

Neal Stollon

On-Chip Instrumentation

Design and Debug for Systems on Chip

 Springer

Contents

1	Introduction	1
1.1	The Need for On-Chip Debug	1
1.2	Instrument- (**in-silicon) and EDA- (Presilicon) Based Verification	3
1.3	SoC Debug Requirements	7
1.4	Instrumentation-Based Debug Infrastructure	11
2	On-Chip Instrumentation Components	17
2.1	Trace and Event Triggering	17
2.2	External Interfaces for On-Chip Instrumentation	18
2.3	Performance Analysis Using On-Chip Instrumentation	19
2.4	On-Chip Logic and Bus Analysis	20
2.5	On-Chip Instrumentation Examples	22
2.5.1	Trace Monitoring and Interfaces	22
2.5.2	Bus Logic Monitoring	23
2.5.3	Real-Time Data Exchange	25
2.6	Multiprocessor Debug	26
3	JTAG Use in Debug	31
3.1	JTAG Pins	32
3.2	Test Access Port	35
3.3	JTAG Registers	38
3.4	JTAG Instructions	39
3.5	Boundary-Scan Description Language	40
3.6	The Road to JTAG: Historical Debug Approaches	44
3.6.1	Background Debug Mode	47
4	Processor System Debug	49
4.1	A Processor Debug Instrument Implementation	52
4.2	Processor Trace Compression	55
4.3	Hunting Code Errors with Self-Trace	59

5	An On-Chip Debug System	61
5.1	OCDS Features	62
5.1.1	Debug Events.....	64
5.1.2	Debug Event Actions.....	64
5.1.3	Debug Registers.....	65
5.2	Operation Modes.....	65
5.2.1	Entering Communication Mode	66
5.2.2	Communication Mode Instructions	66
5.2.3	Monitor-to-Debugger Host Data Transfer (Receive).....	67
5.2.4	Debugger Host-to-Monitor Data Transfer (Send)	67
5.2.5	High-Level Synchronization.....	67
5.3	OCDS Registers	68
5.3.1	Debug Task ID Register.....	68
5.3.2	Instruction Pointer Register	68
5.3.3	Hardware Trigger Comparison Registers	69
5.3.4	Considerations on Accessing OCDS Registers	69
5.4	OCDS JTAG Access	70
5.4.1	Steps to Initialize the JTAG Module	72
5.5	OCDS Module Access	72
5.5.1	Error Protection	72
5.6	OCDS JTAG I/O Instructions	74
5.7	OCDS JTAG Registers.....	76
5.8	Hardware Triggers	77
5.8.1	Structure of a Noninterruptible Monitor Routine.....	79
5.8.2	Structure of an Interruptible Monitor Routine.....	79
5.8.3	Debug Event Control Registers	80
5.9	Additional Features.....	81
5.9.1	System Security	82
5.9.2	Reset from the JTAG Side	83
5.9.3	Reset from the Chip/Processor Side	83
6	Bus System Debug.....	85
6.1	On-Chip Buses.....	85
6.2	Socket-Based SoC Design	87
6.2.1	SoC Interconnect Complexities.....	87
6.3	Bus-Level Integration.....	90
6.3.1	Bus Master Monitoring.....	91
6.3.2	Peripheral Bus Monitoring	91
6.3.3	Slave Monitoring	91
6.4	Internal and External Alternatives for Bus Trace.....	92
6.5	Programmable Bus Performance Monitoring	93
6.6	Bus Performance Monitoring.....	94
6.7	On-Chip and Off-Chip Analysis	98
6.8	Request Response Trace Bus Analysis	101
6.8.1	RRT Operations	103
6.8.2	RRT Implementation	104

7	Multiprocessor Debugging	109
7.1	Cross-Triggering and Global Breakpoint Control	110
7.2	HyperDebug Distributed Cross-Triggering	110
7.2.1	HyperDebug Controller	112
7.2.2	Typical HyperDebug Implementation	113
7.3	Multicore Synchronization Triggering and Global Actions.....	115
8	IEEE 1149.7: cJTAG/aJTAG.....	117
8.1	Test and Debug Views of 1149.7.....	118
8.2	Key T0–T5 Class Functions	120
8.3	MIPI Use of 1149.7.....	129
8.3.1	MIPI System Trace Module	130
8.4	Nexus Use of 1149.7	132
8.4.1	IEEE 1149.7/Nexus Integration.....	134
9	IEEE P1687 – IJTAG	137
9.1	Overlap Zones and Gateway Elements	139
9.2	Classes of P1687 Instruments	141
9.3	IEEE 1500 Instruments	143
10	OCP IP Debug Interfaces	145
10.1	OCP Multicore Debug	146
10.2	OCP Debug Features.....	148
10.3	Three Views of Debugging	150
10.3.1	Pure Software Debugging.....	150
10.3.2	Pure Hardware Debugging	151
10.3.3	System-on-Chip Debugging	151
10.4	Debug Components and IP Interfaces.....	151
10.5	Debug Socket Definitions	152
10.5.1	Core Debug Socket Interfaces	154
10.5.2	Cross-Triggering Socket Interfaces	157
10.5.3	OCP Synchronized Run Control	163
10.5.4	OCP Traffic-Monitoring and Trace Interfaces	163
10.5.5	Performance Monitoring	165
10.5.6	System Timestamping	166
10.5.7	Power Management Monitoring.....	166
10.5.8	Security Debug Interface	167
11	Nexus IEEE 5001.....	169
11.1	Nexus Implementation Classes	171
11.2	Nexus Message Architecture	172
11.2.1	Nexus TCODEs.....	174
11.2.2	Nexus Registers	178
11.3	NEXUS Interfaces.....	180
11.3.1	Nexus JTAG Access	180
11.3.2	NEXUS AUX Interfaces	181

11.4	Multicore Nexus Debug Approaches	185
11.4.1	Input Tool-to-Target Messages	187
11.4.2	Output Target-to-Tool Messages	188
11.5	Nexus Product Implementations	189
11.6	Summary	193
12	MIPS EJTAG	195
12.1	EJTAG Instructions and Registers	197
12.2	PC Sampling	199
12.3	MIPS PDtrace™	199
12.3.1	Trace Output Formats	200
12.3.2	Trace Control Block Registers.....	204
12.4	TCB Trigger Logic Overview	206
12.5	PDtrace External Interface	207
12.6	TCtrace IF	209
12.7	PDTRACE Operations	210
13	ARM ETM.....	213
13.1	ETM Signals	213
13.1.1	External Signals	214
13.2	ETM9 Registers	216
13.3	Trace Interface	218
14	Infineon Multicore Debug Solution	219
14.1	MCDS Trace Protocol Definition	221
14.1.1	Data Trace.....	223
14.2	Debug Transactor: RUN Control Bus Master.....	224
14.3	MCDS Run Control: On-Chip Debug Support.....	225
14.3.1	BCU Level 1 (Bus-Observer Unit on the System Bus)	227
14.3.2	Concurrent Debugging in Level 3 MCDS (Two-Channel Tracing).....	228
14.3.3	Debug Interface (Cerberus) (Debug Bus-Transactor Module).....	228
14.4	RW Mode and Communication Mode	228
14.5	Multicore Break Switch	229
15	EJTAG and Trace in Toshiba TX Cores	231
15.1	Processor Access Overview	232
15.2	Toshiba EJTAG Instructions and Registers.....	233
15.3	Debug Exceptions	235
15.4	Processor Debug Instructions and CP0 Registers	235
15.5	Break Functions	237
15.6	Output by PC Trace.....	238
Index.....		241