequency of the debug port	47
SYStem.LOCK	Tristate the JTAG port	48
SYStem.MemAccess	Select run-time memory access method	49
SYStem.Mode	Establish the communication with the target	50
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	51
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	52
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits	52
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	52
SYStem.Option.BigEndian	Enable big endian mode	53
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	53
SYStem.Option.DAPREMAP	Rearrange DAP memory map	53
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	54
SYStem.Option.IMASKASM	Disable interrupts while single stepping	55
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	56
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	56
SYStem.Option.DUALPORT	Implicitly use run-time memory access	57
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	57
SYStem.Option.EnTRST	Control TAP reset	58
SYStem.Option.HighPriority	Set data access priority	58
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	58
SYStem.Option.INTDIS	Disable all interrupts	59
SYStem.Option.PWRCHECK	Check power and clock	59
SYStem.Option.PWRDWN	Allow power-down mode	59
SYStem.RESetOut	Reset target without reset of debug port	59
C64x+ specific SYStem Commands		60
SYStem.Option.TargetServer	Use target server from Texas Instruments	60
SYStem.Option.TURBO	Use DMA for write accesses	60

CPU specific BenchMarkCounter Commands	61
BMC.<counter>.ATOB	Advise counter to count within AB-range 61
TrOnchip Commands	62
TrOnchip.state	Display on-chip trigger window 62
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 62
TrOnchip.RESet	Set on-chip trigger to default state 62
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource 63
Tracing	64
Controlling the Trace Capture	64
Trace Breakpoints	64
JTAG Connection	65
Mechanical Description of the 20-pin Debug Cable	65
Electrical Description of the 20-pin Debug Cable	66
Mechanical Description of the TI Connector	67
Electrical Description of the TI Connector	67
FAQ	68
Operation Voltage	69
C7000 Debugger and Trace	(debugger_c7000.pdf) 1
History	5
Introduction	6
Brief Overview of Documents for New Users	6
Demo and Start-up Scripts	6
Converter from GEL to PRACTICE	7
Warning	8
DSP specific Implementations	9
Trigger	9
Breakpoints	9
Access Classes	11
DSP specific SYStem Commands	13
SYStem.CONFIG.state	Display target configuration 13
SYStem.CONFIG	Configure debugger according to target topology 14
SYStem.CPU	Select the used CPU 48
SYStem.ENTERPostMortem	Place core into post-mortem state 48
SYStem.JtagClock	Define JTAG frequency 49
SYStem.LOCK	Tristate the JTAG port 50
SYStem.MemAccess	Select run-time memory access method 51
SYStem.MemFORCEREADY	Unblock memory access in post-mortem state 51
SYStem.Mode	Establish the communication with the target 52
SYStem.Option.Address32	Define address format display 53

SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	54
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	54
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits	54
SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	55
SYStem.Option.BigEndian	Enable big endian mode	55
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	55
SYStem.Option.DAPREMAP	Rearrange DAP memory map	56
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	56
SYStem.Option.DUALPORT	Implicitly use run-time memory access	57
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	57
SYStem.Option.EnTRST	Control TAP reset	58
SYStem.Option.ICFLUSH	Invalidate instruction cache before go and step	58
SYStem.Option.IMASKASM	Disable interrupts while single stepping	58
SYStem.Option.INTDIS	Disable all interrupts	59
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	60
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	60
SYStem.Option.ExecutionMode	Sets the CPU execution mode	61
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	61
SYStem.Option.MACHINESPACES	Address extension for guest OSes	62
SYStem.Option.PWRDWN	Allow power-down mode	62
SYStem.RESetOut	Reset target without reset of debug port	63
CPU specific BenchMarkCounter Commands		64
BMC.<counter>.ATOB	Advise counter to count within AB-range	64
CPU specific SETUP Command		65
SETUP.DIS	Disassembler configuration	65
TrOnchip Commands		66
TrOnchip.state	Display on-chip trigger window	66
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	66
TrOnchip.RESet	Set on-chip trigger to default state	66
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	67
Tracing		68
Controlling the Trace Capture		68
Trace Breakpoints		68
Command reference: TRC		69
TRC	Trace control (TRC)	69
TRC.CLEAR	Clear trace settings	69
TRC.CLOCK	Set core clock frequency for timing measurements	70
TRC.DataTrace	Define broadcast of load/store address tracing	71
TRC.OFF	Switch TRC off	71
TRC.ON	Switch TRC on	72
TRC.PCTrace	Enable program counter trace	72
TRC.RESet	Reset TRC settings	72

TRC.STALL	Stall processor to prevent FIFO overflow	72
TRC.StreamBuffer	Enable stream buffer trace	73
TRC.SyncPeriod	Set synchronization frequency	73
TRC.TimeMode	Set timestamp configuration	73
TRC.TimeStampCLOCK	Specify frequency of the global timestamp	74
TRC.Trace	Enable TRC trace export	74
TRC.TraceID	Change the default ID for a TRC trace source	74
TRC.TracePriority	Define priority of TRC messages	74
TRC.state	Display TRC setup	75
Target Adaption		76
Probe Cables		76
Interface Standards JTAG, Serial Wire Debug, cJTAG		76
Connector Type and Pinout		76
FAQ		77

TriCore

TriCore Debugger and Trace	(debugger_tricore.pdf)	1
History		8
Safety Precautions		9
Introduction		10
Brief Overview of Documents for New Users		10
Available Tools		11
Software Installation		13
Configuration		14
Related Documents		15
Demo and Start-up Scripts		15
OCDS Levels		16
Debugging		17
Single-Core Debugging (AUDO)		18
Multicore Debugging (AURIX)		20
Access Classes		27
Breakpoints		28
Single Stepping		30
Flash		31
Onchip Triggers (TrOnchip Window)		33
BenchMarkCounter		34
Watchpins		39
Accessing Cached Memory Areas and Cache Inspection		43
Parallel Usage of a 3rd-Party Tool		48
Debugging an Application with the Memory Protection Unit Enabled		50

Debugging through Resets and Power Cycles	52
Code Overlays	57
Data Overlays	57
Cerberus Access Protection	57
Target Code Execution	58
Internal Break Bus (JTAG)	58
Troubleshooting	59
FAQ	60
Tracing	61
On-chip Trace (OCDS-L3)	61
CPU specific BMC Commands	71
BMC.SELect	Select counter for statistic analysis 71
BMC.<counter>.ATOB	Control A-to-B mode 71
BMC.<counter>.TRIGMODE	BMC trigger mode 71
BMC.<counter>.TRIGVAL	BMC trigger value 72
CPU specific SYStem.CONFIG Commands	73
SYStem.CONFIG.state	Display target configuration 73
SYStem.CONFIG	Configure debugger according to target topology 74
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 78
SYStem.CONFIG.BreakPIN	Define mapping of break pins 79
SYStem.CONFIG.CAN	Configure CAN interface 80
SYStem.CONFIG.CAN.BaseCLOCK	Base clock for CAN interface 80
SYStem.CONFIG.CAN.NominalBRP	Set CAN nominal baud rate prescaler 81
SYStem.CONFIG.CAN.NominalTSEG1	Set CAN nominal Phase_seg1 81
SYStem.CONFIG.CAN.NominalTSEG2	Set CAN nominal Phase_seg2 81
SYStem.CONFIG.CAN.NominalSJW	Set CAN nominal SJW parameter 81
SYStem.CONFIG.CAN.DataBRP	Set CAN data baud rate prescaler 82
SYStem.CONFIG.CAN.DataTSEG1	Set CAN data Phase_seg1 82
SYStem.CONFIG.CAN.DataTSEG2	Set CAN data Phase_seg2 82
SYStem.CONFIG.CAN.DataSJW	Set CAN data SJW 83
SYStem.CONFIG.DAP	Configure DAP interface 84
SYStem.CONFIG.DAP.BreakPIN	Define mapping of break pins 84
SYStem.CONFIG.DAP.CRC6	Enable CRC6 mode 84
SYStem.CONFIG.DAP.DAPENable	Enable DAP mode on PORST 84
SYStem.CONFIG.DAP.SISP	Configure SISP setting 85
SYStem.CONFIG.DAP.USERn	Configure and set USER pins 85
SYStem.CONFIG.DEBUGPORT	Select target interface 87
SYStem.CONFIG.DEBUGPORTTYPE	Set debug cable interface mode 87
SYStem.CONFIG.DXCM	Configure DXCM 89
SYStem.CONFIG.DXCM.TXID	Control frame message ID 89
SYStem.CONFIG.DXCM.TXIDE	Control frame format 89
SYStem.CONFIG.DXCM.TXFDF	Control frame format 90

SYStem.CONFIG.DXCM.TXBRS	Control the use of baud rate switching	90
SYStem.CONFIG.DXCM.RXID	Set ID for frames from target	90
SYStem.CONFIG.DXCM.RXIDE	Expect extended frames from target	90
SYStem.CONFIG.DXCPL	Configure DXCPL	91
SYStem.CONFIG.DXCPL.Timing	Configure SPD timing for DXCPL	91
SYStem.CONFIG.EXTWDTDIS	Disable external watchdog	91
SYStem.CONFIG.PortSHaRing	Control sharing of debug port with other tool	92
SYStem.CPU	Select CPU	93
SYStem.JtagClock	Set the JTAG frequency	94
SYStem.LOCK	Tristate the JTAG port	95
SYStem.MemAccess	Select run-time memory access method	96
SYStem.Mode	Establish the communication with the CPU	97
CPU and Architecture specific SYStem.Option Commands		99
SYStem.Option.BREAKFIX	Enable workaround for asynchronous breaking	99
SYStem.Option.CBSACCEN<x>	Cerberus access protection	100
SYStem.Option.DCFREEZE	Do not modify cache structure	101
SYStem.Option.DCREAD	Control cache behavior of reads	101
SYStem.Option.DSYNC	Force data synchronization	102
SYStem.Option.DOWNMODE	Behavior of SYStem.Mode Down	103
SYStem.Option.DUALPORT	Implicitly use run-time memory access	103
SYStem.Option.DataTrace	Enable data tracing	104
SYStem.Option.EndInitProtectionOverride	Override ENDINIT protection	104
SYStem.Option.HeartBeat	Bug fix to avoid FPI bus conflict	105
SYStem.Option.HoldReset	Reset duration	105
SYStem.Option.HSMRESTART	Restart HSM on connect	106
SYStem.Option.ICFLUSH	Flush instruction cache at 'Go' or 'Step'	106
SYStem.Option.IMASKASM	Disable interrupts while single stepping	106
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	107
SYStem.Option.JTAGENSEQ	Use JTAG initialization sequence	107
SYStem.Option.KEYCODE	Set debug interface password	108
SYStem.Option.KEYCODEWarnNotAccepted	Set warning level	108
SYStem.Option.LBIST	LBIST gap handling	108
SYStem.Option.MACHINESPACES	Address extension for guest OSes	109
SYStem.Option.MAPCACHE	Map cache automatically	110
SYStem.Option.OCDSELOW	Set OCDS line to low	111
SYStem.Option.OVC	Enable OVERLAY memory access	111
SYStem.Option.OVERLAY	Enable overlay support	112
SYStem.Option.PERSTOP	Enable global peripheral suspend	113
SYStem.Option.PMILBFIX	Enable PMI line buffer invalidation workaround	114
SYStem.Option.PostResetDELAY	Delay after RESET is released	115
SYStem.Option.ReadOnly	Block all write accesses	115
SYStem.Option.RESetBehavior	Set behavior when a reset occurs	116
SYStem.Option.ResetDetection	Set how hard resets are detected	116

SYStem.Option.ResetMode	Select reset method	117
SYStem.Option.RESetTMS	State of TMS line at reset	117
SYStem.Option.RUNRESTOREDELAY	Delay of restore after reset	117
SYStem.Option.SLOWRESET	Long timeout for resets	118
SYStem.Option.SOFTLONG	Set 32 bit software breakpoints	118
SYStem.Option.SSWWAIT	Emulate SSWWAIT	118
SYStem.Option.STEPONCHIP	Step with onchip breakpoints	119
SYStem.Option.STEPSOFT	Step with software breakpoints	119
SYStem.Option.TB1766FIX	Bug fix for some TC1766 TriBoards	120
SYStem.Option.UNLOCKTIME	Timeout for debug port unlock	120
SYStem.Option.WDTFIX	Disables the watchdog on SYStem.Up	120
SYStem.Option.WDTSUS	Link the watchdog timer to the suspend bus	121
SYStem.RESetOut	In-target reset	121
SYStem.state	Open SYStem.state window	121
CPU specific TrOnchip Commands		122
TrOnchip.BreakBusN.BreakIN	Configure break pin of 'BreakBus N'	122
TrOnchip.BreakBusN.BreakOUT	Configure break pin of 'BreakBus N'	122
TrOnchip.BreakIN.<target>	Connect break <target> to BreakBus	123
TrOnchip.BreakOUT.<source>	Connect break <source> to BreakBus	123
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	125
TrOnchip.CountX	Event X counter value	125
TrOnchip.CountY	Event Y counter value	125
TrOnchip.EXTernal	Configure TriCore break on BreakBus event	126
TrOnchip.PERSTOPOUT	Route suspend signal to pin	126
TrOnchip.RESet	Reset settings for the on-chip trigger unit	126
TrOnchip.SoftWare	Configure 'TriCore' break on debug instruction	127
TrOnchip.SusSWitch	Enable or disable suspend switch	127
TrOnchip.SusSWitch.FORCE	Force generation of suspend signal	127
TrOnchip.SusSWitch.Mode	Set suspend switch mode	128
TrOnchip.SusTarget	Connect special targets to the suspend bus	128
TrOnchip.SYNCHRONOUS	Switches mode for data breakpoints	128
TrOnchip.TDelay	Trace trigger delay	129
TrOnchip.TExtMode	Mode for external trigger input	129
TrOnchip.TExtPol	Polarity of external trigger input	129
TrOnchip.TMode	Trace mode	129
TrOnchip.TR0	Specify trigger event 0	130
TrOnchip.TR1	Specify trigger event 1	131
TrOnchip.state	Show on-chip trigger window	131
TrOnchip.WatchPin	Route core trigger to pin	132
TrOnchip.X	Select trigger source X	132
TrOnchip.Y	Select trigger source Y	133
Technical Data		134
Trace Connector		134

Technical Data for Debugger	134
Appendix	135
Parallel Off-chip Trace - OCDS-L2 Flow Trace (Analyzer)	135
MCDS User's Guide(mcds_user.pdf)	1
History	6
Introduction	7
Intended Audience	7
How to Read This Document	7
Related Documents	8
Background Information	9
Trace Source	9
Trace Sink	10
Trace Filter and Trigger	10
The Emulation Device Concept	11
TRACE32 Support for Emulation Devices	13
Feature Overview	13
Target Interface	13
MCDS Licensing	14
MCDS Basic Features	16
MCDS Concept	16
MCDS Configuration	17
Trace Control	22
Basic Trace Usage	24
Trace Decoding	32
MCDS Unlocking	39
MCDS Special Features	40
Benchmark Counters	40
Trace Through Resets and Power Cycles	45
Special Trace Sources via OTGM	46
miniMCDS	60
Complex Trigger Language CTL	62
Clock System	63
EEC Clock System	63
MCDS Clock System	68
MCDS Clock Configuration	71
Emulation Memory	74
Background Information	74
EMEM Partitioning	75
AGBT High-speed Serial Trace	82

Background Information	82
Xilinx Aurora	83
Requirements	83
AGBT Configuration	88
Trace Streaming	89
Limitations and Restrictions	89
Advanced Emulation Device Access	91
EEC Access	91
Guarded MCDS Programming	93
Example Scripts	96
Known Issues and Application Hints	97
Missing Instructions	97
Invalid Program Trace at the Beginning of the Trace Recording	97
No Trace Content Displayed	97
FIFOFULL error	98
Concurrent Usage of Different Trace Methods	98
PCP Channel ID	99
Glossary	100
Infineon Glossary	100
Lauterbach Glossary	101
TriCore Application Notes	
Application Note Debug Cable TriCore(app_tricore_ocds.pdf)	1
History	5
Introduction	6
Debug Protocols	7
JTAG	7
DAP	8
Connector Standards and Signals	11
Description of Signals	11
OCDS-L1 Connector	13
Automotive Debug Connector	15
CAN D-Sub Connector	16
Custom Connectors	16
Trace Connectors	18
Debug Cables and CombiProbe Whiskers	22
OCDS Debug Cables	25
AUTO26 Debug Cables	30
AUTO26 Whisker for CombiProbe	35
Debug Interface Configuration	37
Connecting using DAP over Dedicated Pins	37

Connecting using JTAG	38
Connecting using DXCPL/DXCM with DXCPL Box	39
Sharing the Debug Port between TRACE32 and 3rd-Party Tool	42
DAP User Pins	42
Break Pins	43
Controlling an External Watchdog	44
Adapters, Converters and Extensions	45
Adapter 16-pin 100 mil to 50 mil	45
Converter 16-pin JTAG to DAP for TriCore/XC2000/XC800	46
Converter DXCPL Box for TriCore	47
Converter 16-pin JTAG to BOSCH MEDC17 for TriCore	48
Converter AUTO26/ OnCE14-PPC/ JTAG16-TC to ECU14	48
Converter AUTO26 to ECU14	49
Converter JTAG16-TriCore to AUTO26	50
Converter AUTO26 to JTAG16-TriCore	50
Converter AUTO20 to HSTCU	51
Converter Samtec 60 to AMP 40	52
Converter 16-pin OCDS-L1 to Samtec 60 for TriCore	52
Converter 16-pin OCDS-L1/ 40-pin HSSTP to ERF8 for TriCore	53
Converter OCDS-L1/ AUTO26/ PowerTrace Serial to ERF8 for TriCore	54
Converter AUTO26/ 40-pin HSSTP to HSTCU	54
Converter AUTO26/ PowerTrace Serial to HSTCU	55
No-Debug Extension for SAMTEC 22-pin ERF8-ERM8 series	56
Flex Extension for SAMTEC 60-pin QTH-QSH series	57
Flex Extension for SAMTEC 60-pin QTH-QSH series	57
Cable 26-pin for AUTO26 Debug Cables	58
Cable 20-pin for AUTO26 Debug Cables	58
Cable 10-pin for AUTO26 Debug Cables	59
Recommended Connectors	60
Standard 2x8 Connector	60
Half-size 2x8 Connector	60
Half-size 2x5 Connector	61
Half-size 2x5 Connector with Keying Pin 7	61
Half-size 2x10 Connector with Keying Pin 7	62
Half-size 2x13 Connector with Keying Pin 7	63
TFM 2x5 Connector	63
AMP 40 Connector	64
ERF8 22-pin Power.org Connector	64
Samtec 60 Connector	65
USB Type-C Receptacle	65
Technical Data for Trace	66
Mechanical Dimensions	66

Application Note FLASH Programming TriCore	(app_tricore_flash.pdf)	1
Introduction		4
FLASH Programming Commands		5
Organization of TriCore FLASH Scripts		6
FLASH Declaration Scripts		6
FLASH Feature Scripts		7
FLASH Demo Scripts		7
TC2xx Devices		11
Erased FLASH		11
TRACE32 Methods for Safe FLASH Programming		13
OTP and WOP Sectors		21
TC3xx Devices		22
Erased FLASH		22
TRACE32 Mechanisms for Safe FLASH Programming		24
OTP and WOP Sectors		33
DFLASH Single-Ended and Complement Sensing Mode		34
Support for SOTA		35
PCP Debugger Reference	(debugger_pcp.pdf)	1
Warning		6
Introduction		7
Brief Overview of Documents for New Users		7
Demo and Start-up Scripts		7
PCP Debugger Implementations		8
Quick Start		9
Quick Start for OCDS-L1 Debugger		9
Quick Start for Tracing with OCDS-L2 Trace (Analyzer)		12
Quick Start for Tracing with OCDS-L3 Trace (On-chip Trace)		14
OCDS-L1 Debugger		15
Troubleshooting		15
Memory Classes		16
Breakpoints		17
OCDS Trace		18
OCDS-L2 Flow Trace (Analyzer)		18
OCDS-L3 On-chip Trace		20
Simple Trace Control		21
Coupling of PCP and Host-core Debugger		22
Modify TRACE32 configuration files		22
Start PowerView instances		22
Synchronous Break		22

Synchronous Step or Go	24
FAQ	25
Commands	26
SYStem.CONFIG	Configure debugger according to target topology 26
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 30
SYStem.CONFIG.PortSHaRing	Control sharing of debug port with other tool 31
SYStem.CPU	Select CPU 31
SYStem.JtagClock	Set the JTAG frequency 33
SYStem.LOCK	Tristate the JTAG port 34
SYStem.MemAccess	Select run-time memory access method 35
SYStem.Mode	Establish the communication with the CPU 36
SYStem.Option	CPU specific commands 37
SYStem.Option.BreakSig	Generate break signal 37
SYStem.Option.CodeBASE	PCODE base address 37
SYStem.Option.CodeSIZE	PCP PRAM size 37
SYStem.Option.CPUREQ	CPU request address 38
SYStem.Option.DAC	Disable all channels on break 38
SYStem.Option.DIAG	Diagnosis function 38
SYStem.Option.DUALPORT	Run-time memory access for all windows 39
SYStem.Option.PramBASE	PRAM base address 39
SYStem.Option.PermanentBP	Enable breakpoints when single stepping 39
SYStem.Option.PramSIZE	PCP PRAM size 40
SYStem.Option.RegBASE	PCP configuration register base address 41
SYStem.Option.TB1766FIX	Bug fix for some TC1766 TriBoards 42
CPU specific TriggerOnchip Commands	43
Internal Break Bus (JTAG)	43
Trace Break Signals (OCDS-L2)	43
TrOnchip.BreakIN	Connect break target PCP to BreakBus 44
TrOnchip.BreakOUT	Connect break source PCP to BreakBus 44
TrOnchip.CONVert	Not relevant for the PCP architecture 44
TrOnchip.RESet	Reset settings for the on-chip trigger unit 44
TrOnchip.SusTarget	Connect PCP to the suspend bus 45
CPU specific BenchMarkCounter Commands	46
BMC.<counter>.ATOB	Advise counter to count within AB-range 46
JTAG Connector	47
Trace Connector	47
Debugging via Infineon DAS Server	(backend_das.pdf) 1
Introduction	3
Related Documents	3
Contacting Support	3

System Architecture	5
PowerView System Configurations	6
System Initialization via the TRACE32 PowerView GUI	9
System Initialization via the TRACE32 Command Line	11
Keep the Graphical User Interface Responsive	13
Timing Adaption	14
Command Reference	15
SYStem.InfineonDAS	Configure the InfineonDAS debug port 15
SYStem.InfineonDAS.CBSBUSNAME	Bus access transactor 15
SYStem.InfineonDAS.CBSINSTRNAME	Cerberus instruction transactor 15
SYStem.InfineonDAS.CONNECT	Connect to DAS server 16
SYStem.InfineonDAS.DISCONNECT	Disconnect from the server 16
SYStem.InfineonDAS.EXPLore	Explore server interactively 17
SYStem.InfineonDAS.InfineonDAPNAME	DAP transactor 18
SYStem.InfineonDAS.MODELNAME	Select port instance 18
SYStem.InfineonDAS.SERVERCONFIG	Configure server options 19

V850

V850 Debugger and Trace	(debugger_v850.pdf) 1
History	5
Introduction	6
Brief Overview of Documents for New Users	6
Demo and Start-up Scripts	6
Warning	8
Application Note	9
Location of Debug Connector	9
Reset Line	9
FLMD0 Line	10
Mask-Options of V850/Fx3, Cargate	11
Quick Start JTAG	12
Troubleshooting	15
SYStem.Up Errors	15
FAQ	15
Configuration	16
System Overview	16
CPU specific SYStem Settings	17
SYStem.CONFIG.state	Display target configuration 17

SYStem.CONFIG	Configure debugger according to target topology	18
SYStem.CONFIG.CORE	Assign core to TRACE32 instance	22
SYStem.CONFIG.EXTWDTDIS	Disable external watchdog	23
SYStem.CONFIG.DEBUGPORTTYPE	Select debug port type	24
SYStem.CONFIG.PortSHaRing	Control sharing of debug port with other tool	24
SYStem.CPU	CPU type selection	25
SYStem.JtagClock	JTAG clock selection	25
SYStem.LOCK	Lock and tristate the debug port	25
SYStem.MemAccess	Select run-time memory access method	26
SYStem.Mode	System mode selection	27
SYStem.Option.IMASKASM	Interrupt disable	27
SYStem.Option.IMASKHLL	Interrupt disable	28
SYStem.Option.PERSTOP	Disable CPU peripherals if stopped	28
SYStem.RESetOut	Reset target without reset of debug port	28
Exception Lines Enable		29
SYStem.Option.RESET	Reset line enable	29
SYStem.Option.STOP	Stop line enable	29
SYStem.Option.WAIT	Wait line enable	29
SYStem.Option.REQest	Request line enable	30
SYStem.Option.NMI0	NMI0 line enable	30
SYStem.Option.NMI1	NMI1 line enable	30
SYStem.Option.NMI2	NMI2 line enable	30
SYStem.Option.CPINT	CPINT line enable	31
Trace System Settings		32
SYStem.Option.BTM	Branch trace message	32
SYStem.Option.DTM	Data trace message	33
SYStem.Option.KEYCODE	Keycode	33
SYStem.Option.OPWIDTH	Trace interface width	34
SYStem.Option.STALL	Trace STALL mode	35
SYStem.Option.TCMODE	Trace clock mode	35
Breakpoints		36
Software Breakpoints		36
On-chip Breakpoints		36
Breakpoint in ROM		37
Example for Breakpoints		37
TrOnchip Commands		38
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	38
TrOnchip.RCU	ROM-Correction breakpoints	39
TrOnchip.RESet	Set on-chip trigger to default state	39
TrOnchip.Set.Alignment	Alignment error breakpoints	39
TrOnchip.Set.MissAlign	Alignment error breakpoints	40
TrOnchip.SIZE	Trigger on byte, word, long memory accesses	40

TrOnchip.state	Display on-chip trigger window	40
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource	40
CPU specific Functions		41
CPU.BASEFAMILY()	CPU family	41
CPU.SUBFAMILY()	CPU subfamily	41
Memory Classes		42
DataFlash: Memory Class		42
NBD Interface		43
Runtime Measurement		44
JTAG Connector		45
Connector 20 pin 100mil /NWire		45
Trace Connector		46
Connector MICTOR/N-Wire and Trace		46
Connector KEL/N-Wire and Trace		47
NBD Connector		48

x86

Intel® x86/x64 Debugger	(debugger_x86.pdf)	1
History		7
Brief Overview of Documents for New Users		8
Welcome Dialog		8
Help Menu		9
Further Documents		10
Warning		12
Quick Start		13
Troubleshooting		16
FAQ		16
x86 specific Implementations		17
Tool Identification		17
Onchip Breakpoints		17
Breakpoints after Reset/Power Cycle		18
Access Classes		19
Memory Model		30
Segmentation		31
Platform Controller Hub (PCH)		32
Slave Core Debugging		34
CPU specific JTAG.CONFIG Commands		37

JTAG.CONFIG	Electrical characteristics of MIPI-60 debug signals	37
JTAG.CONFIG.DRiVer	Set slew rate of JTAG signals	37
JTAG.CONFIG.PowerDownTriState	Automatically tristate outputs	38
JTAG.CONFIG.TckRun	Free-running TCK mode	38
JTAG.CONFIG.TDOEdge	Select TCK edge	38
JTAG.CONFIG.Voltage.HookKThreshold	Set hook threshold voltages	39
JTAG.CONFIG.Voltage.REFerence	Set reference voltage source	40
JTAG.CONFIG.Voltage.THreshold	Set JTAG threshold voltages	40
CPU specific SYStem.DETECT Commands		41
SYStem.DETECT.CLTapchain	Show SOC IDs of SOC slave cores	41
SYStem.DETECT.CORES	Detect core/thread number	41
SYStem.DETECT.HyperThreads	Detect hyper thread status	42
SYStem.DETECT.TARGET	Fully automatic board setup	43
SYStem.DETECT.TOPOlogy	Detect board topology	44
CPU specific SYStem Settings		45
SYStem.CONFIG.state	Display target configuration	45
SYStem.CONFIG	Configure debugger according to target topology	46
SYStem.CORESTATES	Core states overview	49
SYStem.CPU	Select the target CPU/SOC	51
SYStem.JtagClock	Define JTAG clock	51
SYStem.LOCK	Tristate the JTAG port	51
SYStem.MemAccess	Select run-time memory access method	52
SYStem.Mode	Establish the communication with the target	52
SYStem.Option.Address32	Use 32 bit address display only	53
SYStem.Option.BIGREALmode	Enable Big Real mode handling	54
SYStem.Option.BranchSTEP	Enables branch stepping	55
SYStem.Option.BreakDELAY	Set max. break delay	55
SYStem.Option.C0Hold	Hold CPU in C0 state	55
SYStem.Option.IGnoreDEbugReDirections	Ignore debug redirections	56
SYStem.Option.IGnoreSOC	Ignore SoC TAP chain structure	56
SYStem.Option.IGnoreSWBPreDirections	Ignore SW BP redirections	56
SYStem.Option.IMASKASM	Disable interrupts while single stepping	57
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	57
SYStem.Option.InstrSUBmitForcePHYsicalPRDY	Use physical PRDY	57
SYStem.Option.InstrSUBmitIGnorePHYsicalPRDY	Ignore physical PRDY	57
SYStem.Option.InstrSUBmitTimeout	Timeout for instruction submission	58
SYStem.Option.IntelSOC	Slave core is part of Intel® SoC	58
SYStem.Option.JTAGDirectCPU	JTAG directly to CPU TAPs	59
SYStem.Option.JTAGOnly	Use only JTAG signals	59
SYStem.Option.MACHINESPACES	Address extension for guest OSes	59
SYStem.Option.MEMoryMODEL	Define memory model	60
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	63
SYStem.Option.MultiCoreWhiskers	Server board whisker setup	64

SYStem.Option.NoDualcoreModule	Disable dualcore module support	64
SYStem.Option.NoHyperThread	Disable HyperThreading support	65
SYStem.Option.NoIPAdjust	Do not adjust IP at reset vector	65
SYStem.Option.NoReBoot	Disable watchdog causing reboot	65
SYStem.Option.OSWakeupTIME	Set the OS wake up time	66
SYStem.Option.PC10MODE	Wake up target from package C10	66
SYStem.Option.PreserveDRX	Preserve DRx resources	66
SYStem.Option.PreserveLBR	Preserve LBR resources	66
SYStem.Option.ProbeModeNOSaveRestore	No save/restore	67
SYStem.Option.ProbeModeONDEmand	On demand save/restore	68
SYStem.Option.PWRCycleTime	Set power cycle time	68
SYStem.Option.PWROFFTime	Set power off assertion time	68
SYStem.Option.PWRONTime	Set power on assertion time	69
SYStem.Option.PWRONWaitTime	Set power on time	69
SYStem.Option.ReArmBreakPoints	Rearm breakpoints on reset	69
SYStem.Option.REL	Relocation register	69
SYStem.Option.RESetDELAY	Set reset delay	70
SYStem.Option.RESetDetection	Select reset detection source	70
SYStem.Option.RESetMode	Select reset method	71
SYStem.Option.RESetTIME	Set reset assertion time	71
SYStem.Option.RESetWaitTIME	Set reset input wait time	71
SYStem.Option.S0Hold	Hold SoC in S0 state	72
SYStem.Option.SOFTLONG	Use 32-bit access to set SW breakpoint	72
SYStem.Option.STandBYAttach	In standby mode, only attach to target	72
SYStem.Option.STandBYAttachDELAY	Delay after standby	73
SYStem.Option.STepINToEXC	Step into interrupt or exception handler	73
SYStem.Option.TOPOlogy	Select server board topology	73
SYStem.Option.WatchDogWaitTIME	Set the reset watch dog time	74
SYStem.Option.WFSMemAccess	Allow WFS memory access	74
SYStem.Option.WHISKER	Select a whisker	74
SYStem.Option.ZoneSPACES	Enable symbol management for zones	75
SYStem.PCH	Select the target PCH	77
SYStem.POWER	Control target power	77
SYStem.STALLPhase	Set system into stall phase	78
SYStem.StuffInstruction	Submit instruction to CPU in probe mode	78
SYStem.StuffInstructionRead	Submit instruction and read	78
SYStem.TIMINGS	Display timings window	79
Command Groups for Special Registers		80
CPU specific MMU Commands		81
MMU.DUMP	Page wise display of MMU translation table	81
MMU.List	Compact display of MMU translation table	85
MMU.SCAN	Load MMU table from CPU	87
MMU.Set	Set MMU register	88

CPU specific TrOnchip Commands - Onchip Triggers	89
TrOnchip.PrintList	Print possible onchip triggers 89
TrOnchip.RESet	Reset settings to defaults 89
TrOnchip.Set	Break on event 89
TrOnchip.Set.BootStall	Enter bootstall 89
TrOnchip.Set.C6Exit	Break on C6 exit 91
TrOnchip.Set.ColdRESet	Break on cold reset 91
TrOnchip.Set.CpuBootStall	Enter CPU bootstall 91
TrOnchip.Set.ENCLU	Break on ENCLU event 92
TrOnchip.Set.GeneralDetect	Break on general detect 92
TrOnchip.Set.INIT	Break on init 92
TrOnchip.Set.MachineCheck	Break on machine check 92
TrOnchip.Set.RESet	Break on target reset 93
TrOnchip.Set.ShutDown	Break on shutdown 93
TrOnchip.Set.SMMENTry	Break on SMM entry 93
TrOnchip.Set.SMMEXit	Break on SMM exit 93
TrOnchip.Set.SMMINto	Step into SMM when single stepping 94
TrOnchip.Set.TraceHub	Enter/leave trace hub break 94
TrOnchip.Set.VMENTry	Break on VM entry 94
TrOnchip.Set.VMEXit	Break on VM exit 95
TrOnchip.state	Display onchip trigger window 97
CPU specific Events for the ON and GLOBALON Command	98
CPU specific BenchmarkCounter Commands	99
BMC.<counter>	Select BMC event to count 99
BMC.<counter>.COUNT	Select count mode for BMC 99
CPU specific Onchip Trace Commands	100
Onchip.Buffer	Configure onchip trace source 100
CPU specific Functions	102
SYStem.CoreStates.APIC()	102
SYStem.CoreStates.HYPER()	102
SYStem.CoreStates.MODE()	102
SYStem.CoreStates.PHYS()	103
SYStem.CoreStates.PRIOR()	103
SYStem.CoreStates.SMM()	103
SYStem.CoreStates.VMX()	104
SYStem.Option.MEMoryMODEL()	104
SYStem.Option.TOPOlogy()	104
SYStem.Option.TOPOlogy.SOCKETS()	104
SYStem.ReadPDRH()	105
SYStem.ReadPDRL()	105
TrOnchip.IsAvailable()	105
TrOnchip.IsSet()	106

VMX()	106
VMX.Guest()	106
SYStem Trace Settings	107
Connectors	108
JTAG Connector	108
MIPI34 Connector	109
MIPI60-C Connector	110
MIPI60-Cv2 Connector	112
MIPI60-Q Connector	114
Tools for Intel® x86/x64(tools_intel_x86.pdf)	1
Introduction	3
Legend	3
TRACE32 QuadProbe	4
QuadProbe and PowerDebug Module USB 3.0	5
QuadProbe and PowerDebug PRO	12
Extra	19
Extra (USB-2-CABLE)	19
Extras (TRIGGER-CONNECTOR)	20
Intel® Application Note for Server Setup(app_x86_server.pdf)	1
Introduction	3
Prerequisites	3
How This Manual is Organized	3
Related Documents	3
Contacting Support	4
Server Board Topologies	5
Probe Whiskers	8
Standard Setup	9
Manual Setup	10
Special Cases	11
Intel® Processor Trace (trace_intel_pt.pdf)	1
Configuration	4
Selective Tracing (optional)	4
Tracing to Memory	6
Memory Buffer Size	6
Example Script	7
CPU specific IPT Commands	8
IPT	Intel® Processor Trace (IPT) 8

IPT.CLEAR	Clear memory for new core configuration	8
IPT.CLOCK	CPU core clock	8
IPT.CompressReturn	Compression of near return addresses	8
IPT.CR3	Filtering by CR3	9
IPT.CycleAccurate	Enable cycle accurate tracing	9
IPT.CycleCountThreshold	Cycle count for cycle accurate tracing	9
IPT.DataTrace	Enable data tracing	10
IPT.DataTraceFUP	Trace IP when data trace packet is generated	10
IPT.EXPORTBASE	IPT output region	10
IPT.IgnoreGERR	Ignore FIFO full errors	11
IPT.LessPackets	Do not create IPT packets on certain circumstances	11
IPT.MiniTimeCounter	Enable 'MiniTimeCounter' packets	11
IPT.OFF	Switch IPT off	12
IPT.ON	Switch IPT on	12
IPT.PortRoute	Selection of trace HW	12
IPT.Register	Show IPT registers	12
IPT.PacketCount	Synchronization period	13
IPT.RESet	Reset IPT settings	13
IPT.state	Display IPT settings	14
IPT.SuperTimeSync	Enable 'SuperTimeSync' packets	14
IPT.SyncPeriod	Synchronization period	15
IPT.TImeMode	Timestamp mode	15
IPT.TimeStampCLOCK	Timestamp frequency	15
IPT.TraceCORE	Trace selected cores only	15
IPT.TraceID	Assign STP ID(s) to core(s)	16
IPT.TraceOS	Filtering by current privilege level 0	16
IPT.TraceUSER	Filtering by current privilege levels 1-3	17
IPT.TSC	Enable timestamp counter packets	17
Connectors		18
Intel® MIPI60 Connector		18
MIPI60-P Connector		19

XC800

XC800 Debugger	(debugger_xc800.pdf)	1
Introduction		4
Brief Overview of Documents for New Users		4
Warning		5
Quick Start		6
Troubleshooting		8
SYStem.Up Errors		8

FAQ	8
Configuration	9
XC800 Specific Implementations	10
Breakpoints	10
Software Breakpoints	10
On-chip Breakpoints	10
CPU specific SYStem Settings and Restrictions	11
SYStem.state	Open system window 11
SYStem.CONFIG	Configure debugger according to target topology 11
SYStem.CONFIG.CORE	Assign core to TRACE32 instance 16
SYStem.CONFIG.state	Display target configuration 17
SYStem.CPU	Select CPU 17
SYStem.MemAccess	Select run-time memory access method 18
SYStem.Mode	Establish communication with the target 19
SYStem.LOCK	Tristate the JTAG port 19
System Options	21
SYStem.Option.IMASKASM	Disable interrupts while single stepping 21
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping 21
SYStem.Option.LittleEndian	Treat memory as little endian 21
SYStem.Option.TRAPEN	Change the TRAP_EN bit 22
SYStem.JtagClock	Define JTAG clock 23
TrOnchip Commands	24
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource 24
TrOnchip.RESet	Set on-chip trigger to default state 24
TrOnchip.state	Display on-chip trigger window 24
TrOnchip.VarCONVert	Adjust complex breakpoint in on-chip resource 25
OCDS1 Connector	26
Memory Classes	28
XC800 Application Notes	
Application Note Debug Cable XC800	(xc800_app_ocds.pdf) 1
Introduction	4
Debug Cables	5
OCDS Debug Cables	5
Debug Interface Description	11
JTAG Interface	11
DAP Interface	14
Configuring PowerView	16
Selecting the Interface Mode	16
Enabling the DAP Interface on the Chip	16

DAP User Pins	17
Adapters, Converters and Extensions	19
Adapter 16-pin 100 mil to 50 mil	19
Converter 16-pin JTAG to DAP for TriCore/XC2000/XC800	20
Recommended Connectors	21
Standard 2x8 Connector	21
Half-size 2x8 Connector	21
Half-size 2x5 Connector	22
Half-size 2x5 Connector with Keying Pin 7	22

Xtensa

Xtensa Debugger and Trace	(debugger_xtensa.pdf)	1
History		5
Introduction		6
Brief Overview of Documents for New Users		6
Demo and Start-up Scripts		6
Warning		7
Quick Start of the JTAG Debugger		8
Troubleshooting		11
SYStem.Up Errors		11
FAQ		11
Xtensa Specific Implementations		12
Breakpoints		12
Runtime Measurement		14
Memory Classes		14
MAP.BUS8	Bus width mapping	14
MAP.BUS16	Bus width mapping	15
MAP.BUS32	Bus width mapping	15
CPU specific SYStem Commands		16
SYStem.CONFIG.state	Display target configuration	16
SYStem.CONFIG	Configure debugger according to target topology	17
SYStem.CPU	Select the used CPU	41
SYStem.JtagClock	Define JTAG frequency	42
SYStem.LOCK	Tristate the JTAG port	44
SYStem.MemAccess	Select run-time memory access method	44
SYStem.Mode	Establish the communication with the target	45
SYStem.Option.AHBHPROT	Select AHB-AP HPROT bits	45
SYStem.Option.AXIACEEnable	ACE enable flag of the AXI-AP	46
SYStem.Option.AXICACHEFLAGS	Configure AXI-AP cache bits	46

SYStem.Option.AXIHPROT	Select AXI-AP HPROT bits	46
SYStem.Option.DAP2DBGPWRUPREQ	Force debug power in DAP2	47
SYStem.Option.DAPDBGPWRUPREQ	Force debug power in DAP	48
SYStem.Option.DAPNOIRCHECK	No DAP instruction register check	48
SYStem.Option.DEBUGPORTOptions	Options for debug port handling	49
SYStem.Option.DAPREMAP	Rearrange DAP memory map	50
SYStem.Option.DAP2SYSPWRUPREQ	Force system power in DAP2	50
SYStem.Option.DAPSYSPWRUPREQ	Force system power in DAP	51
SYStem.Option.DISableHwWatchDOG	Disable watchdog when core stops	51
SYStem.Option.DisMode	Define disassembler mode	52
SYStem.Option.Endianness	Specify the byte ordering	52
SYStem.Option.EnReset	Allow the debugger to drive nRESET (nSRST)	53
SYStem.Option.EnTRST	Allow debugger to drive TRST	53
SYStem.Option.IMASKASM	Disable interrupts while single stepping	53
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	54
SYStem.Option.IntelSOC	Slave core is part of Intel® SoC	54
SYStem.Option.MMUSPACES	Separate address spaces by space IDs	54
SYStem.Option.PWROVR	Specifies power override bit	55
SYStem.Option.SOFTLONG	Use 32-bit access to set breakpoint	55
SYStem.Option.ResetDetection	Supervise reset	56
SYStem.Option.RUNSTALLMASKASM	Disable RunStall while step	56
SYStem.Option.SnoopAddressPC	Program counter snoop address	56
SYStem.Option.SPILLLOC	Temporary memory	57
SYStem.Option.TriggerHwWatchDOG	Trigger hardware watchdog	57
SYStem.Option.WindowVectorBase	VECBASE initial value	57
SYStem.Option.WinRegOption	Windowed register option	58
SYStem.TIE	TIE library files	59
SYStem.TIE.AddCoreLibrary	Add library file	59
SYStem.TIE.CMList	Instructions to display custom registers	59
SYStem.TIE.DELeTe	Remove all library files	60
SYStem.TIE.DISable	Unload and disable TIE instructions	60
SYStem.TIE.ENABLE	Load and enable TIE instructions	60
SYStem.TIE.GENper	Generate peripheral file	61
SYStem.TIE.GETArchOPTIONS	Detect architectural options from libraries	61
SYStem.TIE.ToolLibraryPath	Specify path for library tools	62
SYStem.TIE.RESet	Reset TIE	62
Xtensa Specific Benchmarking Commands		63
BMC.<counter>.EVENT	Assign event to counter	63
BMC.<counter>.KRNLCNT	Set compare operator	64
BMC.<counter>.TRACELEVEL	Set counting threshold	64
BMC.<counter>.TRACESCOPE	Set counting threshold	65
CPU specific TERM.METHOD Command		66
TERM.METHOD.BRK1_14	Define communication protocol	66

CPU specific TrOnchip Commands		67
TrOnchip.BIEN	Break-out relay enable	67
TrOnchip.BOEN	Break-in relay enable	67
TrOnchip.CTIEN	Cross-trigger input enable	68
TrOnchip.CTOWS	Cross-trigger output enable when trace stop completes	68
TrOnchip.CTOWT	Cross-trigger output enable when trace stop triggered	68
TrOnchip.PTIEN	Processor trigger input enable	69
TrOnchip.PTOWS	Processor trigger output enable	69
TrOnchip.PTOWT	Processor trigger output enable	69
TrOnchip.RESet	Reset on-chip trigger settings	70
TrOnchip.state	Display on-chip trigger window	70
CPU specific MMU Commands		71
MMU.DUMP	Page wise display of MMU translation table	71
MMU.List	Compact display of MMU translation table	73
MMU.SCAN	Load MMU table from CPU	74
CPU specific NEXUS Commands		76
NEXUS.CLOCK	Specify the frequency of the timestamp counter	76
NEXUS.ON	Switch the NEXUS trace port on	76
NEXUS.RESet	Reset NEXUS trace port settings	77
NEXUS.TraceID	Specify the trace ID	77
NEXUS.TimeMode	Generate timestamps to the trace data	77
JTAG Connection		78
IDC20A Debug Cable		78
14-Pin Debug Cable		79

ZSP

ZSP Debugger	(debugger_zsp.pdf)	1
Introduction		5
Brief Overview of Documents for New Users		5
Demo and Start-up Scripts		5
ESD Protection		6
FAQ		6
Quick Start JTAG		7
Troubleshooting		9
System Up Errors		9
Hardware and Software Debug Modes		11
CPU specific TrOnchip Commands		15
TrOnchip.CONVert	Adjust range breakpoint in on-chip resource	15
TrOnchip.RESet	Set on-chip trigger to default state	15

TrOnchip.state	Display on-chip trigger window	15
Memory Access		16
Memory Addressing (ZSP400)		16
Memory Addressing (ZSP500)		17
CPU specific SYStem Settings		18
SYStem.CPU	Selects the processor type	18
SYStem.JtagClock	Selects the frequency for the debug interface	19
SYStem.MemAccess	Select run-time memory access method	20
SYStem.Mode	Selects target reset mode	21
SYStem.CONFIG	Configure debugger according to target topology	24
SYStem.Option.ADIAFTERBREAKIN	Handling of external breakpoints	29
SYStem.Option.BREAKOUTAFTERSWBREAK	Creating a break-out signal	30
SYStem.Option.EnReset	Allow the debugger to drive nRESET (ZSP400)	30
SYStem.Option.HardwareDebug	Select debug mode (ZSP5xx)	31
SYStem.Option.IBOOT	Configure IBOOT board signal (ZSP5xx)	31
SYStem.Option.IMASKASM	Disable interrupts while ASM single stepping	31
SYStem.Option.IMASKHLL	Disable interrupts while HLL single stepping	32
SYStem.Option.IntelSOC	Slave core is part of Intel® SoC	32
SYStem.Option.MEMDEU	Memory access via DEU (ZSP5xx)	32
SYStem.Option.RisingTDO	TDO sampled on rising TCK edge (LSI402ZX)	33
SYStem.Option.SLOWRESET	Slow reset	33
SYStem.Option.SVTADDR	Configure SVTADDR (ZSP500)	33
Multicore Debugging		34
SYStem.LOCK	Lock debug port (ZSP400)	34
Design Decisions and Limitations (ZSP5xx)		35
Disassembler		35
Timer0, Timer1 Registers		35
Single Stepping over RETI Fails		35
Software Breakpoints in CEXE Blocks		35
On-chip Breakpoints (ZSP500 Hardware Erratum)		36
Software Debug Mode and Hardware Debug Mode		36
Simulator Interface for ZSP5xx Cores		37
Limitations of the Simulators		37
File I/O with Simulator Targets		37
Performance Measurements with Simulator Interface		38
JTAG Connector		39
Mechanical Description of 20-pin Debug Cable for ZSP400/ZSP500		39
Mechanical Description of JTAG Connector for ZSP400 (obsolete)		39
Technical Data		40
Operation Voltage		40

IProbe User's Guide

IProbe User's Guide	(iprobe_user.pdf)	1
Introduction		4
IProbe Features		5
Functional Overview		6
Timing Probe Features		7
Analog Probe Features		9
IProbe Input Connector Location		11
Timing Trace Setup and Configuration		12
POD threshold levels and signal display		13
Signal Names		13
General IProbe Functions		19
IProbe Trace Control		19
Operation Modes		22
Automatic Trace Control		23
Using the Trigger		23
Trace Display		24
Signal Naming		24
The IProbe.List Command		25
The IProbe.Timing Command		27
The IProbe.GET Command		29
The IProbe.View Command		30
The IProbe.DRAW Command		31
Tracking		33
Search and Compare		35
Real-Time Displays		35
Saving Trace Buffers		36
Simple Trigger		39
Simple Trigger for Timing Mode		39
Simple Trigger for Analog Mode		47
Universal Counter Signal Selection		47
Protocol Analysis		49
Timing Mode Restrictions		51
Analog Probe		52
Simple Trigger for Analog Mode		52
Voltage Measurement		52
Current Measurement		53

Power Measurement	55
Energy Analysis	57
Analog Trace Time Coverage Calculation	60
Analog Trace Setup and Configuration	16

PowerIntegrator

PowerIntegrator User's Guide

PowerIntegrator User's Guide	(powerintegrator_user.pdf) 1
Functional Units	4
PowerIntegrator	5
Support Package	6
Input/Output Lines	6
ITRIGGER OUT Connector	7
Probe Connector Assignments	8
Mictor Probe	8
Mictor Difference Probe	9
Standard Probe	9
SAMTEC Probe	10
General Functions	11
Initialization	11
Signal Names	11
POD Threshold Levels and Signal Display	12
Sampling Modes	13
Sampling Mode Configurations	14
Sampling Clock Configuration	15
Analyzer Function	16
Analyzer Control	17
Basic Trace Control	17
Operation Modes	19
Automatic Trace Control	19
Analyzer Display	20
Display Commands	20
Search and Compare	25
Tracking	26
Real-Time Displays	26
Saving Trace Buffers	27
Simple Trigger	28

Function	28
Trigger Word	29
Trigger Combiner	29
Trigger PreDelay	30
Trigger Filter	30
Trigger Counter	30
Trigger Delay	30
Trigger Outputs	31
Trigger Setting	31
Using the Trigger Delay and Predelay	32
Complex Trigger	33
Universal Counter	34
Function	34
Signal Selection	35
Setup	36

PowerIntegrator Programming Guide

PowerIntegrator Programming Guide (powerintegrator_prog.pdf)	1
PowerIntegrator	4
Program Structure	5
Sample Trigger Program	6
Declarations	7
Data Selectors	7
Event Counters	8
Flags	9
Time Counters	9
Global Instructions	10
Local Instructions	11
Output Command table	12
Events	13
Counter Events	13
Data Selectors	14
Flags	14
Time Events	14
Other Events	15
Conditions	16
Levels	18
CONTinue	18
GOTO	18

TRIGGER, BREAK	19
Programming Examples	20
Selective Recording	21
Stopping the PowerIntegrator	23
Stimulating Output Lines	23
Using the Internal Trigger Bus	24
Time and Event Counters	25
Using Flags	26
Switching Trigger Levels	26
Declaration Reference	27
SELECTOR	Data selectors 27
EVENTCOUNTER	Event counter 28
EXTERNSYNCCOUNTER	Synchronous counter 30
FLAGS	Flags 31
TIMECOUNTER	Time counter 31
Instruction Reference	34
BREAK	Trace stop 34
Bus	Bus trigger 34
CONTinue	Sequential level switching 35
Counter	Counter control 36
Flag	Flag control 39
GOTO	Level switching 39
Out	Output control 40
Sample	Recording control 41
Trigger	Trigger control 43
PowerIntegrator Programming Language Syntax	44

PowerIntegrator Application Notes

PowerIntegrator State Trace Application Note

PowerIntegrator State Trace Application Note(powerintegrator_app_state.pdf)	1
State Recording	3
State recording by use of 250 MHz Mode	4
State recording by use of STATE-MODE	6
State recording by use of STATEPLL-MODE	8

PowerIntegrator Trace DisConfig Application Note

PowerIntegrator Trace DisConfig Application Note(powerintegrator_app_dc.pdf)	1
General Function	3

<trace>.DisConfig Commands	3
How to use Trace.DisConfig	4
Signal Group Definition	4
Transient Definition	4
DataCycle Definition	5
AddressCycle Definition	6
Address Calculation	6
Trace Display Definition	6
Example for a MC68332 like Bus	8
Example for a SDRAM bus (MPC8280)	10

PowerIntegrator Setup Application Note

PowerIntegrator Setup Application Note (powerintegrator_app_setup.pdf)	1
General	3
Pin / Name Assignment	4
Integrator Hardware Setup	5
Threshold Level	5
Type of Recording	7
Filtering	10
State Mode	10
Transient Detection	10
SupportPackage	11
Selective Trace	11
Analysis	12
Protocol Analysis	12
Disassembly for Bustrace	13

PowerProbe

PowerProbe User's Guide

PowerProbe User's Guide(powerprobe_user.pdf)	1
Functional Units	4
Timing Analyzer	5
SOC Interface (Optional)	6
Pattern Generator (Optional)	6
Pulse Generator	6
Input/Output Lines	7
Input Connector Assignments	8

General Functions	9
Initialisation	9
Signal Names	9
POD threshold levels and signal display	10
Timing Analyzer	11
Function	11
Analyzer Control	13
Basic Trace Control	13
Operation Modes	14
Automatic Trace Control	15
Using the Trigger Delay and Predelay	15
SOC Trace	16
Display Trace	17
Display Commands	17
Tracking	20
Search and Compare	21
Real-Time Displays	21
Saving Trace Buffers	22
Simple Trigger	23
Function	24
Trigger Channel Selector	24
Trigger Word	25
Trigger Combiner	26
Trigger PreDelay	26
Trigger Filter	26
Trigger Counter	26
Trigger Delay	27
Trigger Out	27
Trigger Setting	27
Complex Trigger	28
Asynchronous Trigger	29
Asynchronous Trigger Setting	32
Pattern Generator	33
Function	33
System Control	34
Clock Generator	35
Trigger System	35
Pattern Storage	36
Pattern Display	39
Counter	41
Signal Selection	41

Universal Counter	42
Function	42
Display Window	43
Setup	44
Pulse Generator	45
Function	45
Setup	46
Examples	46

PowerProbe/Port Analyzer Reference Guide

PowerProbe/Port Analyzer Reference Guide	(powerprobe_ref.pdf)	1
Timing/State Analyzer		5
Probe.ASYNC	Asynchronous trigger system	5
Probe.ASYNC.Clock	Defines clock mask	5
Probe.ASYNC.ClockPOL	Defines data polarity	5
Probe.ASYNC.Data	Defines data mask	5
Probe.ASYNC.DataPOL	Defines data polarity	6
Probe.ASYNC.Mode	Defines data polarity	6
Probe.ASYNC.state	State display	7
Probe.ASYNC.Time	Time setting for pulse width trigger	7
Probe.Break	Stop trace	7
Probe.ComPare	Compare trace buffer	8
Probe.CSElect	Select signal for counter	8
Probe.EXPORT	Generate VHDL wait file	9
Probe.FILE	Load trace file	9
Probe.Find	Find entry	10
Probe.FindChange	Find entry	11
Probe.Get	Display input level	12
Probe.List	Display trace buffer	13
Probe.LOAD	Load reference buffer	17
Probe.Mode	Configuration	18
Probe.Program	Program trigger unit	19
Probe.Rate	Select sampling rate	21
Probe.ReProgram	Program trigger unit	21
Probe.SELect	Select SOC signal for trace	22
Probe.SIZE	Select buffer size	22
Probe.state	Display state	23
Probe.SynchClock	Define synchronous clock	24
Probe.TCount	Set trigger counter	25
Probe.TDelay	Define trigger delay	26
Probe.Timing	Display trace contents as timing diagram	28
Probe.TOut	Enable trigger output line	30

Probe.TPreDelay	Pre-trigger delay	30
Probe.TRIGGER	Manual trigger	31
Probe.TSElect	Select trigger source	31
Probe.TSYNC	Select trigger line and mode	32
Probe.TView	Display trigger settings	34
Probe.TWidth	Set trigger filter	35
Probe.View	Display single record	36
Generic Probe Trace Commands		37
Probe.Arm	Arm the trace	37
Probe.AutoArm	Arm automatically	37
Probe.AutoInit	Automatic initialization	37
Probe.BookMark	Set a bookmark in trace listing	37
Probe.Chart	Display trace contents graphically	37
Probe.DISable	Disable the trace	37
Probe.DisConfig	Trace disassembler configuration	38
Probe.DRAW	Plot trace data against time	38
Probe.FindAll	Find all specified entries in trace	38
Probe.GOTO	Move cursor to specified trace record	38
Probe.Init	Initialize trace	38
Probe.ListNesting	Analyze function nesting	38
Probe.ListVar	List variable recorded to trace	38
Probe.OFF	Switch off	38
Probe.PROfileChart	Profile charts	39
Probe.PROTOcol	Protocol analysis	39
Probe.PROTOcol.Chart	Graphic display for user-defined protocol	39
Probe.PROTOcol.Draw	Graphic display for user-defined protocol	39
Probe.PROTOcol.EXPORT	Export trace buffer for user-defined protocol	39
Probe.PROTOcol.Find	Find in trace buffer for user-defined protocol	39
Probe.PROTOcol.list	Display trace buffer for user-defined protocol	39
Probe.PROTOcol.PROfileChart	Profile chart for user-defined protocol	40
Probe.PROTOcol.PROfileSTATistic	Profile chart for user-defined protocol	40
Probe.PROTOcol.STATistic	Display statistics for user-defined protocol	40
Probe.REF	Set reference point for time measurement	40
Probe.RESet	Reset command	40
Probe.SAVE	Save trace for postprocessing in TRACE32	40
Probe.SelfArm	Automatic restart of trace recording	40
Probe.SnapShot	Restart trace capturing once	41
Probe.TRACK	Set tracking record	41
Probe.ZERO	Align timestamps of trace and timing analyzers	41
Counter		42
Pattern Generator		43
PATTERN.Arm	Arm analyzer	43

PATTERN.CEnable	Pattern clock control	43
PATTERN.CMode	Pattern clock select	44
PATTERN.GOTO	Jump to entry	44
PATTERN.Init	Initialization	45
PATTERN.List	Display pattern memory	45
PATTERN.OFF	Display pattern generator	49
PATTERN.Program	Program pattern generator	49
PATTERN.REF	Set reference point	50
PATTERN.ReProgram	Program pattern generator	50
PATTERN.RESet	Reset pattern generator	51
PATTERN.state	Display state	52
PATTERN.Step	Single step function	53
PATTERN.TEST	Run pattern generator	53
PATTERN.Timing	Display pattern memory	54
PATTERN.TLatch	Trigger latch	55
PATTERN.TMode	Trigger mode	56
PATTERN.TSElect	Trigger input select	57
PORT		58
PORT.SET	Set port value	58
PORT.SLAVE	Select slave mode	58
PULSE		59
PULSE	Pulse generator	59
PULSE.BusA	Trigger on “BusA” line	60
PULSE.PERiod	Cycle duration	60
PULSE.POLarity	Polarity	61
PULSE.Pulse	Programming	62
PULSE.RESet	Reset command	62
PULSE.Single	Release single pulse	63
PULSE.state	State display	64
PULSE.Width	Pulse width	64
RESet	General reset command	65

PowerProbe Trigger Unit Programming Guide

PowerProbe Trigger Unit Programming Guide	(powerprobe_prog.pdf)	1
PowerProbe Programming		4
Program Structure		5
Sample Trigger Program		6
Declarations		7
Data Selectors		7
Event Counters		8
Flags		8

Time Counters	8
Synchronous Counters	9
Global Instructions	10
Local Instructions	11
Output Command Table	12
Events	13
Counter Events	13
Data Selectors	13
Flags	14
Time Events	14
Other Events	14
Conditions	15
Levels	17
CONTinue	17
GOTO	17
TRIGGER, BREAK	18
Programming Examples	19
Selective Recording	19
Stopping the PowerProbe	21
Stimulating Output Lines	21
Using the Internal Trigger Bus	21
Time and Event Counters	23
Using Flags	24
Switching Trigger Levels	25
Declaration Reference	26
SELECTOR	Data selectors 26
EVENTCOUNTER	Event counter 27
EXTERNSYNCCOUNTER	Synchronous counter 29
FLAGS	Flags 30
TIMECOUNTER	Time counter 30
Instruction Reference	33
BREAK	PowerProbe stop 33
Bus	Bus trigger 33
CONTinue	Sequential level switching 34
Counter	Counter control 35
Flag	Flag control 38
GOTO	Level switching 39
Out	Output control 39
Sample	Recording control 40
Trigger	Trigger control 42

Protocol Analyzer

Protocol Analyzer Application Note

Protocol Analyzer Application Note	(protocol_app.pdf)	1
Introduction		3
A short Introduction to DLLs		4
Overview		6
PROTO_Init		7
Process Callback Functions		12
Display Callback Function		16
Draw Callback Function		18

FlexRay Protocol Analyzer

FlexRay Protocol Analyzer	(flexray_app.pdf)	1
General Function		3
Features		3
How to use the PROTOanalyzer		4
<trace>.List	LIST display	5
<trace>.Chart	CHART display	6
<trace>.STATistic	Statistic visualisation	7
Fibex Implementation		7
FAQ		9

LIN Bus Protocol Analyzer

LIN Bus Protocol Analyzer	(linbus_app.pdf)	1
General Function		3
Features		4
How to use the PROTOanalyzer		5
<trace>.List	List display	6
Explanation of LIST Display		9
<trace>.Chart	Chart display	12
<trace>.STATistic	Statistic visualization	12
Overview over Restrictions		13

Restrictions of Configuration Services	13
Restriction of LDF Parser	14

EPROM/FLASH Simulator

EPROM/FLASH Simulator(eprom_simulator.pdf)	1
Introduction	4
Basics	4
Warning	4
Configuration	5
ICD Configuration for ROM Monitor	5
Mapping	7
Mapping the EPROM Simulator	7
Mapping the EPROM Simulator for BDM/ROM	8
Mapper Commands	11
Data Access	12
Break and Exception Control	13
Break	13
Count	14
Counter	14
Counter Commands	15
eXception	16
eXception.ICEINTPOL	Polarity of ICEINT line 16
eXception.NMIBREAK	Break through NMI 17
eXception.NMIDTR	Break through DTR line 17
eXception.NMIPOL	Polarity selection of NMI signal 17
eXception.NMIRTS	Break through RTS line 17
eXception.RESet	Default settings 17
eXception.RESetDTR	Reset through DTR line 18
eXception.RESetPOL	Polarity of RESET signal 18
eXception.RESetRTS	Reset through RTS line 18
eXception.view	Show exception settings 18
RESET	19
RESet	Initialize simulator 19
SYStem Commands	20
SYStem.Down	Deactivates simulator 20
SYStem.Mode	Selects operation mode 20
SYStem.Up	Activates simulator 21
SYStem.state	Shows operation mode 21

Store Settings	22
AutoSTOre	Autosave of settings 22
ClipSTOre	Store a setting to clipboard 23
STOre	Store a setting 23
Adapters	25
Adapter Configuration	25
Pinout Adapters	29
ESICON Adapter Function	38

Stimuli Generator

Stimuli Generator User's Guide

Stimuli Generator User's Guide	(stg_user.pdf)	1
Stimuli-Generator		3
Basics		3
Digital Port Function		4
Digital Commands		7
Analogous Port Functions		7
Analogous Commands		7
State line		8
Pattern Generator		9
Function		9
System Control		10
Clock Generator		11
Trigger System		11
Pattern Storage		12

Stimuli Generator Reference Guide

Stimuli Generator Reference Guide	(stg_ref.pdf)	1
AutoSTOre		4
AutoSTOre	Store setups automatically	4
Con		6
CON	Enable connection tester	6
Count		7
Level Display		8
Counter Functions		9
Count.AutoInit	Automatic counter reset	9
Count.Gate	Gate time	10

Count.GO	Start measurement	10
Count.Init	Reset counter	11
Count.Mode	Mode selection	11
Count.OUT	Switch counter input signal to BNC	13
Count.PROfile	Graphic counter display	13
Count.RESet	Reset command	15
Count.Select	Select input source	15
Count.state	State display	16
Get		17
Get	Show input levels	17
IN		18
IN.Mode	Define input mode	18
IN.PROfile	Graphic input level display	19
IN.RESet	Reset analog input unit	19
IN.view	Show analog input values	20
Mode		21
Mode	Select input/output	21
NAME		22
NAME.RESet	Remove pod names	22
NAME.Set	Define pod names	22
NAME.view	Show pod names	23
OUT		24
OUT.RESet	Reset analog output unit	24
OUT.Set	Define output voltage	24
OUT.view	Show analog output values	25
Pattern Generator		26
Pattern.Arm	Arm analyzer	26
Pattern.CEnable	Pattern clock control	26
Pattern.CMode	Pattern clock select	27
Pattern.Init	Initialization	27
Pattern.OFF	Disable pattern generator	28
Pattern.Program	Program pattern generator	28
Pattern.ReProgram	Program pattern generator	30
Pattern.RESet	Reset pattern generator	30
Pattern.state	Display state	31
Pattern.Step	Single step function	32
Pattern.TEST	Run pattern generator	32
Pattern.TLatch	Trigger latch	33
Pattern.TMode	Trigger mode	33
Pattern.TSelect	Trigger input select	34
PULSE		35

PULSE.PERiod	Cycle duration	35
PULSE.Pulse	Pulse programming	36
PULSE.RESet	Reset command	37
PULSE.SELect	Select output line	37
PULSE.Single	Release single pulse	38
PULSE.view	View setup	39
PULSE.Width	Pulse width	39
RESet		40
Set		41
STORE		42
STOre	Store setups	42

Misc

Error Messages

Error Messages	(error.pdf)	1
General Error Messages		3
General Command Parameter Parser		20
CONNECTION Error Messages		44
Debugger		45
Error Messages Related to the Peripheral View (PER)		45
Error Messages Related to the Peripheral File Converters		47
Error Messages Related to FLASH Programming		65
Error Messages Related to Co-Processor Debugging		70
Error Messages Related to HiPerLoad		71
Error Messages Related to FDX		72
Error Messages Related to Terminal Function		73
Error Messages Related to MMU Address Translation		74
Error Messages Related to RTOS Support		75
Error Messages Related to Differential Download		77
Error Messages Related to Breakpoints		78
Error Messages Related to Debugging		89
Error Messages Related to Debug Hardware and Software		98
Error Messages Related to Analyzer/Trace		101
Error Messages Related to MCDS		103
Error Messages Related to Trace Testfocus/Autofocus		104
Error Messages Related to APU API		107
HLL Expression Parser		108

Inline Assembler	111
Analyzer Trigger Unit Programming	118
Performance Analyzer	160
Timing Analyzer Trigger Unit Programming	161
Stimuli Generator	179

API for Remote Control and JTAG Access in C

API for Remote Control and JTAG Access in C(api_remote_c.pdf)	1
History	6
Licensing Terms	7
Introduction	7
Release Information	7
Compatibility	8
Related Tutorials	8
System Configuration Overview	9
Restrictions in Demo Mode	9
Interfaces	10
Operation of API Requests	12
Conventions for Target Memory Access	12
Building an Application with API	15
API Files	15
Connecting API and Application	15
Logging the API Calls	16
Communication Setup	17
Preparing TRACE32 Software	17
Configuring the API	17
API Functions	18
Error Codes	18
Generic API Functions	18
Functions for using the API with Multiple Debuggers	44
API Functions	50
ICD Direct Access API Functions	123
API Object Handling	180
Buffer Object	181
Address Object	183
Bundle Object	188
Register Object	190
RegisterSet Object	195
Breakpoint Object	197

Controlling TRACE32 via Python 3

Controlling TRACE32 via Python 3	(app_python.pdf)	1
History		3
About this Manual		4
Introduction		4
PYRCL versus TRACE32 Legacy Approach		5
lauterbach.trace32.rcl (PYRCL)		6
Versioning		6
Package		7
Documentation		7
TRACE32 Legacy Approach		8
Establish and Release the Communication to the Debug Device		9
Run a PRACTICE Script		14
TRACE32 Functions		18
Monitor a Variable		19

API for Auxiliary Processing Unit

API for Auxiliary Processing Unit	(api_apu.pdf)	1
Introduction		4
Release Information		4
Features		5
Requirements		5
Files		6
Conventions		6
Programmer's Guide		7
Basic Concept		7
Callback Functions		8
Access to Main Core Debugger		9
Generic Configuration		9
Output Functions		9
Interface Functions		9
APU Callback Structures		10
APU Context		10
APU Library		11
APU API Files		11
Building the Library		11

Loading the Library	12
Writing a new Library	12
Symbol Information	14
APU Commands	15
APU Library Functions	16
APU Entry Functions	16
Generic Configuration Functions	17
Callback Register Functions	21
Memory and Target Access Functions	28
APU Callback Structures	31
Breakpoint Callback Structure	31
Global Callback Structure	32
Disassembler Callback Structure	33
Assembler Callback Structure	35
GetState Callback Structure	36
Memory Callback Structure	37
Parameter Callback Structure	38
Translate Callback Structure	39
Version Control	40

Command List

Command List	(commandlist.pdf)	1
Parameters		4
Operators		4
Arithmetic Rules and Operator Precedence		7
Functions		8
Operation System Commands		35
PRACTICE Commands		40
General Emulator/Debugger/Trace Commands		43
A		43
B		53
C		59
D		71
E		77
F		82
G		88
H		89
I		93
J		99
K		101

L	102
M	106
N	112
O	113
P	115
Q	119
R	119
S	122
T	136
U	150
V	150
W	153
X	153
Y	153
Z	153
PowerProbe	154
PowerIntegrator	158

Debugger Training

Training Basic Debugging

Training Basic Debugging	(training_debugger.pdf) 1
System Concept	6
On-chip Debug Interface	7
On-chip Debug Interface plus On-chip Trace Buffer	10
On-chip Debug Interface plus Trace Port	12
NEXUS Interface	13
Starting a TRACE32 PowerView Instance	14
Basic TRACE32 PowerView Parameters	14
Application Properties (Windows only)	21
Configuration via T32Start (Windows only)	22
About TRACE32	23
Establish your Debug Session	25
TRACE32 PowerView	26
TRACE32 PowerView Components	26
Registers	40
Core Registers	40
Special Function Register	43
Memory Display and Modification	49
The Data.dump Window	50
The List Window	67
Breakpoints	70
Breakpoint Implementations	70
Breakpoint Types	81
Breakpoint Handling	86
Breakpoint Setting at Run-time	86
Real-time Breakpoints vs. Intrusive Breakpoints	87
Break.Set Dialog Box	89
Advanced Breakpoints	113
Display a List of all Set Breakpoints	138
Delete Breakpoints	139
Enable/Disable Breakpoints	139
Store Breakpoint Settings	140

Debugging	141
Debugging of Optimized Code	141
Basic Debug Control	144
Sample-based Profiling	156
Program Counter Sampling	156
TASK Sampling	163

Training Basic SMP Debugging

Training Basic SMP Debugging	(training_debugger_smp.pdf) 1
System Concept	6
On-chip Debug Interface	7
On-chip Debug Interface plus On-chip Trace Buffer	10
On-chip Debug Interface plus Trace Port	12
NEXUS Interface	13
Starting a TRACE32 PowerView Instance	14
Basic TRACE32 PowerView Parameters	14
Application Properties (Windows only)	21
Configuration via T32Start (Windows only)	22
About TRACE32	23
Establish your Debug Session	25
TRACE32 PowerView	26
SMP Concept	26
TRACE32 PowerView Components	29
Basic Debugging (SMP)	44
Go/Break	44
Single Stepping on Assembler Level	46
Single Stepping on High-Level Language Level	48
Registers	50
Core Registers	50
Special Function Register	53
Memory Display and Modification	61
The Data.dump Window	63
The List Window	80
Breakpoints	83
Breakpoint Implementations	83
Breakpoint Types	94
Breakpoint Handling	99
Breakpoint Setting at Run-time	99
Real-time Breakpoints vs. Intrusive Breakpoints	100

Break.Set Dialog Box	104
Advanced Breakpoints	126
Display a List of all Set Breakpoints	145
Delete Breakpoints	146
Enable/Disable Breakpoints	146
Store Breakpoint Settings	147
Debugging	148
Debugging of Optimized Code	148
Basic Debug Control	151

Training Arm ETM

Training Arm CoreSight ETM Tracing

Training Arm CoreSight ETM Tracing(training_arm_etm.pdf)	1
ETM Setup	6
ETM Versions	6
Main Setup Windows	7
ETMv1	9
ETMv3	16
PTM (aka. PFT)	27
FLOWERROR	38
Displaying the Trace Contents	41
Source for the Recorded Trace Information	41
Sources of Information for the Trace Display	43
Influencing Factors on the Trace Information	44
States of the Trace	57
The Autolnit Command	58
Basic Display Commands	59
Display Items	63
Find a Specific Record	70
Belated Trace Analysis	72
Trace-based Debugging (CTS)	78
Forward and Backward Debugging	79
CTS Technique	84
Belated Trace-based Debugging	86
HLL Analysis of the Trace Contents	87
Trace Control by Filter and Trigger	91
Context	91
Filters and Trigger by Using the Break.Set Dialog	95

OS-Aware Tracing	114
OS (No Dynamic Memory Management)	114
OS+MMU (Dynamic Memory Management)	124
Specific Write Access vs. Context ID Packet	133
Task Statistics	134
Context ID Comparator	136
Function Run-Times Analysis	138
Software under Analysis (no OS, OS or OS+MMU)	138
Flat vs. Nesting Analysis	138
Flat Analysis	143
Nesting Analysis	166
Trace-based Code Coverage	185
Optimum ETM Configuration (No OS or OS)	185
Optimum ETM Configuration (OS+MMU)	185

Training Cortex-M Tracing

Training Cortex-M Tracing	(training_cortexm_etm.pdf) 1
History	4
Cortex-M Trace	4
Connectors	7
Basic Trace Configuration	8
Trace Buffer Management	10
MTB Program Flow Trace	13
ETM Program Flow Trace	14
ETM Configuration	14
Trace Capture	16
ETM Stream Mode	17
Displaying the Results	18
Trace Searching	21
Trace Filtering	24
Graphical Navigation	28
Analyzing the Results	30
Trace and Groups	40
Timing	45
Trace Based Code Coverage	48
Trace Based Debugging	49
Off-line Analysis	53
Data Watchpoint and Trace Unit	55
Instrumentation Trace Macrocell	66

Training AURIX Tracing

Training AURIX Tracing	(training_aurix_trace.pdf)	1
History		6
Basic Knowledge		7
Protocol Description		7
Source for the Recorded Trace Information		14
Trace Configuration within TRACE32		21
Onchip Trace Configuration		21
Off-chip Trace Configuration		34
Trace Sources and Their Messages		37
Message Display in TRACE32		43
FIFOFULL		53
Displaying the Trace Contents		55
Sources of Information for the Trace Display		55
Influencing Factors on the Trace Information		56
TRACE32 Trace Configuration Window		57
Basic Display Commands		70
Browsing through the Trace Buffer		82
Find a Specific Event		83
Post Mortem Trace Analysis (PowerTrace only)		84
Belated Trace Analysis		86
Trace Control by Filter and Trigger - Overview		93
Marker		94
Filter		94
Trigger		94
Available Resources		94
Filter and Trigger - Single-Core and AMP		95
WATCH Marker		95
TraceEnable Filter		98
TraceData Filter		112
TraceON/TraceOFF Filter		114
Trace Trigger (Onchip Trace Only)		118
Filter and Trigger - SMP Systems		124
WATCH Marker		124
TraceEnable Filter		128
TraceData Filter		144
TraceON/TraceOFF Filter		146
Trace Trigger (Onchip Trace Only)		150

OS-Aware Tracing - Single-Core and AMP	157
Activate the TRACE32 OS Awareness (Supported OS)	157
Exporting the Task Switches	159
Exporting Task Services	163
Exporting ISR2 (OSEK Interrupt Service Routines)	167
Exporting Task Switches and ISR2	171
Exporting Task Switches and all Instructions	173
Belated Trace Analysis (OS)	176
Enable an OS-aware Tracing (Not-Supported OS)	177
OS-Aware Tracing - SMP Systems	178
Activate the TRACE32 OS Awareness (Supported OS)	178
Exporting the Task Switches	180
Exporting Task Services	187
Exporting ISR2 (OSEK Interrupt Service Routines)	193
Exporting Task Switches and ISR2	199
Exporting Task Switches and all Instructions	201
Belated Trace Analysis (OS)	204
Function Run-Time Analysis - Basic Concept	205
Software under Analysis (no OS or OS)	205
Flat vs. Nesting Analysis	205
Flat Function-Runtime Analysis - Single-Core and AMP	210
Optimum MCDS Configuration (No OS)	210
Optimum MCDS Configuration (OS)	211
Function Timing Diagram (no TASK Information)	212
Function Timing Diagram (TASK information)	213
Numeric Analysis	216
Flat Function-Runtime Analysis for SMP	218
Optimum MCDS Configuration (OS)	219
Function Timing Diagram (no TASK Information)	220
Function Timing Diagram (TASK Information)	222
Numeric Analysis	224
Nesting Function Run-Time Analysis - Single	227
Restrictions	227
Optimum MCDS Configuration (No OS)	228
Optimum MCDS Configuration (OS)	229
Numerical Nesting Analysis for all Software	231
More Nesting Analysis Commands	243
Nesting Function Run-Time Analysis for SMP	248
Optimum MCDS Configuration (OS)	248
Numerical Nesting Analysis for OS	251
More Nesting Analysis Commands	261

Trace-based Code Coverage	264
General SetUp	264

Training Hexagon ETM

Training Hexagon ETM Tracing

Training Hexagon ETM Tracing	(training_hexagon_etm.pdf) 1
Introduction Hexagon ETM	5
Off-chip Trace Port	5
On-chip Trace	17
Specifying the Trace Method	24
FLOW ERROR	29
TARGET FIFO OVERFLOW	32
ETM Based Real-Time Breakpoints	35
Introduction	35
Breakpoint Usage	37
Saving the Breakpoint Settings as a PRACTICE Script	54
Displaying the Trace Contents	55
Fundamentals	55
Display Commands	57
Correlating Different Trace Displays	60
Correlating the Trace Display and the Source Code	61
Default Display Items	62
Additional Display Items	75
Formatting the Trace Display	78
Changing the DEFault Display	80
The AutoInit Option	81
Searching in the Trace	82
Belated Trace Analysis	84
Function Run-Times Analysis	90
Flat vs. Nesting Analysis	91
Flat Analysis	94
Nesting Analysis	113
Cycle Statistic	128
Filtering via the ETM Configuration Window	131
Hardware Thread Filter	132
Software Thread Filter	133
ASID Filter	133
Filtering/Triggering with Break.Set	134

TraceEnable Filter	136
TraceON/OFF Filter	144
TraceTrigger	148
Filtering/Triggering via the ETM.Set	156
The ETM Registers	157
Actions Based on Sequencer Level	159
Actions Based on Sequencer Level and Condition	163
Benchmark Counters	167
Introduction	167
Standard Examples	169
Function Run-time Analysis - Cache Misses/Stalls	180
Summary: Trigger and Filter	183
Appendix A	184
The Calibration of the Recording Tool	184
Calibration Problems	186

Training Intel® x86/x64

Training Basic SMP Debugging for Intel® x86/x64

Training Basic SMP Debugging for Intel® x86/x64 (training_debugger_x86.pdf)	1
Debug Configurations	5
CombiProbe 2 MIPI60-Cv2	6
On-Chip Core Trace	9
Off-Chip System/Core Trace	9
Starting a TRACE32 PowerView Instance	10
Basic TRACE32 PowerView Parameters	10
Application Properties (Windows only)	15
Configuration via T32Start (Windows only)	16
About TRACE32	17
Establish your Debug Session	20
Course of Action	20
Start-Up Script	28
TRACE32 PowerView	31
SMP Concept	31
TRACE32 PowerView Components	35
Further Documentation	50
Basic Debugging (SMP)	51
Go/Break	51

Single Stepping on Assembler Level	53
Single Stepping on High-Level Language Level	54
Registers	56
Core Registers	56
Further Register Sets	59
Special Function Register	60
Memory Display and Modification	65
The Data.dump Window	67
The List Window	78
Breakpoints	81
Breakpoint Implementations	81
Onchip Breakpoints for Intel® x86/x64	85
Breakpoint Types	86
Breakpoint Behavior	91
Breakpoint Setting at Run-time	91
Breakpoints after Reset/Power Cycle	92
Onchip Breakpoints Changed by Target Program	94
Breakpoint Handling	95
Real-time Breakpoints vs. Intrusive Breakpoints	95
Break.Set Dialog Box	100
Advanced Breakpoints	115
Display a List of all Set Breakpoints	131
Delete Breakpoints	131
Enable/Disable Breakpoints	132
Store Breakpoint Settings	133
Debugging	134
Basic Debug Control	134
Debugging of Optimized Code	147
Document your Results	150
Settings	150
Quick Output	153
Advanced Output	155

Training Linux Debugging for Intel® x86/x64

Training Linux Debugging for Intel® x86/x64	(training_rtos_linux_x86.pdf) 1
Introduction	5
Documentation Updates	5
Related Documents and Tutorials	5
Basic Terms on Embedded Linux	6
Linux Components	6

The Linux Awareness	7
Virtual Memory Management in Linux	9
Run-Mode vs. Stop-Mode Debugging	14
Kernel Configuration	16
Setting up a Script for Linux-Aware Debugging	18
Linux Setup-Steps and -Commands	18
Example Linux Setup-Scripts	28
Debugging the Linux Components	31
The Kernel	31
Kernel Modules	35
Processes	37
Linux specific Windows	43
Display of System Resources	43
Kernel Module List	44
File System Information	45
Kernel Log Buffer	46
RAM Dump Generation	47
Troubleshooting	48
FAQ	49

Training Intel® Processor Tracing

Training Intel® Processor Tracing	(training_ipt_trace.pdf)	1
Protocol Description		6
Basic Trace Packets		6
OS-Aware Tracing		7
Time Information		8
Trace Configuration		11
Off-chip Trace		11
SDRAM Trace		18
Trace Errors		22
TRACE32 Abstractions		27
Displaying the Trace Contents		31
Influencing Factors on the Trace Information		31
Settings in the TRACE32 Trace Configuration Window		32
States of the Trace		43
The AutoInit Command		44
Basic Display Commands		45
Display Items		51
Belated Trace Analysis		60
Trace Control by Filters		68

TraceEnable	69
TraceOFF	71
OS-Aware Tracing	73
Process Switch Packets	73
Program Flow and Process Switches	75
Process Runtime Analysis	76
Find Process Switches in the Trace	79
OS-aware Filtering	81
Belated Analysis	86
Trace-based Debugging (CTS)	88
Setup	88
Get Started	89
Forward and Backward Debugging	92
CTS Technique	93
Function Run-Time Analysis - Basic Concept	94
Software under Analysis (no OS, OS or OS+MMU)	94
Flat vs. Nesting Analysis	94
Flat Function-Runtime Analysis	99
Function Time Chart	99
Function Run-time Statistic	103
Further Commands	104
Nesting Function Analysis OS	105
Survey	106
range Column	107
Default Results	110
Net Results	112
Interrupt Details	114
Time in Other Tasks	115
Tree Display	116
Structure your Trace Evaluation	117
GROUPs for OS-aware Tracing	117
GROUP Status ENable	118
GROUP Status ENable+Merge	119
GROUP Status Enable+HIDE	120
GROUP Creation	121

Training Nexus Tracing

Training Nexus Tracing	(training_nexus.pdf)	1
Basic Knowledge		8
NEXUS Characteristics		8
Limited Bandwidth		9
Multicore Tracing		18
Source for the Recorded Trace Information		20
NEXUS Configuration by TRACE32		22
Configuration of the Trace Interface		22
Configuration of the NEXUS Messages		33
NEXUS Trace Clients		44
Target FIFO Overflow		48
FlowErrors		53
Displaying the Trace Content		54
Sources of Information for the Trace Display		54
Settings in the Trace Configuration Window		55
Basic Display Commands		67
Display Items		72
Time Synchronization between TRACE32 Instances (AMP)		77
Find a Specific Record		79
Belated Trace Analysis		80
Trace-based Debugging (CTS)		86
Re-Run the Program		86
Re-Run the Program and Watch the Variables		91
CTS Technique		99
Filter and Trigger (Core) Overview		101
Resources		102
Filter and Trigger (Core) - Single Core		106
Examples for TraceEnable on Instructions		106
Example for TraceEnable on Instruction Range		111
Examples for TraceEnable on Read/Write Accesses		114
Example for TraceData		119
Examples for TraceON/TraceOFF		121
Example for TraceTrigger		134
Example for TraceTrigger with a Trigger Delay		137
Example for BusTrigger		140
Example for BusCount (Watchpoint)		142
Filter and Trigger (Core) - SMP Debugging		146

Examples for TraceEnable on Single Instruction	147
Examples for TraceEnable on Instruction Range	153
Examples for TraceEnable on Read/Write Accesses	156
Example for TraceData	161
Examples for TraceON/TraceOFF	163
Example for TraceTrigger	177
Example for TraceTrigger with a Trigger Delay	180
Example for BusTrigger	184
Example for BusCount (Watchpoint)	186
Filter and Trigger (Trace Clients)	190
Example for TraceEnableClient1	191
OS-Aware Tracing (ORTI File)	193
Activate the TRACE32 OS Awareness	193
Exporting Task Information (Overview)	195
OS-Aware Tracing - Single Core	196
Exporting all Types of Task Information (OTM)	196
Exporting all Types of Task Information and all Instructions (OTM)	205
Exporting Task Information (Write Access)	208
Exporting Task Switches and all Instructions (Write Access)	220
Belated Trace Analysis (OS)	223
OS-Aware Tracing - SMP Systems	224
Exporting all Types of Task Information (OTM)	224
Exporting all Types of Task Information and all Instructions (OTM)	235
Exporting Task Information (Write Access)	238
Exporting Task Switches and all Instructions (Write Access)	257
Belated Trace Analysis (OS)	260
Function Run-Times Analysis (Overview)	261
Software under Analysis (no OS or OS)	261
Flat vs. Nesting Analysis	261
Function Run-Times Analysis - Single	266
Flat Analysis	266
Nesting Analysis	276
Third-party Timing Tools	305
Function Run-Times Analysis - SMP Instance	306
Flat Analysis	306
Nesting Analysis	316
Third-party Timing Tools	334
Structure the Trace Evaluation	335
GROUP Creation	335
Working with GROUPs	339

Training Source Level Debugging

Training Source Level Debugging (training_source_level_debugging.pdf)	1
Load the Application Program	5
The Symbol Database	22
Structure of the Internal Symbol Database	22
General Information on the Symbol Database	23
Symbol Browser	24
Details about a Selected Symbol	29
Searching in Source Files	31
Display Variables	33
Watch Window	33
View Window	35
Referenced Variables	36
Local Variables	37
Stack Frame	38
Special Display for Arrays	39
Linked Lists	42
Change a Variable Value	44
Format Variable	47
Format a Variable using the Format Dialog Box	47
Format a Variable Using the Command Line	57
General SETUPS	58
Variable Monitoring	59
Basics	59
Preparation	60
Format Option %E	62
Var.PROfile Command	64
Variable Logging	66
SNOOPer Trace	66
Var.LOG Command	86
Testing of Functions	88

Training Script Language PRACTICE

Training Script Language PRACTICE(training_practice.pdf)	1

History	5
E-Learning	5
Ready-to-Run Scripts	5
Related Documents	5
Introduction to Script Language PRACTICE	6
Area of Use	6
Run a Script	7
Create a PRACTICE Script	9
Convert TRACE32 Settings to a Script	9
Command LOG	14
Command History	15
Script Editor PEDIT	16
Syntax Highlighting	18
Debugging of PRACTICE Script	19
Debug Environment	20
Display the PRACTICE Stack	24
PRACTICE Language	25
Program Elements	25
PRACTICE Flow Control	27
Conditional Program Execution	27
Subroutine Calls	32
GOTO/JUMPTO	35
Script Calls	38
PRACTICE Macros	39
Script Examples	52
Run Through Program and Generate a Test Report	52
Check Contents of Addresses	58
Check Contents of Address Range	59
Check the Contents of Variables	62
Record Formatted Variables	63
Record Variable as CSV	64
Test Functions	66
Test Function with Parameter File	67
Parameter Passing	69
Pass Parameters to a PRACTICE Script or to a Subroutine	69
PARAMETERS/RETURNVALUES vs. ENTRY	77
Operating System Interaction	82
Operating System Detection	82
Printing Results	83
Accessing Environment Variables	85

Running a Command	86
File Manipulation	87
Time and Date Functions	91
I/O Commands	92
Output Command	92
Input Command	92
I/O via the AREA Window	94
Event Control via PRACTICE	96
Simple Dialogs	97
Dialog Programming	100
Control Positioning	103
Control Properties	105
File Browsing	113
Icons	115
Dialog Example	116
PRACTICE in a Multi-Core Environment	121
Communication via InterCom	122
Designing Robust PRACTICE Scripts	124
Path Functions and Path Prefixes	125
Host Operating System	127
Debug Hardware	128
Target CPU and Board	129
TRACE32 Version	130
TRACE32 Settings	131
Robust Error Handling	136
Argument Handling	138
Creating a Custom Command	139
Common Pitfalls	140

Training Menu Programming

Training Menu Programming	(training_menu.pdf)	1
Customizable GUI Elements		3
The TRACE32 Default Menu		4
The Built-in Menu Editor		5
Customizing the Main Menu Bar		7
Adding a New Menu to the Main Menu Bar		7
Altering an Existing Drop-down Menu		12
Customize the Toolbar		17
Add a New Button to the Toolbar		17

Training RTOS

Training Linux Debugging

Training Linux Debugging	(training_rtos_linux.pdf)	1
Introduction		5
Documentation Updates		5
Related Documents and Tutorials		5
Basic Terms on Embedded Linux		6
Linux Components		6
The Linux Awareness		7
Virtual Memory Management in Linux		9
Run-Mode vs. Stop-Mode Debugging		15
Kernel Configuration		17
Setting up a Script for Linux-Aware debugging		20
Linux Setup-Steps and -Commands		20
Example Linux Setup-Scripts		32
Debugging Linux Components		34
The Kernel		34
Kernel Modules		37
Processes		39
Linux Specific Windows		45
Displaying the Task List		45
Kernel Module List		46
File System Information		47
Kernel Log Buffer		48
Device Tree		49
RAM Dump Generation		49
Linux Trace		50
Overview		50
Context ID Trace for Arm Cortex-A		51
OTM Trace for PowerArchitecture based QorIQ Processors		51
Using the LOGGER for Task Switch Trace		52
Troubleshooting		55
FAQ		56

Training Power Probe

Training Power Probe	(training_pp.pdf)	1
Basics		3
The PowerProbe Configuration Window		3
The PowerProbe Connector		4
Standard Sampling Mode		5
Arm/Disarm the PowerProbe		6
Signal Names		7
Link the PowerProbe to the Application Debugging		10
Fifo/Stack Mode		10
Measurement Statistics		11
Postprocessing		12
Simple Trigger		14
Asynchronous Trigger		22
Trigger Outputs		25
Protocol Analysis		26
Track Option		28
Complex Trigger Introduction		30
Synchronous Recording		34
Pulse Generator		37
Pattern Generator		38

Training Simulator and Demo Software

Training Simulator and Demo Software	(demo.pdf)	1
About the Demo		3
Starting the TRACE32 Simulator		3
User Interface - TRACE32 PowerView		4
TRACE32 Command Line and Softkeys		6
Window Captions - What Makes Them Special in TRACE32?		7
Debugging the Program		8
Basic Debug Commands		8
Debug Modes		10
Displaying the Stack Frame		12

Breakpoints	13
Setting Breakpoints	13
Setting Read/Write Breakpoints	14
Listing all Breakpoints	15
Variables	16
Displaying Variables	16
Displaying Variables of the Current Program Context	18
Using the Symbol Browser	18
Formatting Variables	19
Modifying Variables	21
Memory	22
Displaying Memory	22
Modifying Memory	23
Peripheral View	24

TRACE32 Installation Guide

TRACE32 Installation Guide (installation.pdf)	1
History	4
Warning	5
Introduction	6
How This Manual is Organized	6
Contacting Support	6
Brief Overview of Documents for New Users	8
Tool Configuration	9
Power Supply	9
TRACE32 Debug Tools	9
μTrace (MicroTrace)	12
CombiProbe	13
TRACE32 Debug and Trace Tools	14
Software Installation	20
MS Windows	21
PC_LINUX	23
Mac OS	32
Configuration File	35
Section “OS”	37
Section LICENSE	38
Section SCREEN	38
Section PRINTER	41
Section PBI	42
Command Line Arguments for Starting TRACE32	53
Firewall Setup	62
Uninstall TRACE32	63
Microsoft Windows	63
LINUX and Mac OS	63
Troubleshooting	64
FAQ	66
Debugger Licenses	67
Multicore License	67
Software Warranty	67
LEDs on TRACE32 Hardware Modules	68

Floating Licenses

Floating Licenses	(floatinglicenses.pdf)	1
History		4
Introduction		5
TRACE32 Floating License Overview		5
Licensing Terms Glossary		7
How To Install Floating Licenses - Overview		8
How To Upgrade Floating Licenses		8
License Management Server		9
How RLM Floating Licenses work		10
License Server Setup		11
How to get the RLM Host ID		11
How to Configure an RLM Server for Auto-Start		11
Floating License Transfer		14
License Client Setup		15
Multiple RLM servers		18
Supported Operating Systems		19
Floating License Pools		20
Software-Only License Types		21
License Pool Setup		22
Example Session		24
Note		25
License Roaming		26
Using the environment variable		26
Using LICENSE.REQuest		27
LICENSE Function and Commands		29
Display a License List		29
Request a License from TRACE32		30
Get License State		32
PRACTICE script example		33
Acknowledgments		34

Legacy Release History

Legacy Release History	(release_legacy.pdf)	1
General Information		5
Code		5
Release Information		5
Software Release from 01-Feb-2021		6
Software Release from 01-Sep-2020		9
Software Release from 01-Feb-2020		12
Software Release from 01-Sep-2019		14
Software Release from 01-Feb-2019		17
Software Release from 01-Sep-2018		20
Software Release from 01-Feb-2018		25
Software Release from 01-Sep-2017		28
Software Release from 01-Feb-2017		33
Software Release from 01-Sep-2016		37
Software Release from 01-Feb-2016		40
Software Release from 01-Sep-2015		42
Software Release from 02-Feb-2015		45
Software Release from 01-Sep-2014		48
Software Release from 16-Feb-2014		51
Software Release from 16-Aug-2013		54
Software Release from 16-Feb-2013		56
Software Release from 16-Aug-2012		59
Software Release from 08-Feb-2012		63
Software Release from 04-Jun-2011		66
Software Release from 10-Nov-2010		69
Software Release from 01-Apr-2010		71
Software Release from 01-Dec-2009		75
Software Release from 07-Oct-2008		83
Software Release from 12-Sep-2007		88
Software Release from 11-Jan-2007		93
Software Release from 15-Apr-2006		97
Software Release from 20-Aug-2005		104
Software Release from 12-Feb-2005		107
Software Release from 21-Jul-2004		119
Software Release from 10-Jan-2004		135
Software Release from 14-Jul-2003		148
Sorted by Commands		173

Software Updates

Software Updates	(updates.pdf)	1
History		3
About this Document		3
Do you have a valid Software License Key?		4
Temporary License Keys		4
TRACE32 Software Updater		5
Prerequisites		5
Supported Operating Systems		6
How to Start the TRACE32 Software Updater		6
Contacting Support		8
Manual Installation of Software Updates		9
Subdirectory Structure		9
Update of the Host Driver Software		10
USB Interface		10
Firmware Update		11
Serial Numbers		12

TRACE32 Index	(index.pdf)	1

TRACE32 Directory	(directory.pdf)	1