

# **Training Nexus Tracing**

Release 02.2024



TRACE32 Online Help	
TRACE32 Directory	
TRACE32 Index	
TRACE32 Training	þ
Training Nexus	Þ
Training Nexus Tracing	1
Basic Knowledge	8
NEXUS Characteristics	8
Limited Bandwidth	9
Branch Trace Messages (All NEXUS Classes)	9
Indirect Branch History Messages (All NEXUS Classes)	10
Data Trace Messages (NEXUS Class 3 only)	12
Ownership Trace Messages (All NEXUS Modules)	14
Watchpoint Trace Messages (All NEXUS Classes)	15
Data Acquisition Messages (IEEE-ISTO 5001-2008/2012 and NEXUS Class 3 only)	16
Multicore Tracing	18
AMP Tracing	18
SMP Tracing	19
Source for the Recorded Trace Information	20
NEXUS Configuration by TRACE32	22
Configuration of the Trace Interface	22
Parallel Interface	22
Serial Interface	28
Trace to Memory	30
Multicore Tracing	31
Configuration of the NEXUS Messages	33
Basic Messages	33
Additional Messages for IEEE-ISTO 5001-2008 and IEEE-ISTO 5001-2012	35
Add Timestamps to NEXUS Messages (MPC57xx/SPC57x only)	39
Multicore Tracing	43
NEXUS Trace Clients	44
Trace Client Types	44
Configuration	47
Target FIFO Overflow	48
Diagnosis	49
Stall Program Execution on Overflow Threat	50
Suppress Data Trace Messages on Overflow Threat	51

Further Countermeasures	52
FlowErrors	53
Displaying the Trace Content	54
Sources of Information for the Trace Display	54
Settings in the Trace Configuration Window	55
Recording Modes	55
States of the Trace	64
The AutoInit Command	65
Multicore Tracing	66
Basic Display Commands	67
Default Listing	67
Basic Formatting	69
Correlating the Trace Listing with the Source Listing	70
Browsing through the Trace Buffer	71
Display Items	72
Default Display Items	72
Further Display Items	76
Time Synchronization between TRACE32 Instances (AMP)	77
Setup	77
Utilization	78
Find a Specific Record	79
Belated Trace Analysis	80
Save the Trace Information to an ASCII File	81
Postprocessing with TRACE32 Instruction Set Simulator	82
Trace-based Debugging (CTS)	86
Re-Run the Program	86
Setup	86
Get Started	88
Forward and Backward Debugging	90
Re-Run the Program and Watch the Variables	91
Setup	91
Get Started	93
Forward and Backward Debugging	97
Details on HLL Instructions	98
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
Global TraceON/Trace OFF	121
ProgramTraceON/Trace OFF	126
DataTraceON/Trace OFF	130
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
Global TraceON/Trace OFF	163
ProgramTraceON/TraceOFF	168
DataTraceON/Trace OFF	173
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
Statistic Analysis of Task Switches	199
Statistic Analysis of OSEK Service Routines	201
Statistic Analysis of OSEK ISR2s	203
Statistic Analysis of Task-related OSEK ISR2s	204
Exporting all Types of Task Information and all Instructions (OTM)	205
Statistic Analysis of Interrupts	205
Statistic Analysis of Interrupts and Tasks	206
Statistic Analysis of Interrupts in Tasks	207
Exporting Task Information (Write Access)	208
Task Switches	208
OSEK Service Routines	211
OSEK ISR2s	214
Task-related OSEK ISR2s	217
Exporting Task Switches and all Instructions (Write Access)	220

Statistic Analysis of Interrupts	220
Statistic Analysis of Interrupts and Tasks	221
Statistic Analysis of Interrupts in Tasks	222
Belated Trace Analysis (OS)	223
OS-Aware Tracing - SMP Systems	224
Exporting all Types of Task Information (OTM)	224
Statistic Analysis of Task Switches	227
Statistic Analysis of OSEK Service Routines	229
Statistic Analysis of OSEK ISR2s	231
Statistic Analysis of Task-related OSEK ISR2s	233
Exporting all Types of Task Information and all Instructions (OTM)	235
Statistic Analysis of Interrupts	235
Statistic Analysis of Interrupts and Tasks	236
Statistic Analysis of Interrupts in Tasks	237
Exporting Task Information (Write Access)	238
Task Switches	238
OSEK Service Routines	243
OSEK ISR2s	248
Task-related OSEK ISR2s	253
Exporting Task Switches and all Instructions (Write Access)	257
Statistic Analysis of Interrupts	257
Statistic Analysis of Interrupts and Tasks	258
Statistic Analysis of Interrupts in Tasks	259
Belated Trace Analysis (OS)	260
Function Run-Times Analysis (Overview)	261
Software under Analysis (no OS or OS)	261
Flat vs. Nesting Analysis	261
Basic Knowledge about Flat Analysis	262
Basic Knowledge about Nesting Analysis	263
Summary	265
Function Run-Times Analysis - Single	266
Flat Analysis	266
Optimum NEXUS Configuration (No OS)	266
Optimum NEXUS Configuration (OS)	267
Function Time Chart	268
Nesting Analysis	276
Restrictions	276
Optimum NEXUS Configuration (No OS)	276
Optimum NEXUS Configuration (OS)	277
Items under Analysis	278
Numerical Nested Function Run-time Analysis for all Software	281
Additional Statistics Items for OS	289

Timing Improvements for OS	292
Problems and Workarounds for OS	293
More Nesting Analysis Commands	297
Third-party Timing Tools	305
Function Run-Times Analysis - SMP Instance	306
Flat Analysis	306
Optimum NEXUS Configuration (No OS)	306
Optimum NEXUS Configuration (OS)	307
Function Timing Diagram	308
Function Timing Diagram (Including Task Information)	310
Nesting Analysis	316
Restrictions	316
Optimum NEXUS Configuration (OS)	317
Numerical Nested Function Run-time Analysis for all Software	318
Timing Improvements for OS	326
More Nesting Analysis Commands	327
Third-party Timing Tools	334
Structure the Trace Evaluation	335
GROUP Creation	335
Working with GROUPs	339
GROUP Status ENable	339
GROUP Status ENable + Merge	341
GROUP Status ENable + HIDE	342
Trace-based Code Coverage	343

Version 04-Mar-2024

## **NEXUS Characteristics**

NEXUS is a message-based trace protocol. A NEXUS hardware module generates the trace messages. Trace messages can be generated for activities of core(s), eTPU(s), GTM(s), for activities of DMA controller(s), of FlexRay controller(s), of SRAM port sniffers and other units. The **Source Processor Identifier** in the NEXUS messages identifies the trace source.

NEXUS hardware modules are available in two versions:

- NEXUS Class 2 + Modules provide the visibility of the instruction flow and task switches.
- **NEXUS Class 3 + Modules** provide the visibility of the instruction flow, load/store operations, task switches and trace information generated by code instrumentation.

Trace messages generated by a NEXUS module:

- can be exported off-chip via a parallel trace interface.
- can be exported off-chip via a serial (Aurora) trace interface.
- can be stored to an on-chip trace memory (trace to memory).

NEXUS hardware modules are compliant to one of the following standards:

• IEEE-ISTO 5001™-2012

Serial (Aurora) trace interfaces are always compliant to this standard.

- IEEE-ISTO 5001<sup>™</sup>-2008
- IEEE-ISTO 5001™-2003

#### Before you continue with this training, refer to your processor manual and check:

- Which class is supported by your NEXUS module?
- Are trace messages exported off-chip via a parallel or serial trace interface?
- Are trace messages stored to an on-chip trace memory?
- Which NEXUS standard is supported by your NEXUS module?

Regardless of the implementation of your NEXUS module (off-chip export or on-chip trace memory) it may happen while testing that more trace messages are generated than the trace interface/memory interface can convey. This may disturb your tests.

For a better understanding of this issue and its counter-measures, a short introduction into the NEXUS protocol is given. The following example configuration is used: MPC5775K with parallel trace interface consisting of 16 pins (MDO) for the export of NEXUS messages. The term *trace beat* is used for the trace information that is transferred per trace clock.

## Branch Trace Messages (All NEXUS Classes)

Branch trace messages provide a standard protocol for instruction flow visibility.

## **Direct Branch Messages**

TCODE number = 3 (6 bits)Source processor identifier<br/>(4 bits)Number of sequential<br/>instructions executed<br/>since the last taken<br/>branch (1 to 8 bits)Timestamp (optional)<br/>(0 to 30 bits)

11 to 48 bits in 1 to 4 trace beats

## Indirect Branch Messages

**TCODE number = 4** (6 bits)

Source processor identifier (4 bits)

Address space indicator (1 bit)

Number of sequential instructions executed since the last taken branch (1 to 8 bits)

Branch destination address (1 to 32 bits)

**Timestamp (optional)** (0 to 30 bits)

13 to 81 bits in 1 to 6 trace beats Indirect Branch History Messages can be used to save bandwidth, since only indirect branches cause messages. Information on direct branches is stored in the Direct Branch History.

Indirect Branch History Messages are recommended for:

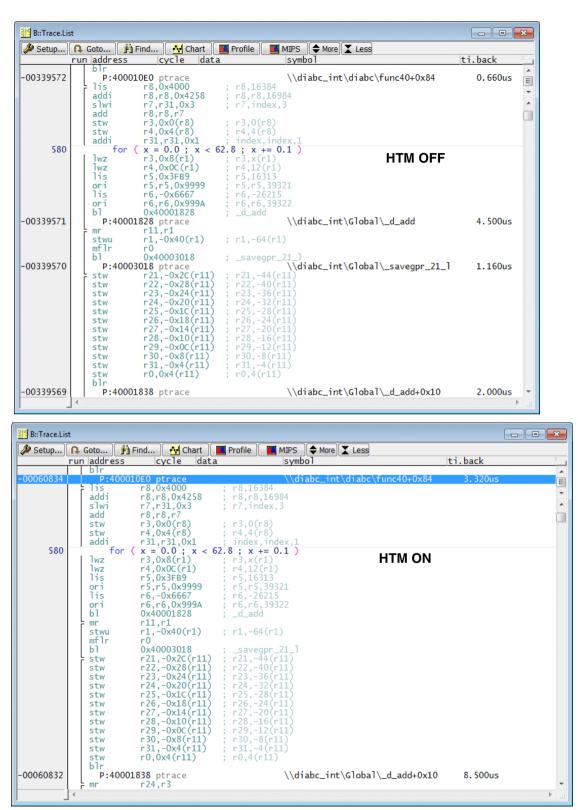
- small trace ports if they have bandwidth problems
- long instruction flow traces
- TRACE32 Trace Mode STREAM
- multi-source traces

### **Indirect Branch History Messages**

TCODE number = 28 (6 bits)
Source processor identifier (4 bits)
Address space indicator (1 bit)
Number of sequential instructions executed since the last taken branch (1 to 8 bits)
Branch destination address (1 to 32 bits)
Direct branch history (1 to 32 bits)
<b>Timestamp (optional)</b> (0 to 30 bits)

### 14 to 113 bits in 1 to 8 trace beats

The caveat of the use of Indirect Branch History Messages is a less accurate timestamp, since less NEXUS messages are generated and timestamped.



Data trace messages are used to export information on the load/store operations.

### Data write messages

TCODE number = 5 (6 bits)
Source processor identifier (4 bits)
Address space Indicator (1 bit)
Data size (4 bits)
Data write address (1 to 32 bits)
Data write value (1 to 64 bits)
<b>Timestamp (optional)</b> (0 to 30 bits)

17 to 141 bits in 2 to 9 trace beats

#### **Data read messages**

**TCODE number = 6** (6 bits)

Source processor identifier (4 bits)

Address space Indicator (1 bits)

Data size (4 bits)

Data read address (1 to 32 bits)

Data read value (1 to 64 bits)

Timestamp (optional) (0 to 30 bits)

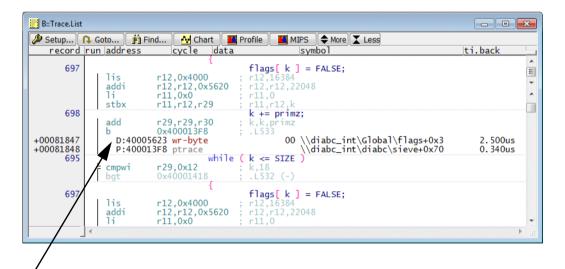
**17 to 141 bits** in 2 to 9 trace beats

Exporting information on load/store operations may easily generate more trace messages than the interface in use can convey. This is most likely to occur when several data accesses are carried out in quick succession.

If information on all load/store operations is exported, each data access can be correlated to its instruction (data cycle assignment).

B::Trace.List					
🌽 Setup 🚺	🕽 Goto 🏼 🏥 F	ind	Nofile	MIPS 🔶 More 🗶 Less	
record	run address	cycle dat	a	symbol	ti.back
+00187894	P:4000	13AC ptrace		\\diabc_int\diabc\sieve+0x24	0.320us
	- cmpwi	r31,0x12	; i,18		E
	bgt	0x400013CC	; .L527		
	lís		; r12,16		•
	addi	r12,r12,0x5620		2,22048	<b>^</b>
	1i	r11,0x1	; r11,1		
	stbx	r11,r12,r31	; r11,r1		
+00187895		5623 wr-byte		01 \\diabc_int\Global\flags+0x3	0.500us
	addi	r31,r31,0x1	; i,i,1		
	b	0x400013AC	; .L529		
+00187896		<b>13AC</b> ptrace		\\diabc_int\diabc\sieve+0x24	0.340us
	cmpw1	r31,0x12	; i,18		
	bgt	0x400013CC	; .L527		
	lís	r12,0x4000	; r12,16		
	addi	r12,r12,0x5620		.2,22048	
	1i	r11,0x1	; r11,1		
	stbx	r11,r12,r31	; r11,r1		0.500
+00187897		5624 wr-byte		01 \\diabc_int\Global\flags+0x4	0.500us
	addi	r31,r31,0x1	; i,i,1		*
	•				E. ₹

If a trace filter is used to export only some load/store operations, the correlation to the instruction is not always possible.



It was not possible to correlate the load/store operation to its instruction. For this reason the data access cycle is printed in red and is displayed preceding the next Branch Trace Message.

Ownership trace messages are trace messages that are generated when a write access to the Process ID register PID0 (8 bit) occurs.

Ownership trace messages can be used to export OS-related information e.g. task switch information for NEXUS Class 2 Modules.

### **Ownership Trace Message**

TCODE number = 2 (6 bits)

Source processor identifier (4 bits)

Task/Process ID tag (32 bits)

> 42 bits in 3 trace beats

## Alternative for IEEE-ISTO 5001<sup>™</sup>-2012

Since 8 bits are often not sufficient to encode OS-related information, the 32-bit NEXUS PID Register (NPIDR) can be used as an alternative. Ownership Trace Messages have also a slightly different format for IEEE-ISTO 5001<sup>TM</sup>-2012.

## **Ownership Trace Message**

**TCODE number = 2** (6 bits)

Source processor identifier (4 bits)

Task/Process ID tag (1 to 32 bits)

Timestamp (optional) (0 to 30 bits)

11 to 72 bits in 1 to 5 trace beats The Ownership Trace Messages can not clearly be assigned to an instruction. Similar to the filtered Data Trace Messages they are printed in red and displayed preceding the next Branch Trace Message.

B::Trace.List	NEXUS List.TASK DEFault					
	📭 Goto 🦻 Find 🥂 Chart 📲 Profile 📲 MIPS 🔷 More 🗶 Les					
record	nexus	run address	cycle data	symbol	ti.back	
		rlwinm		x17; r8,r7,resetvector,18,23		
		or	r10,r0,r8			E
		extrwi	r6,r12,0x6,0x10	; r6,r12,6,16		-
		or	r3,r10,r6	and the second		
000000000		b]	0x2655C	; osSetRegisterPID24		<u></u>
-02662152	TCODE=03 SRC=0 PT-DBM ICNT=000F	F:0002	655C ptrace	Global\osSetRegisterPID24	1.720us	
		mtpid	r3			
		nop	-3 -3 08			
		srawi mtpid	r3,r3,0x8 r3	; r3,r3,8		
			1.2			
		nop srawi	r3,r3,0x8	; r3,r3,8		
		mtpid	r3	, 10,10,0		
		blr	19			
	task: bTask1 (000000C)	1.80				
-02662149	TCODE=02 SRC=0 OTM PROCESS=000000F0		owner 00000	00C	1.860us	
-02662148	TCODE=04 SRC=0 PT-IBM MAP=0 ICNT=0008 U-ADDR=000001DC	F:0002	612C ptrace	TimerInterrupt_cat2c+0xD4		-
	Construction of the second se second second sec					· + //
اہ (						

## Watchpoint Trace Messages (All NEXUS Classes)

## The Onchip breakpoints of the MPC5xxx/SPC5xxx can be used:

- to stop the program execution at a specific event.
- to generate a pulse on EVTO at a specific event.
  - Not available for AMP systems if synchronous break is activated.
  - Not available for SMP systems.
- to export Watchpoint Hit Messages.

## Data Acquisition Messages (IEEE-ISTO 5001-2008/2012 and NEXUS Class 3 only)

Data Acquisition Messaging (DQM) allows code to be instrumented to export customized trace information.

Data Acquisition Messages are trace messages that are generated when a write access to the Debug Data Acquisition Message register DDAM (32 bit) occurs. DQTAG (8 bit) is sampled from the DEVENT register when a write to DDAM is performed.

The DQTAG field can be used to attribute the information written to DDAM. E.g. the DQTAG field can be interpreted by the trace tool as a channel ID.

#### Data acquisition message

**TCODE number = 7** (6 bits)

- Source processor identifier (4 bits)
  - Identification tag from DQTAG (8 bits)

Data from DDAM (1 to 32 bits)

**Timestamp (optional)** (0 to 30 bits)

19 to 80 bits in 2 to 6 trace beats The command group **DQMTrace** is used to display and analyze the Data Acquisition Messages.

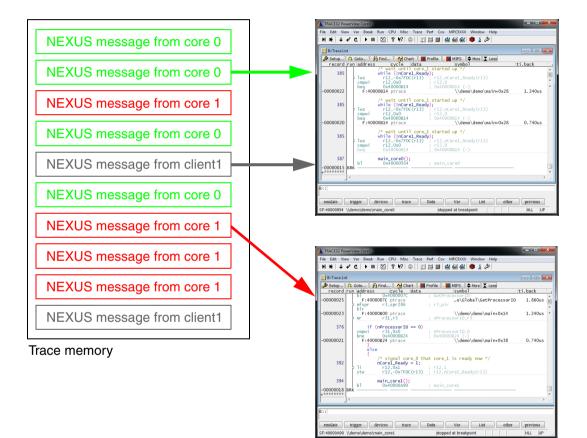
DQMTrace.List

B::DQM	Trace	List							×
🔑 Setup	<b>D</b>	Goto]	Find	Chart	Profile	MIPS	And the More	Less	
record	run	addre	ss	cycle	data		symbol	ti.back	
+148007	1	D:	00000001	ddam-wr		00A81D83		3.183us	
+148010	0	D:	00000001	ddam-wr		00A85148		3.102us	=
+148013	1	D:	00000001	ddam-wr		00A81D84		3.006us	
+148016	0	D:	00000001	ddam-wr		00A85149		3.003us	*
+148019	0	D:	00000001	ddam-wr		00A8514A		0.368us	*
+148022	0	D:	00000001	ddam-wr		00A8514B		0.307us	
+148025	1	D:	0000001	ddam-wr		00A81D85		3.145us	-
									▶

column layout					
address	Identification tag				
cycle	Always "write access to DDAM"				
data	Exported data				
ti.back	Timestamp				

## AMP Tracing

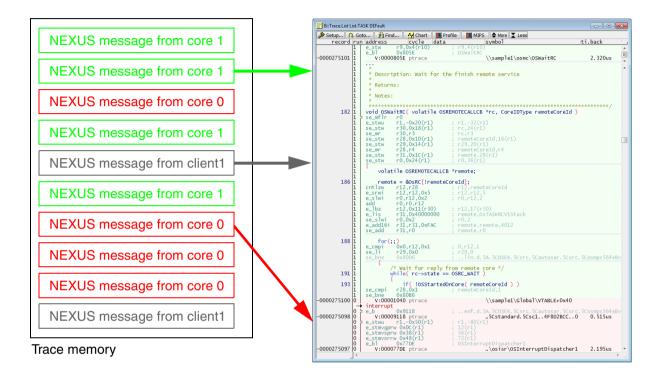
Trace Display	Each TRACE32 instance analyzes and displays the trace information generated by the core(s) it controls.				
	Trace messages generated by clients (DMA, FlexRay etc.) are assigned to the TRACE32 instance that enabled the messaging.				



## **SMP Tracing**

**Trace Display** 

Trace information from all trace sources in the SMP system is displayed together.



If TRACE32 is started when a PowerTrace hardware and a NEXUS ADAPTER / PREPROCESSOR SERIAL is connected, the source for the trace information is the so-called **Analyzer** (Trace.METHOD Analyzer).

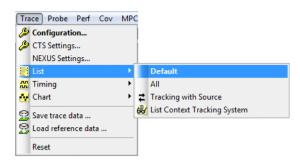
Trace Probe Perf Cov MPC				
💋 Configuration				
CTS Settings NEXUS Settings List				
x Timing				
Chart •	B::Trace			
😰 Save trace data	- METHOD -			
🔀 Load reference data		Analyzer 🔘 Onchip		ER 🔘 SNOOPer 🔘 FDX 🛛 🔍 LA
Reset	Analyzer O CA	analyzer © Onenip	O AKI O LOGG	Integrator      Probe     Probe     Probe
	– state –––––	used	ACCESS	TDelay
	O DISable		auto 👻	0. TrOnchip
	OFF	0.		0% - 🔑 NEXUS
	© Arm	- SIZE	CLOCK	
	🔘 trigger	134217728.		THreshold
	🔘 break			1.25 ▼ ¥ advanced
	SPY	Mode	Mode	© VCC
		Fifo	SLAVE	© CLOCK
	commands	Stack		autofocus
	RESet	🔘 Leash		
	O Init	STREAM		TERMination
	SnapShot	© PIPE		
	List	© RTS		TestFocus
	AutoArm	L		XX AutoFocus
	AutoInit			XX ShowFocus
	SelfArm			

The setting Trace.METHOD Analyzer has the following impacts:

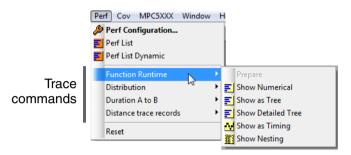
#### 1. **Trace** is an alias for **Analyzer**.

Trace.List	; Trace.List means ; Analyzer.List
Trace.Mode Fifo	; Trace.Mode Fifo means ; Analyzer.Mode Fifo

2. All commands from the Trace menu apply to the Analyzer.



3. All Trace commands from the Perf menu apply to Analyzer.



4. TRACE32 is advised to use the trace information recorded to the Analyzer as source for the trace evaluations of the following command groups:

CTS. <sub_cmd></sub_cmd>	Trace-based debugging
COVerage. <sub_cmd></sub_cmd>	Trace-based code coverage
ISTAT. <sub_cmd></sub_cmd>	Detailed instruction analysis
MIPS. <sub_cmd></sub_cmd>	MIPS analysis
BMC. <sub_cmd></sub_cmd>	Synthesize instruction flow with recorded benchmark counter information

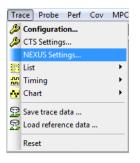
This NEXUS Training uses always the command group **Trace**. If your trace information is stored to an **on-chip trace memory**, just select the trace method Onchip and nearly all features will work as demonstrated for the trace method Analyzer

Trace.METHOD Onchip

# **Configuration of the Trace Interface**

## **Parallel Interface**

The interface configuration is done via the NEXUS window. The TRACE32 NEXUS window has a different look for IEEE-ISTO 5001<sup>TM</sup>-2003, IEEE-ISTO 5001<sup>TM</sup>-2008 and IEEE-ISTO 5001<sup>TM</sup>-2012.



## NEXUS window for IEEE-ISTO 5001 $^{\rm TM}\mbox{-}2003$

B::NEXUS				
nexus OFF OFF ON RESet Trace List	Selection BTM HTM OTM WTM DTM OFF	option	Configuration PortSize MD012 ▼ PortMode 1/2 ▼ DDR PCRCONFIG	CLIENT1 SELECT NONE MODE OFF CLIENT2 SELECT NONE MODE OFF

## NEXUS window for IEEE-ISTO 5001<sup>TM</sup>-2008

nexus     selection     option     configuration     CLIENT1       OFF     Ø BTM     POTD     PortSize     SELECT       Ø ON     HTM     STALL     MD012     NONE	B::NEXUS				
OTM       OFF       PortMode       MODE         WTM       Suppression       1/2       OFF       CLIENT2         DQM       Suppression       DDR       CLIENT2       SELECT         OFF       SpenDQM       SpenPTM       NONE       MODE         PTCM       SpenDTM       OFF       MODE       OFF         BL_HTM       SpenOTM       OFF       MODE       OFF         TLBNEW       SupprHReshold       1/4       V       V	<ul> <li>○ OFF</li> <li>● ON</li> <li>RESet</li> <li> <i>P</i> Trace     </li> </ul>	BTM HTM OTM OTM DQM DQM DTM OFF PTCM PID_MSR BL_HTM TLBNEW	POTD STALL OFF Suppression SpenDQM SpenWTM SpenPTM SpenDTM SpenOTM SupprTHReshold	PortSize MDO12 V PortMode	SELECT NONE * OFF * CLIENT2 - SELECT NONE * MODE

## NEXUS window for IEEE-ISTO $5001^{\text{TM}}$ -2012

B::NEXUS				- • •
PERECOS	Selection	option POTD PTMARK DTMARK STALL OFF Suppression SpenDQM SpenVTM SpenPTM SpenOTM SpenOTM SupprTHReshold 1/4	Configuration - PortSize MDO12 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE MODE OFF CLIENT2 SELECT NONE CLIENT3 SELECT NONE MODE OFF MODE OFF MODE

NEXUS.state

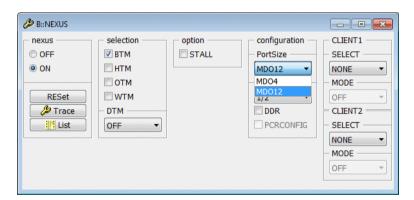
### ; display NEXUS window

The configuration for a parallel NEXUS interface is identical for all compliant standards. This is why only the simpler IEEE-ISTO 5001<sup>TM</sup>-2003 is shown in the configuration examples.

### Select NEXUS Port Size

Selecting the NEXUS port size is only possible if SYStem Mode Down is selected.

B::SYStem				- • •
Mode © Down NoDebug Prepare Go Attach ChendPu	MemAccess — CPU NEXUS Denied CpuAccess — Enable	Option IMASKASM IMASKHLL ICFLUSH ICREAD V DCREAD	Option MMUSPACES DUALPORT WATCHDOG LPMDebug	Option.DisMode Option.DisMode AUTO ACCESS FLE VLE
© StandBy ○ Up (StandBy) ◎ Up	Denied Nonstop	V FREEZE	ResetDetection	CONFIG
CPU MPC5554	BdmClock 4.0MHz	Vectors —		NEXUS



NEXUS port size	
MOD2   MOD4   MDO8	Specify MDO[f:0]
MDO12   MDO16	(number of Message Data Out pins available)

NEXUS.PortSize MD012

; specify a trace port width of ; 12 MDOs

The PortMode determines the frequency of MCKO (Message ClocK Out) relative to the system clock (SYS\_CLK). Max. MCKO is usually 80 MHz, please refer to the **Nexus characteristics** in the data sheet of your chip for details.

B::NEXUS				- • •
nexus OFF OFF ON RESet Trace List	Selection BTM HTM OTM DTM OFF	option	configuration       PortSize       MD012 ▼       PortMode       1/2 ▼       1/8       1/4       1/3       1/2       1/1	CLIENT1 SELECT MODE OFF CLIENT2 SELECT NONE MODE OFF

NEXUS.PortMode 1/4

To measure the MCKO frequency with TRACE32 proceed as follows:



12.00	0480	MH	z MCKO	(*160)
ode	Gate		Select	
Frequency	© 0.01 s	Init	PODBUS	
Period	◎ 0.1 s	🔲 AutoInit	© IAC1	
PulsLow	© 1 s		© IAC2	
) PulsHigh	© 10 s	out	© IAC3	
EventLow	endless	TUO 🔲	© IAC4	
EventHigh	variable		© DAC1	
EventHOld		]] []	© DAC2	
		PROfile	O DCNT1	
			O DCNT2	
			MCKO	
			C DMA1	
			C DMA2	
			© FRAY1	
			© FRAY2	

B::NEXUS				
nexus OFF ON RESet Trace	selection BTM HTM OTM WTM DTM OFF	option	configuration       PortSize       MD012       PortMode       1/2       ♥ DDR       PCRCONFIG	CLIENT1 SELECT MODE OFF CLIENT2 SELECT MODE OFF

Advise the NEXUS module to export trace information on the rising and falling edge of MCKO (not supported by all chips/cores).

NEXUS.DDR ON

Chips with serial interface provide a Nexus module compliant to the IEEE-ISTO 5001<sup>TM</sup>-2012 standard.

## Select NEXUS PortSize

Selecting the NEXUS PortSize is only possible if **SYStem Mode Down** is selected.

B::SYStem				- • ×
<ul> <li>Mode</li> <li>Down</li> <li>○ NoDebug</li> </ul>	MemAccess     O     CPU     O     NEXUS	Option IMASKASM	Option MMUSPACES	Option.DisMode AUTO ACCESS
© Prepare © Go © Attach © StandBy Up (StandBy) © Up	<ul> <li>Denied</li> <li>CpuAccess</li> <li>Enable</li> <li>Denied</li> <li>Nonstop</li> </ul>	CFLUSH ICREAD CREAD FREEZE NOTRAP	- WATCHDOG OFF  - LPMDebug - ResetDetection OFF  - V	© FLE © VLE
CPU MPC5746M	BdmClock 4.0MHz V	Vectors		NEXUS

nexus       selection       option       configuration       CLIENT1         OFF       ØTM       POTD       PortSize       SELECT         ØON       HTM       PTMARK       Hane       MODE         TimeStamps       DQM       STALL       OFF       MODE         OTM       OFF       Stall       OFF       Stall         OFF       OFF       Suppression       Stale       OFF         Ista       OFF       SpenDQM       OFF       NONE         PTCM       SpenDPM       SpenDTM       Stale       OFF         SupprTHReshold       SupprTHReshold       MODE       Stale	B::NEXUS				
1/4 • OFF •	<ul> <li>OFF</li> <li>ON</li> <li>TimeStamps</li> <li>RESet</li> <li>Trace</li> </ul>	BTM HTM UVTM DQM OTM OFF T DTM OFF F PTCM PID_MSR	POTD PTMARK DTMARK STALL OFF Suppression SpenDQM SpenWTM SpenPTM SpenDTM SpenOTM SupprTHReshold	PortSize 4Lane 1Lane 2Lane 3Lane 4Lane 5Lane	SELECT NONE OFF CLIENT2 SELECT NONE OFF CLIENT3 SELECT NONE MODE OFF MODE MODE

NEXUS port size	
2Lane   4Lane	Specify number of (Aurora) lanes

NEXUS.PortSize 2Lane

; specify a trace port with 2 lanes

Set the bit clock according to the processor's data sheet.

B::NEXUS				
nexus	selection	option	- configuration	CLIENT1
OFF	BTM	POTD	- PortSize	- SELECT
ON	HTM	PTMARK	4Lane 🔻	NONE -
	WTM	DTMARK	- PortMode	- MODE
TimeStamps	DQM	- STALL	1250Mbps -	OFF 👻
	- OTM	OFF -	625Mbps	CLIENT2
RESet	OFF -		_ 750Mbps	- SELECT
Trace	- DTM	_ suppression	850Mbps 1000Mbps	NONE -
List	OFF 🔻	SpenDQM	1250Mbps	- MODE
	- PTCM	SpenWTM	1500Mbps	OFF -
	PID_MSR	SpenPTM	1700Mbps 2000Mbps	CLIENT3
	BL HTM	SpenDTM	2500Mbps	- SELECT
		SpenOTM	3000Mbps 3125Mbps	NONE -
		- SupprTHReshold -	3400Mbps	MODE
		1/4 -	4000Mbps	OFF -
		<u> </u>	4250Mbps 5000Mbps	
			6000Mbps -	
			6250Mbps	

NEXUS.PortMode 1250Mbps

Automotive processors usually need an external reference clock for Aurora operation. Lauterbach's PREPROCESSOR SERIAL can provide this clock signal. It is enabled using **NEXUS.RefClock ON**.

B::NEXUS				
nexus	selection	option	configuration —	CLIENT1
OFF	BTM	POTD	- PortSize	- SELECT
ON	I HTM	PTMARK	4Lane 🔻	NONE -
	WTM	DTMARK	- PortMode	MODE
TimeStamps	DQM	- STALL	1250Mbps -	OFF -
	- OTM	OFF -	RefClock	CLIENT2
RESet	OFF -			- SELECT
🥬 Trace	- DTM	- suppression		NONE -
List	OFF 🔻	SpenDQM		MODE
	- PTCM	SpenWTM		OFF -
	PID_MSR	SpenPTM		CLIENT3
	BL_HTM	SpenDTM		- SELECT
		SpenOTM		NONE -
		- SupprTHReshold		MODE
		1/4 👻		OFF -

The usage of the onchip trace memory requires that trace memory is allocated.

```
; allocate trace memory for 4K NEXUS packets (packet size = 32 bit)
; A: stands for physical memory
Onchip.TBARange A:0xD000000--0xD003fff
```

B::Onchip		
state	used	ACCESS TDelay
OISable		auto 🔻 📑 TrOnchip
OFF	0.	- NEXUS
O Arm	- SIZE	CLOCK BMC
○ TRIGGER	4096.	
O break		
L	Mode	- TBARange
commands	Fifo	A:0xD0000000xD003FFF
RESet	Stack	
🛛 🛛 Init	Ceash	
SnapShot		
List	SLAVE	
AutoArm	FlowTrace	
AutoInit		
SelfArm		

Emulation devices may provide more trace memory.

; allocate trace memory for 32K NEXUS packets (packet size = 32 bit) ; EEC: stands for emulation device memory

Onchip.TBARange EEC:0xC000000--0xC01FFFF

🔑 B::Onchip		
state DISable OFF Arm TRIGGER break	0. SIZE 32768.	ACCESS TDelay
commands RESet SnapShot List AutoArm AutoInit SelfArm	Mode Fifo Stack Leash SLAVE FlowTrace	TBARange EEC:0xC000000-0xC01FFFF

Trace information generated for multiple cores is:

- Exported via a single off-chip trace interface.
- Stored in a joint on-chip trace memory.

#### SMP Systems

Due to the fact that one TRACE32 instance is used to control multiple cores in an **SMP system** there is only one NEXUS configuration window, and thus no problem to keep the Nexus interface setting consistent.

Since trace messaging from more than one core may easily generate more trace messages than the interface in use can convey, it is possible to enable the message generation only for the cores that are in the focus of the analysis.

	CoreENable 1 ores: 0, 1	
[ok]	<core></core>	
SV:000024A0 \\sample1\osset\StartOS+0x23C		

Enable core tracing for listed logical cores.

The situation is different for multiple cores in an **AMP system**. Here each core is controlled by its own TRACE32 instance, each with its own NEXUS configuration window. Since the TRACE32 Resource Management does not keep the Nexus interface settings in multiple TRACE32 instances consistent, this is the job of the user.

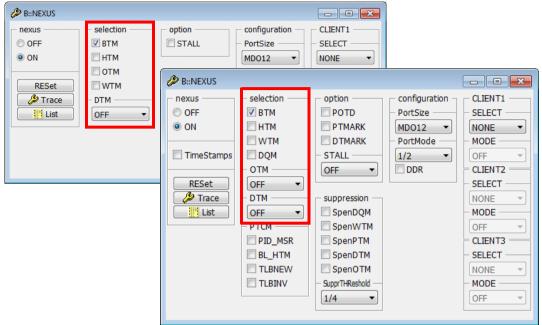
Since trace messaging from more than one core may easily generate more trace messages than the interface in use can convey, it is recommended to disable the message generation for core(s) that are not in the focus of the analysis.

nexus 🌮				
B::NEXUS	Selection BTM HTM OTM WTM DQM DTM	option POTD STALL OFF suppression SpenDQM	Configuration PortSize MDO12 V PortMode 1/2 V DDR	CLIENT1 SELECT NONE MODE OFF CLIENT2 SELECT
	OFF  PTCM PID_MSR BL_HTM TLBNEW TLBNV	SpenWTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4		NONE * MODE

NEXUS.OFF

## **Basic Messages**

NEXUS window for IEEE-ISTO 5001  $^{\rm TM}\mbox{-}2003$ 



NEXUS window for IEEE-ISTO 5001<sup>TM</sup>-2012

Messages			
BTM ON	Enable Branch Trace Messages.		
BTM ON + HTM ON	Enable Indirect Branch History Messages.		
<b>OTM ON</b> (2003/2008 Standard)	Enable Ownership Trace Messages via 8-bit PID0.		
OTM PID0 (2012 Standard)	Enable Ownership Trace Messages via 8-bit PID0.		
OTM NPIDR (2012 Standard)	Enable Ownership Trace Messages via 32-bit NPIDR.		
WTM ON	Enable Watchpoint Hit Messages.		
	Watchpoint Hit Messages are usually not used by the user. TRACE32 enables them automatically, if they are needed.		

DTM Read DTM Write DTM ReadWrite DTM IFETCH	Enable Data Read Messages. Enable Data Write Messages. Enable Data Read and Write Messages. Instruction fetches are exported as Data Read Messages.
	The basic idea of the <b>Limited</b> settings is to exclude stack read/writes from the message generation and thus avoid bandwidth problems.
DTM ReadLimited (2012 Standard)	Enable Data Read Messages, but exclude read accesses using GPR R1 in effective address computations.
DTM WriteLimited (2012 Standard)	Enable Data Write Messages, but exclude write accesses using GPR R1 in effective address computations.
DTM ReadWriteLimited (2012 Standard)	Enable Data Read and Write Messages, but exclude read/write accesses using GPR R1 in effective address computations.

NEXUS.BTM ON	; enable Branch Trace Messages
NEXUS.DTM ReadWrite	; enable Data Trace Messages for ; both read and write operations
NEXUS.OTM NPIDR	; enable Ownership Trace Messages ; via NPIDR register

# Additional Messages for IEEE-ISTO 5001-2008 and IEEE-ISTO 5001-2012

B::NEXUS				
nexus OFF OFF ON RESet Trace List	Selection BTM HTM OTM DQM DTM OFF PTCM PID_MSR BL_HTM TLBNEW TLBINV	option POTD STALL OFF SpenDQM SpenWTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4	Configuration PortSize MDO12 * PortMode 1/2 * DDR	CLIENT1 SELECT NONE * OFF * CLIENT2 SELECT NONE * MODE OFF *

## **Data Acquisition Messages**

Messages	
DQM ON	Enable Data Acquisition Messages.

When a write to the PID or NPIDR register occurs, a Program Trace Correlation Message can be generated instead of an Ownership Trace Message.

The Program Trace Correlation Message contains the address of the instruction that wrote the OS-related information to the PID or NPIDR register and the OS-related information itself.

This has the following advantages:

- **Trace.List:** the OS-related information can be directly assigned to instruction that wrote to the PID or NPIDR register.
- Trace.STATistic.Func: the accuracy of all task-aware function run-time measurements is improved.

OTM ON PID_MSR ON (2008 Standard) Or	Enable Program Trace Correlation Messages for Ownership tracing.	
OTM PID0 PID_MSR ON (2012 Standard) Or		
OTM NPIDR PID_MSR ON (2012 Standard)		
POTD	Periodic Ownership Trace message Disable. OFF: Periodic Ownership Trace Message is enabled (default). ON: Periodic Ownership Trace Message is disabled. Recommended if PID0 register is used.	

B::Trace.List List.TASK NEXUS DEFault					
🎾 Setup । 🖓 Goto । 👘 Find । 👬 Chart 🛛 🌉 Profile । 🚺 MIPS । 🜲 More 🛛 🗶 Less					
record nexus	run address cycle	data	symbol	ti.back	
0000013790 TCODE=03 SRC=0 PT-DBM ICNT=0002	V:400005BE ptrace		\OSSetPID0+0x6	0.750us	_
task: TASKO (0000009)	mtpid r3	; value			(
0000013789 TCODE=21 SRC=0 PT-PTCM EVCODE=0005 CDF=0002 ICNT=0001 HIST=00000	00 V:400005C2 owner	0000009	OSSetPID0+0x0A	1.250us	
	msync se_isync				
0000013787 TCODE=03 SRC=0 PT-DBM ICNT=0002	V:400005C8 ptrace		OSSetPID0+0x10	1.250us	
					•

NEXUS.OTM NPIDR

NEXUS.PID\_MSR ON

NEXUS.POTD ON

A Program Trace Correlation Message is generated when a direct branch function call (bl/bcl/bla/bcla) occurred while Indirect Branch History messaging is used.

BL_HTM	Program Trace Correlation Message is generated on a direct branch function call (for NEXUS.HTM ON only).

NEXUS.BTM ON

NEXUS.HTM ON

	📭 Goto ) 🎒 Find ) 🛃 Chart 🛛 📕 Profile 🖉 🎽 MIPS 🕽 🖨 Mo				
ecord r 474	lexus	run address cmpwi ble	cycle data r29,0x0 0x40000CD0	a  symbol ; e.0 ; .L343 (-) c13(b, f, e-1);	ti.back
	No NEXUS message	subi mr mr bl	r5,r29,0x1 r3,r28 r4,r26 0x40000C88	; r5,e1 ; r3,b ; r4,f ; func13	
	for function call	int func	13(a,c,e)	/* arguments and	locals stack-tracking */
466		,	1 4		
466		fit a stwu mflr	r1,-0x20(r1) r0	; r1,-32(r1)	
466		{ stwu stmw stw mr	r1,-0x20(r1) r0 r26,0x8(r1) r0,0x24(r1) r31,r3	; f,8(r1) ; r0,36(r1) ; a,r3	
466		{ stwu stmv stmw mr mr mr	r1,-0x20(r1) r0 r26,0x8(r1) r0,0x24(r1)	; f,8(r1) ; r0,36(r1)	

#### NEXUS.PTCM BL\_HTM ON

	📭 Goto 🥻 👬 Find 🥂 Chart 🛛 🌉 Profile 🛛 🗮 MIPS 🛛 🖨 More 🗶 Less				
record 470	nexus r	un address	f = b+a;	ata symbol	ti.back
		add	r26,r28,r31	; f,b,a	
471			<pre>d = f*b; r27,r26,r28</pre>	; d,f,b	
473		cmpwi ble	if ( e > 0 ) r29,0x0 0x40000CD0	; e,0 ; .L343 (-)	
474	NEXUS message for function call		c += fu r5,r29,0x1 r3,r28	unc13( b, f, e-1 ); ; r5,e,1 ; r3,b ; r4,f	
0825982	TCODE=21 SRC=0 PT-PTCM EVCODE=000A CDF=0001 ICNT=0004 HIST=	F:4000	r4,r26 DOC88 ptrace	; r4,f \\diabc\diabc\func13	2.580us
100		int func int a, c	:13 <mark>(</mark> a,c,e) :,e;	/* arguments and loc	als stack-tracking *,
466		1 stwu	r1,-0x20(r1)	; r1,-32(r1)	
		mflr stmw stw mr mr	r0 r26,0x8(r1) r0,0x24(r1) r31,r3 r30,r4	; f,8(r1) ; r0,36(r1) ; a,r3 ; c,r4	

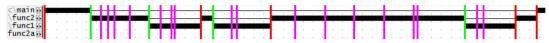
Enabling Program Trace Correlation Messages for direct branch function calls allows the optimum message generation for function run-time measurements. The screenshots below show this for the TRACE32 command **Trace.Chart.Func**.

Legend:

- I: Indirect Branch Message generated for function exits ("BLR"), function pointers, interrupts etc.
- I: Direct Branch Message generated for function calls (opcode "BL")
- I: Direct Branch Message generated for conditional branches

#### BTM ON

more trace messages are generated than required.



### BTM ON + HTM OM

too little trace messages are generated for an accurate run-time measurement.



### BTM ON + HTM ON + BL\_HTM ON

an optimum number of trace messages is generated for an accurate run-time measurement.

c\main 00		· · · · · · · · · · · · · · · · · · ·	
\Tunc2		 	<b>.</b>
\func1 III			
func2a	 	 · · · · · · · · · · · · · · · · · · ·	· ·
- and a bill	 	 	

The Nexus Module implemented on the MPC57xx/SPC57x is able to add a timestamp field to the Nexus messages. The timestamp value is applied to the messages as they enter the Nexus message queues.

To use this feature proceed as follows:

1.	Check TimeStamps	in the NEXUS configuration window.
----	------------------	------------------------------------

B::NEXUS				- • ×
nexus OFF OFF ON TimeStamps RESet Trace List	selection Selection BTM WTM DQM OTM OFF TDM PTCM PTCM PID_MSR BL_HTM TLBNEW TLBINV	option POTD PTMARK DTMARK STALL OFF Suppression SpenDQM SpenVTM SpenDTM SpenOTM SuprTrRestoid 1/4	Configuration – PortSize MDO16 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE CLIENT2 SELECT NONE CLIENT2 SELECT OFF CLIENT3 SELECT NONE SELECT NONE SELECT NONE SELECT

NEXUS.TimeStamps ON

B:: Trace.List NEXUS TIme.Back DEFault	
🖉 Setup   🗛 Goto   🎁 Find   🙀 Chart.   🍱 Profile   🍱 MIPS   🜩 More   🗶 Less	
record Inexus ti.back  run  addres	s cycle
∠ se_bne	
r se_lwz	
se_bts	
se_bne	
✓ se_blr	
	0404F96 ptrace
se_add se_mt1	
se_bir	
	0404F24 ptrace
r se_wz	
selwz	
selwz	
se_lwz	
se_mtl	
se_add	i r1,0x20
se_blr	
	04047B4 ptrace
e_lwz	r7,-0x7FE0(r
se_bts	
↓ se_beq	0x4047CC 🔻
< III	►

TRACE32 calculates its trace time information (ti.back) out of the values of the Nexus timestamp field (TS=). To calculate the time information TRACE32 needs to know the core clock frequency.

2. Inform TRACE32 about the core clock frequency.

B::Trace METHOD ——				- • •
Analyzer O C	Analyzer 🔘 Onchip	◎ ART ◎ LO	GGER  SNOOPer Integrator	
				Probe O Probe
state	used	ACCESS	TDelay	] [
DISable		auto 🔻	0.	TrOnchip
OFF	67108864.		0% 🔻	🔑 NEXUS
Arm	- SIZE	- CLOCK		💮 BMC
🔘 trigger	67108864.	40.0MHz	THreshold —	
Dreak			1.25 -	
SPY 🔿	Mode	Mode	□ VCC	Sector Secto
	Fifo	SLAVE	CLOCK	
commands	Stack		<ul> <li>autofocus</li> </ul>	
RESet	C Leash			-
S Init	STREAM		✓ TERMination	
SnapShot	O PIPE			-
List	© RTS		TestFocus	
✓ AutoArm			XX AutoFocus	
			XX ShowFocus	
SelfArm				
e en ann				

Trace.CLOCK 40.MHz

Trace.CLOCK <freq></freq>	Specify core clock frequency.
Trace.CLOCK { <freq>}</freq>	The core clock frequency can be set per core in an SMP system.

If you do not know your core clock frequency you can measure it as follows:

B::BMC									×
- control	<b>P F </b>	- profile		snoop —		SELect			1 [-
RESet	V Freeze	<u>Illui</u> P	ROfile	🔑 SNOC	Per 🔣 🔣 List	CNT0	L	TREE	14
🛛 🛇 Init	AutoInit Attach			SnoopS	et 🚺 PROfileChart	🛛 🔟 sYr	nbol	sYmbol	
counter name	event		atob	size	value	ra	tio	ratio valu	ie
CNT0	PROC-CYC (Processor cycle	2S 👻	OFF	32BIT 👻			)FF 🚽		
- CNT1	OFF (Nothing)		OFF	32BIT			)FF		
CNT2	OFF (Nothing)		OFF	32BIT		0	FF		
CNT3	OFF (Nothing)		OFF	32BIT		0	FF		
									-
•									►

B::BMC.PROfile					
Fetup 🛇 Init	O Hold	egend) 🕩 In) 🕩 [	)• 🗘 In	Ž 🕻 🛦	to
events/sec	-25.0s			0.	_
5000000.					^
4000000.					
3000000.					
2000000.					
1000000				ſ	_
o.					Ŧ
	•			∏ ►	

BMC.CNT0.EVENT PROC-CYC	; configure CNTO to count processor ; cycles
BMC.PROfile	; display the frequency of CNTO

3. If all configurations are done start and stop the program execution.

### 4. Display the result.

B::Trace.l	ist NEXUS TIme.Back DEFault	
Setup record	Coto	un address cycle
		se_bne 0x40512C se_lwz r7,0x0(r6) se_btsti r7,0x4 se_bne 0x40512C se_blr
-00578141	TCODE=1C SRC=0 PT-IBHM MAP=0 I=0001 U=00000838 H=00001F7E TS=00732A39 5.900us	V:00404F96 ptrace se_lwz r0,0x14(r1) se_addi r1,0x10 se_mtlr r0
-00578139	TCODE=1C SRC=0 PT-IBHM MAP=0 I=0004 U=000000B2 H=00000001 TS=00732A42 0.225us	se_blr V:00404F24 ptrace se_lwz r0,0x24(r1) se_lwz r31,0x1C(r1) se_lwz r30,0x18(r1) se_lwz r29,0x14(r1) se_mtlr r0 se_addi r1,0x20 se_blr
-00578137	TCODE=1C SRC=0 PT-IBHM MAP=0 I=0007 U=00000890 H=00000001 TS=00732A59 0.575us	V:004047B4 ptrace e_lwz r7,-0x7FE0(r se_btsti r7,0x1F se_beq 0x4047CC

Adding timestamp information to Nexus messages has the following advantages:

- The time is more precise, because it is added at the trace source. Parallel execution is clearly visible.
- Nexus timestamps are the only way to get time information for trace-to-memory (onchip trace).
- Nexus timestamps solve some issues of the serial trace recording.

But Nexus timestamps have also disadvantages:

- They need additional bandwidth (approx. 20%).
- The TRACE32 trace decoding becomes slower, since the time information has to be calculated for the complete recording before it can be displayed (Tracking).
- Since TRACE32 uses a fixed core frequency to calculate trace time information out of the Nexus timestamps, this calculation is not possible for variable clock frequencies.

To display only the Nexus timestamp information in the trace display, use the following command:

Trace.List Nexus CLOCKS.Back DEFault TIme.Back.OFF

• It may happen, that not all cores in a chip provide the ability to generate Nexus timestamps.

If NEXUS TimeStamps is unchecked, the TRACE32 tool timestamp mechanism is used. This means a Nexus message is timestamped after it is completely received and stored into the trace memory of the TRACE32 tool. The TRACE32 tool timestamp has a resolution of 20ns for POWERTRACE/ETHERNET or POWERTRACE PX and 5 ns for POWERTRACE II / POWERTRACE III. The time is less precise, because it is added at the trace sink. The merging of the parallel trace streams to a single serial trace stream and the TRACE32 recording logic are the main reasons that make TRACE32 tool timestamp less precise.

**SMP systems:** Due to the fact that one TRACE32 instance is used to control all cores of the SMP system, the message setup is identical for all controlled cores.

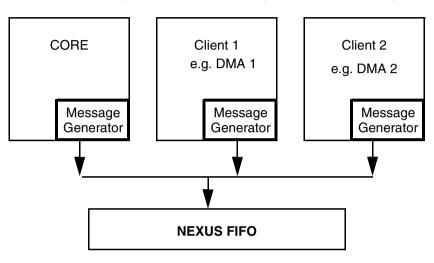
**AMP systems:** Due to the fact that one TRACE32 instance is used per core, an individual message setup per core is possible.

A MPC5xxx/SPC5xxx core can provide several models for the trace clients.

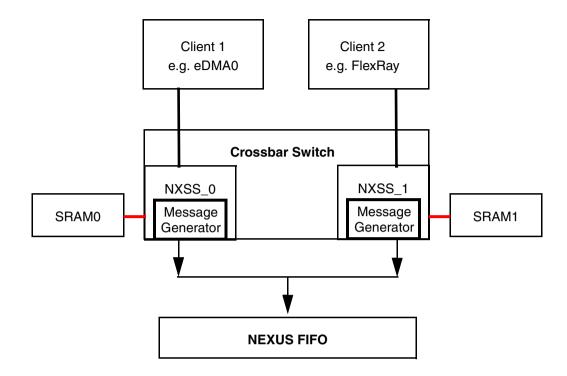
### **Trace Client Types**

### **Dedicated Trace Clients**

Each client that can generate NEXUS messages has its own Message Generator.

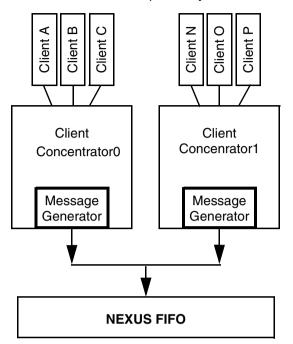


A port sniffers is used to generate the NEXUS messages for the selected clients.



Please be aware, that the NXSS (Nexus Crossbar Slave Port Data Trace Module) can only snoop read/write accesses from the selected trace client to the connected SRAM.

The Message Generator of the Client Concentrator generates NEXUS messages for the connected clients. Clients can be enabled independently.



## Configuration

B::NEXUS				- • •
nexus OFF ON RESet Trace	selection BTM HTM OTM WTM DTM Write	option	Configuration PortSize MD012 • PortMode 1/2 • DDR PCRCONFIG	CLIENT1 SELECT MODE DTM V CLIENT2 SELECT NONE MODE OFF V

NEXUS.CLIENT1 SELECT DMA

NEXUS.CLIENT1 MODE DTM

; specify DMA for CLIENT1

- ; NEXUS messages are generated
- ; according to the DTM settings
- ; (here write)

Possible clients:

DMA/DMA2	DMA controller					
ETHERNET	Ethernet controller					
FLEXRAY	FlexRay controller					
PDI	Parallel Digital Interface controller					

Possible modes:

OFF	No NEXUS messages are generated.
Read	Generate NEXUS messages for all read accesses.
Write	Generate NEXUS messages for all write accesses.
ReadWrite	Generate NEXUS messages for all read and write accesses.
DTM	The client is using the DTM settings.

B::Trace.Lis	t	
🌽 Setup	📭 Goto 🎁 Find 👬 Chart 🔛 Profile 🔛 MIPS 🗢 More 🗶 Less	
record	run address cycle data symbol 1	ti.back
+00000826	<pre>blt 0x40001DD4 ; 0x40001DD4 (-) P:40001DD4 ptrace</pre>	1.340us
	and r9,r28,r3 or. r10,r9,r10 slwi r30,r30,0x1 ; r30,r30,1	*
	inslwi r30,r31,0x1F; r30,r31,1,31 slwi r31,r31,0x1 ; r31,r31,1 beq 0x40001DA0 ; 0x40001DA0 (-)	
+00000829	<pre>TARGET FIFO OVERFLOW, PROGRAM FLOW LOST P:40001DB8 ptrace blt 0x40001DD4 ; 0x40001DD4 (-) subfc r31,r27,r31</pre>	4.660us
	mr r28,r28 ori r29,r29,0x1 ; r29,r29,1	+ 

**TARGET FIFO OVERFLOW, PROGRAM FLOW LOST** occurs, when so much trace information is generated that it can not be buffered in the NEXUS FIFO.

In order to get an immediate display of the trace contents TRACE32 uploads only the currently displayed section from the physical trace memory to the host. To check if there are FIFOFULLs it is recommended to upload the complete trace contents to the host by the command: **Trace.FLOWPROCESS**.

The number of FIFOFULL is printed to the TRACE32 state line as result of the following command:

```
Trace.Find FIFOFULL /ALL
```

B::   found in (-80530636	58.)(-1.)	) 49942. ti	imes
components trace	Data	Var	List
P:00178406			

The following TRACE32 functions allows you to process the result in a script:

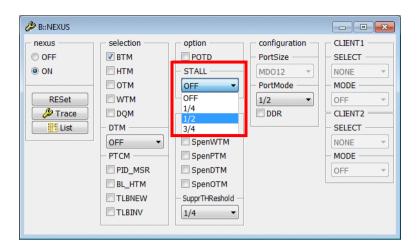
FOUND.COUNT()

The single FIFOFULLs can be found in the trace:

B::Trace.Lis	t							[	
🌽 Setup	Goto	👘 Find	Chart	P	rofile 🛛 🚺 MI	PS 🔷 More	Less		
record	run addre	SS C	ycle dat	a	s	ymbol		ti.	back
	slwi		r30,0x1		r30,r30,1				*
	inslw		r31,0x1,0		r30,r31,1,	31			
	slwi		r31,0x1		r31,r31,1	2.5			+
	beq		001DA0		0x40001DA0	(-)			
+00000829				OGRA	M FLOW LOST	\diaba in	+\Clabal\ d a	1	4 66000
+0000829	blt	0001DB8 p	trace 001DD4		0x40001DD4		nt\Global\_d_c	11V+0X28C	4.660us
	subfc		r27,r31	,	0X40001DD4	(-)			
	mr	r28,			👸 Trace Find				
	ori		r29,0x1	;	F3 Hace Hind			Ľ	
	subfe		r26,r30	,	Expert	Cycle	Group	Changes	🔘 Up
	oris		r28,0x0	;		-		Signal	O Down
	oris	r29,	r29,0x0					I Signal	O DOWII
	and		r29,r4		– items ––––				
	and		28,r3		FIFOFULL				
	or.		r9,r10		TIOTOLL				· · · · ·
	şlwi		r30,0x1		L				
	inslw		r 31,0x1,0	α⊢;					
	slwi bea		r31,0x1 001DA0						
+00000831		0001DA0 p		2					
+00000001	cmplw								
	[ s]wi		r28,0x1						
	inslw		r29,0x1,0	1F':	Find Next	Find First	Find Here Fin	d All Clear	Cancel
	4		,,	01111111	I III NEXL	i ind i ii sc		Ciedi	Curicer
-									

FIFOFULLs may occur during your tests, they are not errors. But FIFOFULLs may disturb your trace analysis. There are various strategies to avoid FIFOFULLs.

### Stall Program Execution on Overflow Threat



Overrun control								
STALL OFF	Generate overrun message when a new message can not be queued due to the NEXUS FIFO being full. No new message is queued to the NEXUS FIFO until it is completely empty.							
STALL ON (2003 Standard)	Stall the program execution whenever the on-chip NEXUS FIFO threatens to overflow.							
STALL 1/4 (2008/2012 Standard)	Stall the program execution when 1/4 of the on-chip NEXUS FIFO is filled.							
STALL 1/2 (2008/2012 Standard)	Stall the program execution when 1/2 of the on-chip NEXUS FIFO is filled.							
STALL 3/4 (2008/2012 Standard)	Stall the program execution when 3/4 of the on-chip NEXUS FIFO is filled.							

NEXUS.STALL 3/4

Since Data Trace Messages are high-risk for NEXUS FIFOs getting full, it may be helpful to suppress these messages when the NEXUS FIFO reaches a certain fill level.

B::NEXUS				- • ×
nexus	selection	option	configuration —	CLIENT1
OFF	<b>B</b> TM	POTD	- PortSize	- SELECT
ON	🔲 HTM	- STALL	MDO12 -	NONE -
	🔲 ОТМ	OFF 🔻	- PortMode	- MODE
RESet	MTM 🗌		1/2 🔻	OFF 👻
🥬 Trace	DQM	suppression	DDR	CLIENT2
📰 List	- DTM	SpenDQM		- SELECT
L	ReadWrite 🔻	SpenWTM		NONE 👻
	- PTCM	SpenPTM		- MODE
	PID_MSR	SpenDTM		OFF 👻
	BL_HTM	SpenOTM		
	TLBNEW	- SupprTHReshold -		
	TLBINV	3/4 💌		

NEXUS.DTM ReadWrite	
NEXUS.SupprTHReshold 3/4	; Sets the NEXUS FIFO fill level, at which ; messages will be suppressed
NEXUS.SpenDTM ON	; Suppress Data Trace Messages when the ; NEXUS FIFO reaches the specified filling ; level
	; Start and stop the program execution
Trace.FindAll,CYcle Write	; Search for all write accesses

👸 B::Trace.Find/	All , CYcle Write			- • ×
3690883	run address	cycle	data symbol	ti.back
-0006475828	D:40007F10	wr-long	40007F68 \\diabc\Global\SP_TEST+0x518	3.950us 🔺
-0006475825	D:40007F20	wr-long	00000000 \\diabc\Global\SP_TEST+0x534	1.105us 🖃
-0006475824	D:40007F30	wr-long	00000000 \\diabc\Global\SP_TEST+0x538	0.620us 🗐
-0006475823	D:40007F34	wr-long	00000000 \\diabc\Global\SP_TEST+0x53C	0.620us 🍈
-0006475782	D:400040EC	wr-long		18.745us 🔺
-0006475780	D:400040F0	wr-word		0.865us
-0006475778	D:400040F0	wr-long	0001E000 \\diabc\Global\vbfield+0x8	1.230us
-0006475776	D:400040F0	wr-long	0001A000 \\diabc\Global\vbfield+0x8	1.235us
-0006475775	D:400040E8	wr-long	FC080800 \\diabc\Global\vbfield	0.615us
-0006475771	D:400040E8	wr-long	FFF80800 \\diabc\Global\vbfield	1.730us
-0006475767	D:400040E8	wr-long	FFFFF800 \\diabc\Global\vbfield	1.475us 📃
-0006475763	D:40007F68	wr-long	40007F80 \\diabc\Global\SP_TEST+0x570	1.730us
-0006475761	D:40007F78	wr-long		0.985us
-0006475759	D:40007F70	wr-long		0.740us
-0006475758	D:40007F84	wr-long	400011C8 \\diabc\Global\SP_TEST+0x58C	0.620us 👻
	1.			►
	3			

The NEXUS protocol does not indicate the message suppression. But read/write cycles that can not be assigned to its instruction (displayed in red) are a good indicator, the a message suppression occurred.

If you do not want to stall the program execution or suppress messages, just reduce the number of the generated trace messages:

- Enable HTM (NEXUS.BTM ON and NEXUS.HTM ON)
- Switch DTM to off when possible (**NEXUS.DTM** OFF).
- Disable the NEXUS message generation for cores you are not interested in for you current analysis.
- Filter the DTMs. Refer to "Filter and Trigger (Core) Overview", page 101 for details.

TRACE32 indicates FLOWERRORS:

- If the instruction flow information generated by NEXUS does not match with the code image in the target.
- If invalid NEXUS messages are generated.

The complete number of FLOWERRORS is printed to the TRACE32 state line as result of the following command:

```
Trace.Find FLOWERROR /ALL
```

The single FLOWERROR can be found in the trace:

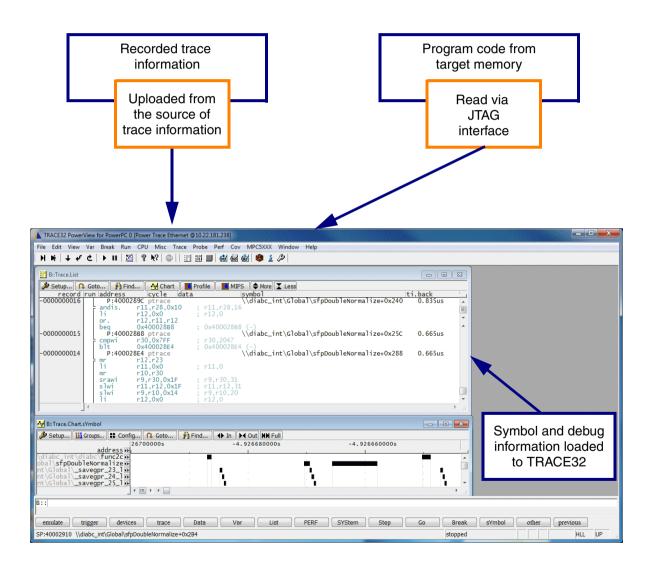
B::Trace.Li	t												23
🌽 Setup				Chart	📔 Pro	ofile		5 🔶 More	Less				
ERRORS	rur	1 addres		cycle	data			symbol			ti	.back	_
-01835974 -01835977 -01835977 -01835977 -01835966 -01835966 -01835966 -01835966 -01835966 -01835966 -01835966 -01835965 -01835955 -01835955 -01835955 -01835955		and or. slwi inslwi slwi	r9, r10 r30 i r30 r31	cycle r28,r3 ,r28,r3 ,r30,0x1 ,r31,0x1 XPECTED 1 unknown	,0x1F	ir 30 ir 31 BRANC	),r30,1 ),r31,1	, 31 © Cycl		Group	) ti Changes Signal	1.340us Up Down	
	•			andrown									

FLOWERRORs are errors and it is recommended to fix them. Please contact your local Lauterbach support if you need assistance.

## Sources of Information for the Trace Display

In order to provide an intuitive trace display the following sources of information are merged:

- The trace information recorded.
- The program code from the target memory read via the JTAG interface.
- The symbol and debug information already loaded to TRACE32.



The main influencing factor on the trace information is the NEXUS hardware module. It specifies what type of trace information is generated for the user.

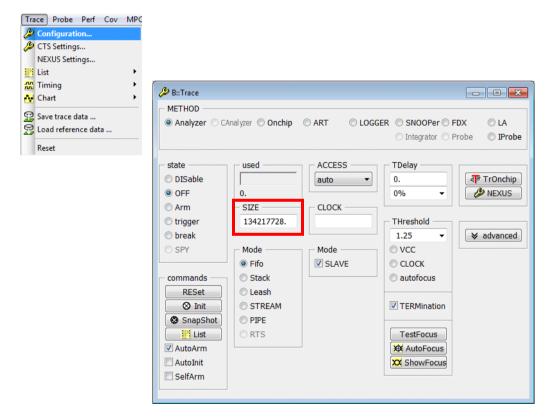
Another important influencing factor are the settings in the TRACE32 Trace Configuration window. They specify how much trace information can be recorded and when the trace recording is stopped.

### **Recording Modes**

The **Mode** settings in the Trace configuration window specify how much trace information can be recorded and when the trace recording is stopped.

The following modes are provided:

 Fifo, Stack, Leash Mode: allow to record as much trace records as indicated in the SIZE field of the Trace Configuration window.



**STREAM Mode (PowerTrace II hardware only):** STREAM mode specifies that the trace information is immediately streamed to a file on the host computer. STREAM mode allows a trace memory size of several T Frames.

• **PIPE Mode (PowerTrace II hardware only):** PIPE mode specifies that the trace information is immediately streamed to a named pipe on the host computer.

PIPE mode creates the path to convey trace raw data to an application outside of TRACE32 PowerView. The named pipe has to be created by the receiving application before TRACE32 can connect to it.

Trace.Mode PIPE	
Trace.PipeWRITE <pipe_name></pipe_name>	Connect to named pipe
Trace.PipeWRITE \\.\pipe\ <pipe_name></pipe_name>	Connect to named pipe (Windows)
Trace.PipeWRITE	Disconnect from named pipe

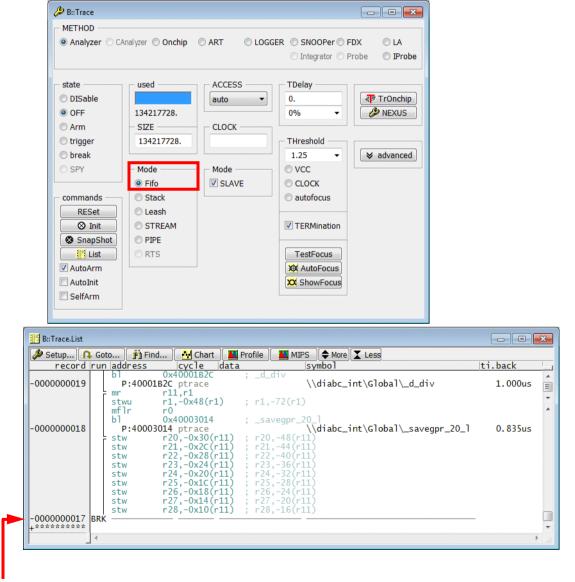
Trace.Mode PIPE	; switch trace to PIPE mode
Trace.PipeWRITE \\.\pipe\pproto00	; connect to named pipe ; (Windows)
Trace.PipeWRITE	; disconnect from named pipe



NEXUS packets (no tool timestamp) are conveyed in PIPE mode.

**RTS Mode (PowerTrace II hardware only):** RTS mode enables the processing while the trace data are recorded. The main use case for **RTS** is a live display of the code coverage results.

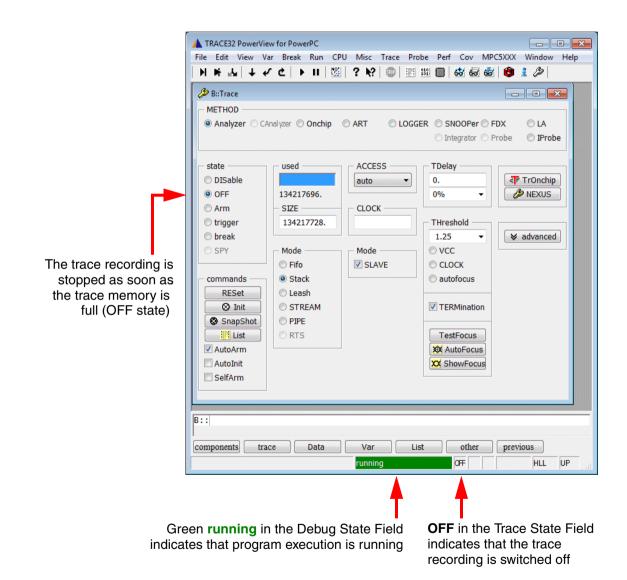
Trace.Mode Fifo ; default mode ; when the trace memory is full ; the newest trace information will ; overwrite the oldest one ; the trace memory contains all ; information generated until the ; program execution stopped



In Fifo mode negative record numbers are used. The last record gets the smallest negative number.

Trace.Mode Stack

- ; when the trace memory is full ; the trace recording is stopped ; the trace memory contains all ; information generated directly
- after the start of the program
- ; execution

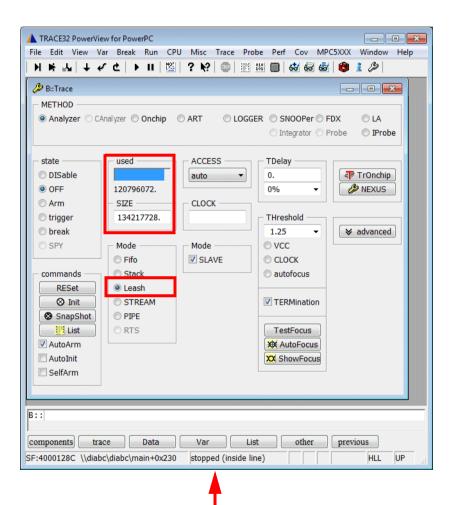


Since the trace recording starts with the program execution and stops when the trace memory is full, positive record numbers are used in **Stack** mode. The first record in the trace gets the smallest positive number.

🌽 Setup 🔃	Goto 🤖 Fi	nd 🚺 👬 Chart 🛛 🌉	Profile MIPS 🗢 More 🗶 Less	
record	run address	cycle data	a symbol	ti.back
	GO			
+000000001		3038 ptrace	\\diabc_int\Global\_savegpr_29_	_1
	stw		; r29,-12(r11)	
	stw	r30,-0x8(r11)		
	stw	r31,-0x4(r11)		
	stw	r0,0x4(r11)	; r0,4(r11)	
	blr	1020	\\	2.10
+000000002		1B3C ptrace	\\diabc_int\Global\_d_div+0x10	2.16
	mr	r30,r3		
	srwi clrlwi	r10,r30,0x14 r23,r10,0x15	; r10,r30,20	
	srwi	r12,r30,0x15		
	cmpwi	r23.0x7FF	; r23,2047	
	mr	r26.r5	, 123,2047	
	mr	r22,r12		
	mr	r27,r6		

Trace.Mode Leash

- ; when the trace memory is nearly ; full the program execution is ; stopped ; Leash mode uses the same record
- ; numbering scheme as Stack mode



The program execution is **stopped** as soon as the trace buffer is nearly full.

Since stopping the program execution when the trace buffer is nearly full requires some logic/time, **used** is smaller than the maximum **SIZE.** 

NEXUS.HTM ON	; enable Indirect Branch History ; Messaging to get compact raw ; trace data
Trace.Mode STREAM	; STREAM the recorded trace ; information to a file on the host ; computer
	; STREAM mode uses the same record ; numbering scheme as Stack mode

The trace information is immediately streamed to a file on the host computer after it was placed into the trace memory. This procedure extends the size of the trace memory to several T Frames.

- STREAM mode requires a TRACE32 trace hardware that allows streaming the trace information while recording. This is currently supported by PowerTrace II.
- STREAM mode required a 64-bit host computer and a 64-bit TRACE32 executable to handle the large trace record numbers.

By default the streaming file is placed into the TRACE32 temporary directory (**OS.PresentTemporaryDirectory()**).

The command **Trace.STREAMFILE** *<file>* allows to specific a different name and location for the streaming file.

```
Trace.STREAMFILE d:\temp\mystream.t32 ; specify the location for ; your streaming file
```

TRACE32 stops the streaming when less then the 1 GByte free memory left on the drive by default.

The command Trace.STREAMFileLimit <+/- limit in bytes> allows a user-defined free memory limitation.

```
Trace.STREAMFileLimit 500000000. ; streaming file is limited to
; 5 GByte
Trace.STREAMFileLimit -5000000000. ; streaming is stopped when less
; the 5 GByte free memory is left
; on the drive
```

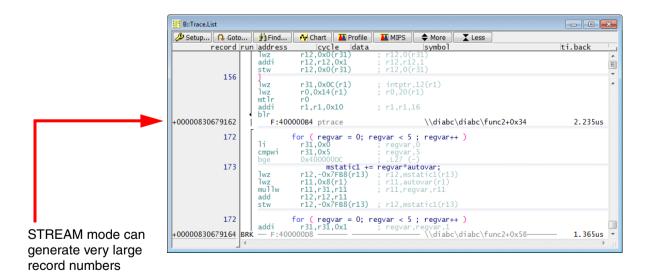
Please be aware that the streaming file is deleted as soon as you de-select the STREAM mode or when you exit TRACE32.

#### In STREAM mode the used field is split:

Indication of how much trace information is intercepted in the trace memory of the TRACE32 trace tool

B::Trace METHOD Analyzer C	Analyzer 🔘 Onchip		ER 🔘 SNOOPer 🔘 FDX	C LA IProbe
DISable	- used	ACCESS auto		rOnchip NEXUS
<ul> <li>Arm</li> <li>trigger</li> <li>break</li> <li>SPY</li> <li>commands</li> <li>RESet</li> <li>② Init</li> <li>③ SnapShot</li> <li>③ List</li> <li>☑ AutoArm</li> <li>AutoInit</li> <li>③ SelfArm</li> </ul>	Mode Fifo Stack Eeash PIPE RTS	CLOCK	THreshold	dvanced

Number of records saved to streaming file



STREAM mode can only be used if the average data rate at the trace port does not exceed the maximum transmission rate of the host interface in use. Peak loads at the trace port are intercepted by the trace memory, which can be considered to be operating as a large FIFO.

If the average data rate at the trace port exceeds the maximum transmission rate of the host interface in use, a **PowerTrace FIFO Overrun** occurs. TRACE32 stops streaming and empties the PowerTrace FIFO. Streaming is re-started after the PowerTrace FIFO is empty.

A PowerTrace FIFO Overrun is indicated as follows:

1. A ! in the **used** area of the Trace configuration window indicates an overrun of the PowerTrace FIFO.

B::Trace				
Analyzer OCA	knalyzer 🔘 Onchip	© ART © LOG		) LA ) IProbe
state © DISable	used	ACCESS	TDelay	Onchip
© OFF	997015552. !	auto		EXUS
Arm	- SIZE	CLOCK		
© trigger © break			THreshold	vanced
SPY	Mode	Mode	0 VCC	
	© Fifo	SLAVE	CLOCK	
commands	© Stack		autofocus	
RESet	<ul><li>Leash</li><li>STREAM</li></ul>		TERMination	
SnapShot	© PIPE © RTS		TestFocus	
V AutoArm AutoInit	L	]	XX AutoFocus	
SelfArm				

2. The OVERRUN is indicated in all trace display windows.

🔑 Setup 🔼 Goto.	🛉 🛉 Find	🕂 Chart 🔛 Profile	MIPS More Less	
OVERRUN	un address	cycle data	symbol	ti.back
	mr cmpw bge li blr	r8,r5 r7,r8 0x40001B74 r3,0x0	; 0x40001B74 (-) ; r3,0	[
+00001008068134	F:4000	00244 ptrace r3,0x0 0x40000298	\\diabc\diabc\func2c+0x74 ; r3,0 ; .L70 (-)	1.110us
215			regvar*autovar;	
	lwz lwz lwz lwz bl	r3,0x10(r1) r4,0x14(r1) r5,0x8(r1) r6,0x0C(r1) 0x40001C48	; r3,regvar(r1) ; r4,20(r1) ; r5,autovar(r1) ; r6,12(r1) : _d_mul	C
+00001008068135	E-400	01C48 ptrace	\\diabc\Global\_d_mul	0.865us

The trace buffer can either sample or allows the read-out for information display.

METHOD Analyzer OCA	Analyzer 🔘 Onchip	© ART ◎ LOG	GER © SNOOPer © F	
state	– used –	ACCESS		
OISable		auto 🔻	0.	TrOnchip
OFF	9867656.		0% -	A NEXUS
Arm	- SIZE	CLOCK		
trigger	134217728.		THreshold	
o break			1.25 -	Sector Secto
SPY	Mode	Mode	© vcc	
	Fifo	SLAVE	CLOCK	
commands	Stack		o autofocus	
RESet	🔘 Leash			
S Init	STREAM		TERMination	
SnapShot	© PIPE			
List	© RTS		TestFocus	
🗹 AutoArm			XX AutoFocus	
🕅 AutoInit			X ShowFocus	
SelfArm				

States of the Trace	
DISable	The trace is disabled.
OFF	The trace is not sampling. The trace contents can be red-out and displayed.
Arm	The trace is sampling. The trace contents can not be red.

The current state of the trace is always indicated in the Trace State field of the TRACE32 state line.

B::		
emulate trigger devices trace Data Var List PERF SYStem	other	previous
running	Trace	HLL UP
	0 Trace DISable	
	OFF	
	<ul> <li>ARM</li> </ul>	

The Trace states trigger and break are introduced in detail later in this training.

● METHOD ● Analyzer ○ CA	nalyzer 🔘 Onchip	© ART © LOG	GER O SNOOPer O FDX O LA
state DISable OFF Arm trigger break SPY	used	ACCESS auto	TDelay 0. 0% Tronchi NEXUS NEXUS NEXUS NEXUS NEXUS NEXUS NEXUS NEXUS NEXUS
Commands RESet S Init SapShot List AutoArm AutoInit SelfArm	<ul> <li>Stack</li> <li>Leash</li> <li>STREAM</li> <li>PIPE</li> <li>RTS</li> </ul>		<ul> <li>autofocus</li> <li>TERMination</li> <li>TestFocus</li> <li>X AutoFocus</li> <li>ShowFocus</li> </ul>

Init Button	Clear the trace memory. All other settings in the Trace configuration window remain valid.
AutoInit CheckBox	ON: The trace memory is cleared whenever the program execution is started (Go, Step).

The focus of the Trace configuration window is:

- Setup and maintenance of the TRACE32 trace tool (METHOD Analyzer).
- Setup and maintenance of the onchip trace (METHOD Onchip).

**SMP systems:** Due to the fact that one TRACE32 instance is used to control all cores, setups and states are identical for all controlled cores.

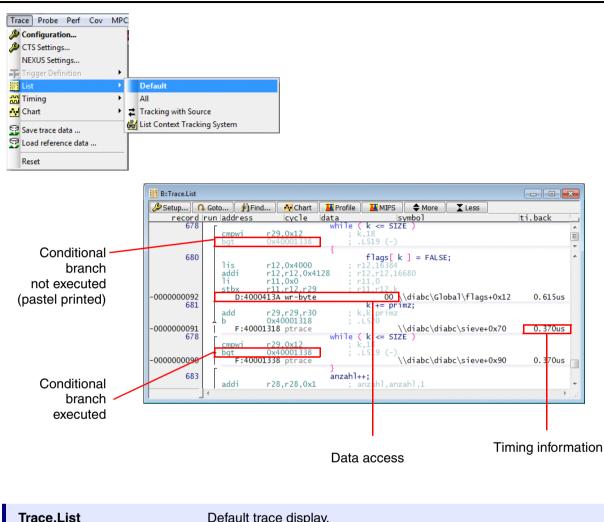
**AMP systems:** Due to the fact that the setups and states are maintained by multiple TRACE32 instances, the TRACE32 Resource Management maintains consistency for all joint settings and joint states.

Consistency maintenance means: status changes in one TRACE32 instance affect all other TRACE32 instances.

### **Joint Settings and States**

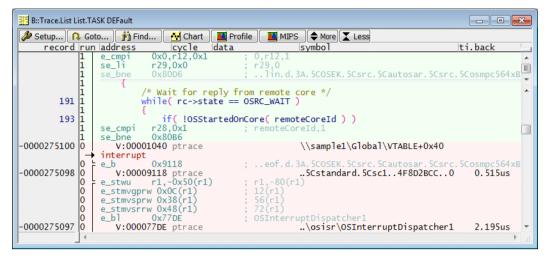
🔑 B::Trace			[	- • •
METHOD				
Analyzer O CA	nalyzer 🔘 Onchip	◎ ART ◎ LOGGI	ER 🔘 SNOOPer 🔘 FDX	© LA
			🔘 Integrator 🔘 Prob	e 🔘 IProbe
L				
state	used	ACCESS	- TDelay	
DISable		auto 👻	0.	TrOnchip
OFF	5207992.		0% 👻	NEXUS
Arm	- SIZE	CLOCK		
🔘 trigger	134217728.		– THreshold ––––	
O break			1.25 🔻	
O SPY	– Mode –––––	Mode	© VCC	
	Fifo	SLAVE	CLOCK	
– commands ——	Stack		autofocus	
RESet	Ceash			
Ø Init	STREAM		TERMination	
SnapShot	© PIPE			
List	© RTS		TestFocus	
AutoArm			XX AutoFocus	
AutoInit			X ShowFocus	
SelfArm				

### **Default Listing**



Default trace display.

The trace information for all cores is displayed by default in the **Trace.List** window if you are working with an **SMP** system. The column run and the coloring of the trace information are used for core indication.

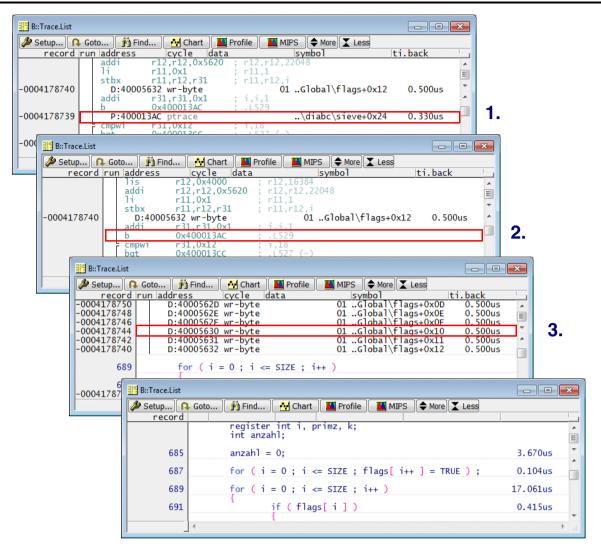


Trace.List /CORE <n>

The option CORE allows a per core display.

B::Trace.List /	CORE 1	
		More Less
record		ymbol ti.back
-0000001083	e_lwz r0,0x44(r1) ; r0,68(r1) se_mtlr r0 e_addi r1,r1,0x40 ; r1,r1,64 se_blr V:0000912C ptracea e_lmvsrrw 0x48(r1) ; 72(r1) e_lmvsprw 0x38(r1) ; 56(r1) e_lmvgprw 0x0C(r1) ; 12(r1) e_addi r1,r1,0x50 ; r1,r1,80 se_rfi	andard.5Csc14F8D2BCC0+0x14 7.605us
-0000001077		\sample1\osset\StartOS+0x23C 5.545us
-0000001073		\sample1\osset\StartOS+0x238 1.155us
-0000001066		\sample1\osset\StartOS+0x23C 2.840us

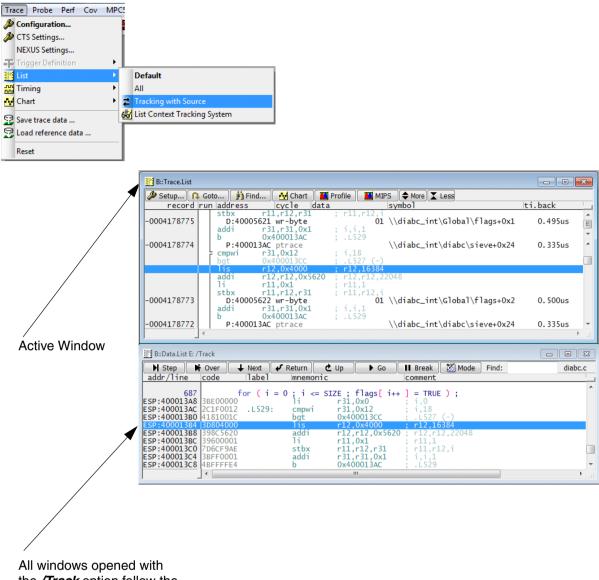
## **Basic Formatting**



1. time Less	Suppress the display of the program trace package information (ptrace).
2. time Less	Suppress the display of the assembly code.
3. time Less	Suppress the data access information (e.g. wr-byte cycles).

The *More* button works vice versa.

## Correlating the Trace Listing with the Source Listing



the **/Track** option follow the cursor movements in the active window

# Browsing through the Trace Buffer

B::Trace.List						
🎾 Setup 🖪	🖡 Goto 🏼 👘 Fin			📕 MIPS 🛛 🖨 More 🗶 Less		
record	run address		data	symbol	ti	.back
-0000001002	P:40003	0x40001DA0 1DA0 ptrace r30,r26	; 0x400	001DA0 (-) iabc_int\Globa	1\_d_div+0x274	1.165us
	slwi inslwi slwi bne	r28,r28,0x1 r28,r29,0x1 r29,r29,0x1 0x40001DB8	0x1F; r28, ; r29,1	r29,1,31		*
-0000001001	= blt	DB8 ptrace 0x40001DD4	; 0x400			
-0000001000		1DD4 ptrace		Record / Time / Boo	kmark —	
	and and or. slwi	r10,r29,r4 r9,r28,r3 r10,r9,r10		-1000.	•	Goto
	inslwi slwi	r30,r30,0x1 r30,r31,0x1 r31,r31,0x1	; r31,i	Previous	First Trigger	Zero
-0000000999	P:4000: cmplw	0x40001DA0 LDAO ptrace r30,r26	; 0x400	0010 Next	Last Ref	
	4					Cancel

Pg↑	Scroll page up.
Pg ↓	Scroll page down.
Ctrl - Pg ↑	Go to the first record sampled in the trace buffer.
Ctrl - Pg ↓	Go to the last record sampled in the trace buffer.

### **Default Display Items**

🔑 Setup 🚺	Goto 🧃	Find 🔂 🕂 Chart	📕 Profile			
	run addres	s cycle	data	symbol	ti.back	_
678	cmpwi bgt	<b>r29,0x12</b> 0x40001338		<mark>( k &lt;= SIZE )</mark> k,18 .L519 (-)		* 
680	lis addi li stbx	r12,0x4000 r12,r12,0x41 r11,0x0 r11,r12,r29	28	<pre>flags[ k ] = FALSE; r12,16384 r12,r12,16680 r11,0 r11,r12,k</pre>		•
-0000000092 681		r29,r29,r30 0x40001318	;	00 \\diabc\Global\flags+0x k += primz; k,k,primz 1520	12 0.615us	
-0000000091 678	F:4	0001318 ptrace	while	<pre>\\diabc\diabc\sieve+0x7 ( k &lt;= SIZE )</pre>	0 0.370us	
-0000000090	<pre>     cmpwi     bgt     F:4 </pre>	r29,0x12 0x40001338 0001338 ptrace		k,18 .L519 (-) \\diabc\diabc\sieve+0x9	0 0.370us	-
683	addi	r28.r28.0x1	anzahl	++; anzahl,anzahl,1		

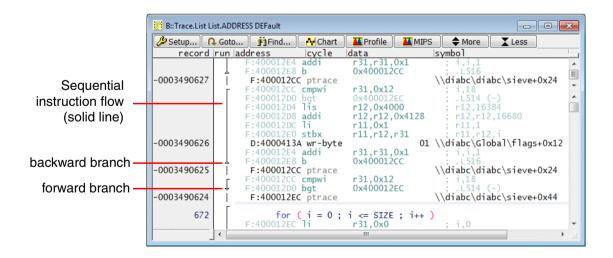
#### Column record

Displays the record numbers

#### Column run

The column run displays some graphic element to provide a quick overview on the instruction flow.

Trace.List List.ADDRESS DEFault



The column run also indicates interrupts and TRAPs.

B::Trace.List			
Setup	Goto ) 🎁 Find 🚺 Chart 🛛 🚻 Profile 🖉 🔛	AIPS 🔶 More 🗶 Less	
record	run address cycle data	symbol	ti.back
	e_rlwinm r6,r3,0x3,0x0,0x1C; r6,si	it,3,0,28	
	se_sraw r7,r6 e_lis r5,0xFFF50000 ; r5,42		
	e_stw r7,-0x7FE0(r5) ; r7,-3	(736(r5)	
-0000010061	<pre>se_blr v:40005408 ptrace</pre>	\\imO2_bf1x\imO2\FuncTASK3+0x38	1.250us
	se_lbz r7,0x0(r31) ; r7,0(n → interrupt	31)	
-0000010060	V:40000040 ptrace	<pre>\\im02_bf1x\Global\VTABLE+0x40</pre>	1.250us
-0000010059	V:40004FEE ptrace	x\Global\OSInterruptDispatcher	0.750us
	e_stwu r1,-0x50(r1) ; r1,-8		
	e_stmvgprw 0x0C(r1) ; 12(r1) e_stmvsprw 0x38(r1) ; 56(r1)		
	e_stmvsrrw 0x48(r1) ; 72(r1		
		erruptDispatcher1	
-0000010058	V:40003D50 ptrace	.x\osisr\OSInterruptDispatcher1	0.750us 👻
	• • • • • • • • • • • • • • • • • • •		h. I⊀

#### Column cycle

The main cycle types are:

- ptrace (program trace information)
- rd-byte, rd-word, rd-long (read access)
- wr-byte, wr-word, wr-long (write access)
- owner (ownership trace messages)
- unknown (Branch Trace Messages that can not be decoded)

B::Trace.List NEXUS DEFault				
	un address	cycle data unknown unknown unknown unknown unknown 012F0 ptrace r31,0x12 0x40001344	symbol sieve+0x48 ; .LS17 (-)	ti.back 1.110us 0.370us 1.480us 0.495us 1.355us 0.740us
674 -0000999806 TCODE=03 SRC=0 PT-DBM ICNT=0007	lis addi lbzx cmpwi ↓ beq F:400		; r12,16384	0.860us -

The decoding of the Branch Trace Messages can start, as soon as a full address (F-ADDR) is exported. Branch Trace Messages that can not be decoded are marked as **unknown**.

#### Column address/symbol •

B::Trace.List								×
🔑 Setup 🔒	Goto 🏼 🎁 Fir	nd 🔂 💀 Chart	🌉 Profile	MIPS	🔷 More	Less		
	run address	cycle	data		ymbol		ti.back	_
678	cmpwi bgt	<b>r29,0x12</b> 0x40001338		<mark>( k &lt;= SIZ</mark> k,18 .L519 (-)	Ε)			* 
680	lis addi li	r12,0x4000 r12,r12,0x41 r11,0x0 r11,r12,r29		flags[k r12,16384 r12,r12,160 r11,0 r11,r12,k	] = FALSE	;		*
-000000092	D:400	0413A wr-byte		00 \\	\diabc\Glo	bal\flags+0x12	0.615us	
681	add	r29,r29,r30	ł	k += prıı k,k,primz	mz;			
-0000000091	F:400	01318 ptrace		1	\diabc\dia	bc\sieve+0x70	0.370us	
-0000000090	∫ cmpwi ↓ bgt F:400	r29,0x12 0x40001338 01338 ptrace		( k <= SIZ k,18 .L519 (-)		bc\sieve+0x90	0.370us	
0000000000	1.400	01330 perace	}	1	(urabe (ura	be (STEVETONDO	0.57003	
683	addi	r28,r28,0x1	anzah1 ;	++; anzahl,anza	ahl,1		)	<b>.</b> ∎

The **address column** shows the following information: <access class>:<logical\_address>

Access Classes						
F	Program address, disassembly shows standard PowerPC instructions					
V	Program address, disassembly shows VLE encoded instructions					
D	Data address					

The symbol column shows the corresponding symbolic address.

## • Column ti.back

B::Trace.List										×
🔑 Setup 👔	Goto	Find	Chart	🔼 Profile	MIPS	And the More	Less			
	run add	ress	cycle	data		ymbol			ti.back	_
678	cmp bgt		<b>9,0x12</b> 40001338		( k <= SIZ k,18 .L519 (-)	E )				•
680	lis add li stb	li r1 r1	2,0x4000 2,r12,0x41 1,0x0 1,r12,r29	.28	flags[k r12,16384 r12,r12,16 r11,0 r11,r12,k	] = FALSE	Ξ;			*
-000000092			A wr-byte	,	00 \\		bal\flags	+0x12	0.615us	
-0000000091	add A b		9,r29,r30 40001318 8 ptrace		k += prin k,k,primz .L520		abc\sieve+	0x70	► 0.370us	
678 -000000090	↓ cmp		9,0x12 40001338 8 ptrace	;	( k <= SIZ k,18 .L519 (-)		abc\sieve+	0x90	0.370us	
683	add	li r2	8,r28,0x1	} anzahi ;	l++; anzah1,anza	ahl,1				-
	] •								,	

The **ti.back column** shows the time distance to the previous record.

#### **Time Information**

Time.Back         Time relative to the previous record (red)				
Time.Fore	Time relative to the next record (green).			
Time.Zero	Time relative to the global zero point.			

Trace.List TIme.Back TIme.Fore TIme.Zero DEFault

B::Trace.List TI	me.Back TIme.F	ore TIme.Zero DI	EFault						X
🌽 Setup 📭	Goto 🛐	Find 📔 🛃 Cl	hart 🛛 🔼 Profile	e 🚺 MIPS	♦ More  Less				
record	ti.back	ti.fore	ti.zero	run address	cycle	data		symbol	- L- 1
				b	0x400013AC	;	.L529		
-0014175399	0.335us	0.500us	14.435ks	P:4000	013AC ptrace			<pre>\\diabc_int\diabc\sieve+0x24</pre>	
				cmpwi	r31,0x12		i,18		
				bgt	0x400013CC		.L527 (-)		
				lís	r12,0x4000		r12,16384		^
				addi	r12,r12,0x5		r12,r12,22	2048	
				li	r11,0x1		r11,1		
				stbx	r11,r12,r31	7	r11,r12,i		
-0014175398	► 0.500us	0.335us	14.435ks		)5621 wr-byte			\\diabc_int\Global\flags+0x1	
				addi	r31,r31,0x1		i,i,1		
				b	0x400013AC	;	.L529		
-0014175397	0.335us	0.500us	14.435ks		013AC ptrace		1.40	\\diabc_int\diabc\sieve+0x24	
				cmpwi	r31,0x12		i,18		
				bgt lis	0x400013CC		.L527 (-)		
				115	r12,0x4000	;	r12,16384		*
	•				III				5 N

## Set the Global Zero Point (tool timestamp only)

🔑 Setup 📭						More Less				
record t	i.back t	ti.fore	ti.zero	run	address	cycle	data	. 530	symbol	_
0014175399	0.335us	0.500us	0.000us		D R+40001	0x400013AC	. ;	.L529	\\diabc_int\diabc\sieve+0x24	
0014173333	0.00045	0.00003	0.00002	Т	race	112	:	i.18	( (drabe_file (drabe (3)ever0x24	
				R+ S	et Ref	1300		.L527 (-)		
				20 5	et Zero	4000		r12,16384		
							(5620 ;	r12,r12,2	2048	
					oggle Bookma	rk IX.r3		r11,1 r11,r12,i		
0014175398	0.500us	0.335us	0.500us	; 🚳 S	et CTS	byte	,	01	\\diabc_int\Global\flags+0x1	
				Q v	lieur	1,0>	d;	i,i,1		
014175307	0.335	0 500	0.035				;	.L529		
0014175397	0.335us	0.500us	0.835us			ace 12		i,18	\\diabc_int\diabc\sieve+0x24	
				<u> </u>	Chart	1300	$\langle \cdot \cdot \rangle$	.L527 (-)		
				Te	qnore in Statist	4000		r12,16384		
	•				-	-	1			Þ
				_	Jse in Statistic					
				🛉 🖣 F	irst in Statistic					
				🛨 L	ast in Statistic		\			
				± F	ull Statistic		\			
				h	iere	•   E	stabli	sh the		
						C	alaata	ed record		
						•				
						a	lobal	zero poi	nt	

# Setup

If a AMP multi-core debugging session is set up, start/stop synchronization for the cores is established.

A TR	ACE32 PowerView Core0		
File	Edit View Var Break	Run CPU Misc Trace Perf Cov MPC5XXX Window	Help
н	₩ <b>+ 4 6   F   </b>	🏼   🤋 👀   💿     📰 🖽 🔲 👪 📾 🏜   🎱 🛓 🖉	
	B::SYnch.state		
	- synch	Connect	
	OFF	localhost:10000	Start/stop
	ON	localhost:10001	synchronization
	RESet		between cores/instances
		master slave	
		✓ MasterGo ✓ SlaveGo	
		V MasterBreak	
		✓ MasterStep     ✓ SlaveStep     ✓ SlaveStep	
		Image: MasterSystemMode         Image: SlaveSystemMode	
B::	TRACE32 PowerView C		
			ndow Help
em	N# + 4 6	▶ Ⅱ   🖄   ? №!   💿   🖹 🕮 🔳   🕹 🗟 🛃 👂	<u>i</u> (2)
SF:4			
	B::SYnch.st		
	- synch	Connect	
	OFF	localhost:10001	
	ON     RESet	localhost:10000	
	KESEL		
		MasterGo SlaveGo	
		MasterBreak SlaveBreak	
		✓ MasterStep ✓ SlaveStep	
		✓ MasterSystemMode ✓ SlaveSystemMode	
	B::		
	emulate trigger	devices trace Data other 1	previous
		demo\Acknoledg(stopped by devt2	HLL UP
	C. Troudor Io (Jaemola	terre ( terress), propped by dette	

For trace synchronization the following commands have to be executed:

SYnch.XTrack	localhost:10001	;	in	TRACE32	instance	for	core0
SYnch.XTrack	localhost:10000	;	in	TRACE32	instance	for	corel

The base for the trace synchronization is the tool timestamp or if enabled the Nexus timestamps.

```
Trace.List TIme.Zero DEFault /Track
```

- ; /Track enables here the
- ; time synchronisation to
- ; trace display windows in
- ; other TRACE32 instance

	lero3M - core_1 (z0)							
	Var     Break     Run     CPU     Misc	Trace Probe Perf Cov MPCSXXX Window Help 이   1월 1월 📕 🗟 🛃 🚳 🤹 🖉						
B::Trace.List 9	🔢 [B::Trace.List %TimeFixed TIme.Zero DEFault /Track]							
	Goto Find Cha							
375 -0000792617	ti.zero ru 0.016364910s 0.016366330s	n address cycle data symbol se_mflr r0 e_stmw r30,0x8(r1) ; nProcessorID,8(r1) se_stw r0,0x14(r1) ; r0,20(r1) int nProcessorID = GetProcessorID(); /* get processor ID from PIR e_bl 0x40000112 ; GetProcessorID v:40000112 ptrace \\demo\Global\GetProcessorID mfspr r3,spr286 ; r3,pir se_blr						
-0000792614	0.0163663305	V:40000768 ptrace \\demo\demo\main+0x10 se_mr r30,r3 ; nProcessorID,r3						
377 -0000792612	0.016367105s	<pre>e_add16i r31,r13,-0x7FF0 ; r31,r13,-32752 if (nProcessorID == 0) se_cmpi r30,0x0 ; nProcessorID,0 se_bne 0x40000784 v:40000784 ptrace \\demo\demo\demo\main+0x2C } else {     ""</pre>	Time					
_	]•		synchronization					
B::			between					
	·		TBACE32					
	rigger devices trace	Data Var List PERF SYStem other previous stopped by devt2 HLL UP a	instances					
File Edit View		Trace Probe Perf Cov MPCSXXX Window Help						
[B::Trace.List 9	%TimeFixed TIme.Zero DEFault /Tra	.k] 📃 🗖 🗮 🗮						
		t Profile MIPS						
375 -0000792617 -0000792614	0.016364910s	<pre>se_mflr r0 e_stmw r30,0x8(r1) ; nProcessorID,8(r1) se_stw r0,0x14(r1) ; r0,20(r1) int nProcessorID = GetProcessorID(); /* get processor ID from PIR e_bl 0x40000112 ; GetProcessorID V:40000112 ptrace \\demo\Global\GetProcessorID mfspr r3,spr286 ; r3,pir</pre>						
	0.016366330s 0.016367105s	<pre>se_mflr r0 e_stmw r30,0x8(r1) ; nProcessorID,8(r1) int nProcessorID = GetProcessorID(); /* get processor ID from PIR e_bl 0x40000112 gtrace</pre>						
-0000792617 -0000792614 377	0.016366330s	<pre>se_mflr r0 e_stmw r30,0x8(r1) ; nProcessorID,8(r1) int nProcessorID = GetProcessorID(); /* get processor ID from PIR e_b1 0x4000012 ; GetProcessorID(); /* get processor ID from PIR w:4000012 ptrace</pre>						
-0000792617 -0000792614 377	0.016366330s 0.016367105s	<pre>se_mflr r0 e_stmw r30,0x8(r1) ; nProcessorID,8(r1) int nProcessorID = GetProcessorID(); /* get processor ID from PIR e_bl 0x40000112 gtrace</pre>						
-0000792617 =0000792614 377 -0000792612 B::	0.016366330s 0.016367105s	<pre>se_mflr r0 e_stmw r30,0x8(r1) ; nProcessorID,8(r1) int nProcessorID = GetProcessorID(); /* get processor ID from PIR e_bl 0x40000112 ; GetProcessorID v:40000112 ptrace \\\demo\demo\main+0x10 imfspr r3,spr286 ; r3,pir se_blr v:40000768 ptrace \\demo\demo\main+0x10 se_mm r30,r3 ; nProcessorID,r3 e_add16i r31,r13,-0x7FF0 ; r31,r13,-32752 if (nProcessorID == 0) se_one 0x40000784 ptrace \\demo\demo\main+0x2C } else { imin int int int int int int int int int i</pre>						
-0000792617 -0000792614 377 -0000792612 B:: emulate tr	0.016366330s 0.016367105s	<pre>se_mflr r0 e_stmw r30,0x8(r1) ; nProcessorID,8(r1) int nProcessorID = GetProcessorID(); /* get processor ID from PIR e_bl 0x40000112 gtrace</pre>						

# **Find a Specific Record**

B::Trace.List		
	Goto <mark>♯ Find </mark> M Chart <b>M</b> Pro	file MIPS 🔶 More 🗶 Less
202	for ( regvar = 0; regvar addi r30,r30,0x1 ;	<pre></pre>
-0007906470	P:40000198 ptrace cmpwi r30,0x5 bge 0x400001B8	.L59 \\diabc_int\diabc\func2b+0x24 0.500us ^
-0007906469 204	<pre>P:400001B8 ptrace } lwz r30,0x8(r1);</pre>	Fi Trace Find
-0007906468	D:40007F78 rd-long lwz r31,0x0C(r1) ;	© Expert
-0007906466	D:40007F7C rd-long lwz r0,0x14(r1) ;	Signal      Down     address / expression
-0007906465	D:40007F84 rd-long mtlr r0 addi r1,r1,0x10 ;	- 🔬 🗆 HLL
-0007906463	blr P:40001188 ptrace	Cycle Data
604 -0007906462	/**/ func2c();/**/ bl 0x40000100 ; P:400001D0 ptrace	
		Find Next Find First Find Here Find All Clear Cancel

Example: Find a specific symbol address.

B::Trace.List		
	🖌 Goto  🎁 Find 🛛 🚻 Chart 🖉 🌉 Prof	
record	run address cycle data	symbol ti.back
-0014175413	mtlr r0 addi r1,r1,0x20 ; blr P:40001358 ptrace	r1,r1,32 \\diabc_int\diabc\main+0x220 0.665us
		· · · · · · · · · · · · · · · · · · ·
670	for (j = 0; j < li r31,0x0 cmpwi r31,0x0A ;	10; j++)
	bge 0x4000114C	🛉 Trace Find 📃 🗖 🚾
672	{ sieve(); bl 0x40001388 ;	Expert Occle Group Changes Up     Signal
-0014175412	P:40001388 ptrace	
	char flags[SIZE+1];	address / expression sieve
681	int sieve() { } stwu r1,-0x18(r1) ;	Cycle Data
-0014175411	D:40007F68 wr-long mflr r0	
-0014175409	stw r28,0x8(r1) ; D:40007F70 wr-long	Find Next Find First Find Here Find All Clear Cancel

A more detailed description on how to find specific events in the trace is given in "Application Note for Trace.Find" (app\_trace\_find.pdf).

There are several ways for a belated trace analysis:

- 1. Save a part of the trace contents into an ASCII file and analyze this trace contents by reading.
- 2. Save the trace contents in a compact format into a file. Load the trace contents at a subsequent date into a TRACE32 Instruction Set Simulator and analyze it there.

Saving a part of the trace contents to an ASCII file requires the following steps:

1. Select **Print** in the **File** menu to specify the file name and the output format.

File Edit View Var Break Run C Run Script Edit Script		
Search for Script     Open File     Load File	B::PRinTer	
Type File Ump File Stop Command	© printer	type WIN (Windows Default)
📕 Printer Settings	ClipBoard	ASCIIE (ASCII ENHANCED)
Window Screenshot to File  x exit	@ FILE	FileType ASCIIE (ASCII ENHANCED) C:\T32_MPC\testrun.lst browse

PRinTer.FileType ASCIIE	; specify output format ; here enhanced ASCII
PRinTer.FILE testrun.lst	; specify the file name

2. It only makes sense to save a part of the trace contents into an ASCII-file. Use the record numbers to specify the trace part you are interested in.

TRACE32 provides the command prefix **WinPrint**. to redirect the result of a display command into a file.

```
; save the trace record range (-8976.)--(-2418.) into the ; specified file 
WinPrint.Trace.List (-8976.)--(-2418.)
```

3. Use an ASCII editor to display the result.

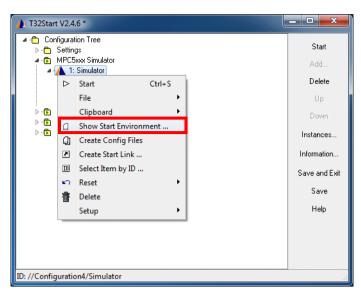
1. Save the contents of the trace memory into a file.

Trace Probe Perf Cov I	MPC			
Configuration CTS Settings				
NEXUS Settings				
🧮 List	•			
📆 Timing	•	(		
🙌 Chart	•	B::Trace.SAVE		
😤 Save trace data		filename		
😤 Load reference data		testrun1		browse
Reset		range		─ mode ──── ◎ BusTrace
		ZIP	Compress	FlowTrace
		Ok		Cancel

The default extension for the trace file is .ad.

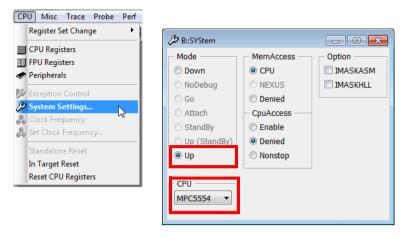
Trace.SAVE testrun1

2. Start a TRACE32 Instruction Set Simulator (PBI=SIM).



🔥 Start Environment Viewer - MPC5xxx Simulator/Simulator	x
Batch Job	
C: cd C:\T32_MPC C:\T32_MPC\bin\windows64\t32mppc.exe -c C:\TMP\T32_1000187.t32	+ +
T32 Configuration File C:\TMP\T32_1000187.t32	
;This configuration file is generated with T32Start2	-
<pre>, ;Environment Variables OS= ID=T32_1000187 TMP=C:\TMP SYS=C:\T32_MPC HELP=C:\T32_MPC\pdf ; Standard License File (client.lic) used ;T32 API Access ; not used ;T32 Intercom</pre>	ш
; not used	
;T32 GDB ; not used	
;Connection to Host PBI=SIM	+
Edit History Settings     Save Batch Job As     Save Config As     Clos	•

3. Select your target CPU within the simulator. Then establish the communication between TRACE32 and the simulator.



#### 4. Load the trace file.

Configuration	B::B::Trace.LOAD *			×
CTS Settings				
Trigger Definition 🔹 🕨	S S S S S S S S S S S S S S S S S S S	mpc55xxdemo	✓ <sup>€</sup> → Search mpc55	ōxxdemo 🔎
List 🕨	Organize 🔻 New folder			= 0
Timing •		A Name	*	
Chart 🕨	🔶 Favorites	A Name		Date modified
Save trace data	📃 Desktop	testrun.a	d	25.08.2011 11:44
Load reference data	👔 🔒 Downloads	=		
Reset	🖳 Recent Places	-		
Keset				
	🕞 Libraries			
	Documents			
	J Music			
	Pictures			
	Videos			
	🛤 Computer	<b>▼</b>	III	•
	File nan	ne: testrun.ad	✓ Current (*.ad)	-
			Open	Cancel
			Open	
			ALL	

#### Trace.LOAD testrun

5. Display the trace contents.

🌽 Setup	📭 Goto 👘 Find 🙌 Chart 🔳 Profile 🔳 MIPS 🗢 More 🗶 Less	
record	run address cycle data symbol	ti.back
-000021	P:20001DD4 ptrace	4.500us 🔺
	and r10,r29,r4	=
	and r9,r28,r3	*
	or. r10,r9,r10	
	slwi r30,r30,0x1 ; r30,r30,1	*
	inslwi r30,r31,0x1,0x1F; r30,r31,1,31	
	slwi r31,r31,0x1 ; r31,r31,1	
000000	beq 0x20001DA0 ; 0x20001DA0 (-)	0.335
-000020	P:20001DA0 ptrace	8.335us
	cmp]w r30,r26	
	slwi r28,r28,0x1 ; r28,r28,1 inslwi r28,r29.0x1.0x1F : r28,r29,1,31	
	inslwi r28,r29,0x1,0x1F ; r28,r29,1,31 slwi r29,r29,0x1 ; r29,r29,1	
	bne 0x20001DB8 ; 0x2001DB8 (-)	
-000018	BRK	
+******	DIKK	-
LOAD		
LUAD		E. ₹
-		

LOAD indicates that the source for the trace information is the loaded file.

6. Load symbol and debug information if you need it.

Data.LOAD.Elf diabc\_ext.x /NoCODE

The TRACE32 Instruction Set Simulator provides the same trace display and analysis commands as the TRACE32 debugger.

Trace-based debugging allows to re-run the recorded program section within TRACE32 PowerView.

If Data Trace Messages were enabled for ReadWrite, it is also possible to watch memory, variable and register changes while re-running the recorded program section.

# **Re-Run the Program**

#### Setup

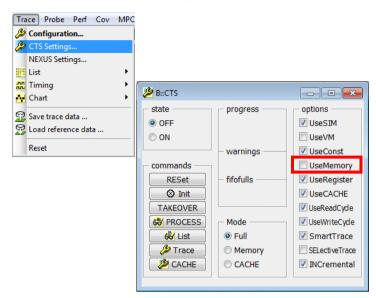
In order to re-run the program, it is sufficient to only enable Branch Trace Messaging. One of the following configurations is suitable:

- BTM ON
- BTM ON + HTM ON
- BTM ON + HTM ON + BL\_HTM ON

B::NEXUS				
nexus OFF OFF ON RESet Trace List	Selection  Selection  Set the set of the set	option POTD STALL OFF SpenDQM SpenDVTM SpenPTM SpenDTM SpenOTM SupprHReshold 1/4	Configuration PortSize MDO12 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE * OFF * CLIENT2 SELECT NONE * MODE OFF *

If you use an OS, it is recommended to also record the task switch information. See "OS-Aware Tracing (ORTI File)" in Training Nexus Tracing, page 193 (training\_nexus.pdf).

Un-check **UseFinalMemory** in the CTS configuration window. A full explanation on this is given later in the chapter "CTS Technique", page 99



#### **CTS.state**

**CTS.UseFinalMemory OFF** 

Specify the starting point for the trace re-run by selecting **Set CTS** from the Trace pull-down menu. The starting point in the example below is the entry to the function **func10**.

🎾 Setup 🛛 🗛 🤇	Goto 🛉	ind 🚺 Chart	Profile	MIPS	More	Less		
record ri	un address		data		/mbol		ti.back	
0001997651	addi ↓ blr ↓ F:40	r1,r1,0x18 0011C8 ptrace	; r	1,r1,24	\diabc\dia	abc\main+0x16C	2.955u	5
621		func1	0();					
355	<pre>bl func10( { stwu</pre>			unc10			Trace R→ Set Ref Z→ Set Zero	
	mflr	r1,-0x58(r1 r0 r14,0x10(r1 r0,0x5C(r1)	) ; v ; r				Toggle Bookmark  Set CTS  View	
		register i, j register v1, register v9,	v2, v3, v4			v16, v17;	List	
360	li stw	v17 = 0; r12,0x0 r12,0x8(r1)	; r	12,0 12.v17(r1)	)		Ignore in Statistic	
361	li cmpwi bge	<pre>for ( i = 0 ;     r31,0x0     r31,0x3     0x400007D0</pre>	i < 3 ; i	++ )			<ul> <li>First in Statistic</li> <li>★ Last in Statistic</li> <li>♦ Full Statistic</li> </ul>	
362	lwz	v17 + r12,0x8(r1)		12,v17(r1)	)		here	-

Selecting Set CTS has the following effect:

• TRACE32 PowerView will use the preceding trace packet as starting point for the trace re-run.

B::Trace.List									×
🔑 Setup	🗘 Goto 🛛 🎁 Fin	d Chart	📕 Profile	MIPS 4	More	Less			
record	run address		data	symb	0]			ti.back	
	lwz mtlr	r0,0x1C(r1) r0		0,28(r1)					Â
	addi • blr	r1,r1,0x18	; r	1,r1,24					-
-0001997651	F:4000	011C8 ptrace		\\di	abc\diabo	:\main+0>	(16C	2.955us	<b>^</b>
621		func10							
		0x40000798	; f	unc10					
355	func10()								
	stwu mflr	r1,-0x58(r1) r0	; r	1,-88(r1)					
	stmw stw	r14,0x10(r1) r0,0x5C(r1)		16,16(r1) 0,92(r1)					
		egister i, j;							• ►

 The TRACE32 PowerView GUI does no longer show the current state of the target system, but it shows the target state as it was, when the starting point instruction was executed. This display mode is called CTS View.

#### CTS View means:

- The instruction pointer is set to the values it had when the starting point instruction was executed. This is done for all cores if an SMP system is under test.
- The content of the core registers is reconstructed (as far as possible) to the values they had when the starting point instruction was executed. This is done for all cores if an SMP system is under test. If TRACE32 can not reconstruct the content of a register it is displayed as empty.
- TRACE32 PowerView uses a yellow look-and-feel to indicate CTS View.
- The **Off** button in the Source Listing can be used to switch off the CTS View.

▲ TRACE32 PowerView for PowerPC         File Edit View Var Break Run CPU Misc Trace Probe Perf Cov MPC5XXX Window Help         ▶ ★ ▲ ↓ ✔ C ▶ II ○ ? ♥? ○ ○ ○ MPC5XXX Window Help         ▶ ★ ▲ ↓ ✔ C ▶ II ○ ? ♥? ○ ○ ○ ▲ ○ ● ? ●         ■ ■ ▲ ↓ ✔ C ▶ II ○ ? ♥? ○ ○ ○ ▲ ○ ● ? ●         ■ ■ ▲ ↓ ✔ C ▶ II ○ ? ♥? ○ ○ ○ ▲ ○ ● ? ●         ■ ■ ▲ ↓ ✔ C ▶ II ○ ? ♥? ○ ○ ○ ▲ ○ ● ? ●         ■ ■ ▲ ↓ ✔ C ▶ II ○ ? ♥? ○ ○ ○ ▲ ○ ● ? ●         ■ ■ ▲ ↓ ✔ C ▶ II ○ Power         ● ■ ▲ ↓ ✔ C ▶ II ○ Power         ● ■ ▲ ↓ ✔ C ▶ II ○ Power         ● ■ ▲ ↓ ✔ C ▶ II ○ Power         ● ■ ● ● ●         ● ■ ▲ ↓ ✔ C ▶ II ○ Power         ● ■ ■ ▲ ↓ ✔ C ▶ II ○ Power         ● ■ ■ ●         ● ■ ■ ●         ● ■ ● ●         ● ■ ■ ●         ● ■ ■ ●         ● ■ ■ ●         ● ■ ■         ● ■ ■         ● ■ ■         ● ■ ■         ● ■ ■         ● ■ ■         ● ■ ■         ● ■ ■         ● ■ ■         ● ■         ● ■         ● ■         ● ■         ● ■         ● ■         ● ■         ● ■         ● ■         ● ■
H K L +        C + II (2) ? K?       III (2) ? K?
Step       Over <ul> <li>Diverge</li> <li>Return</li> <li>Up</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Off</li> <li>Ore</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Ore</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Ore</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Step</li> <li>Over</li> <li>Over</li> <li>Over</li> <li>Entry</li> <li>Ore</li> <li>Over</li> <l< th=""></l<></ul>
Step       Over <ul> <li>Diverge</li> <li>Return</li> <li>Up</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Off</li> <li>Ore</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Ore</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Ore</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Entry</li> <li>Off</li> <li>Step</li> <li>Over</li> <li>Step</li> <li>Over</li> <li>Over</li> <li>Over</li> <li>Entry</li> <li>Ore</li> <li>Over</li> <l< th=""></l<></ul>
addr/line code       label       mnemonic       comment         619       func9();       ox40000710       func9         621       func10();       ox40000798       func10         621       func11(5);       sf:400011C8       4BFFF5D1       bl       ox40000798       ; func10         623       func11(5);       sf:400011C0       38600005       li       r3,0x5       ; r3,5         5F:400011C0       4BFFFA21       bl       ox40000BF0       ; func11       ,         B:Register
619       func9();         619       4BFFF54D         621       func10();         5F:400011C8       4BFFF5D1         623       func11(5);         5F:400011C2       38600005         38600005       1i         r3,0x5       ; r3,5         5F:400011C2       38600005         38600005       1i         r3,0x5       ; r3,5         strinc11       *
SF:400011C4       4BFFF54D       b1       0x40000710       ; func9         621       func10();       0x40000798       ; func10         SF:400011C8       4BFFF5D1       b1       0x40000798       ; func10         623       func11(5);       ;       r3,0x5       ; r3,5         SF:400011C0       4BFFF21       b1       0x40000BF0       ; func11         SF:400011D0       4BFFFA21       b1       0x40000BF0       ; func11         SF:400011C8       R8       0       R16       0       R24       4000259A       S       Stack         R1       40007F80       R9       FFFFFFFF       R17       0       R25       40001B88       E         R2       4000AB70       R10       40000000       R18       0       R26       40001B80       E
SF:400011C8       4BFFF5D1       b1       0x40000798       ; func10         623       func11(5);       ;
SF:400011C8       4BFFF5D1       b1       0x40000798       ; func10         623       func11(5);       ;       ; r3,0x5       ; r3,5         SF:400011C0       38600005       1i       r3,0x5       ; r3,5         SF:400011D0       4BFFFA21       b1       0x40000BF0       ; func11         Image: SF:400011D0       4BFFFA21       b1       0x40000BF0       ; func11         Image: SF:400011C8       R8       0 R16       0 R24       4000259A       S       Stack         Image: R1       40007F80       R9       FFFFFFF       R17       0 R25       40001B88       E         R2       4000AB70       R10       40000000       R18       0 R26       40001B8       E
SF:400011CC       38600005       1i       r3,0x5       ; r3,5         SF:40001100       4BFFFA21       b1       0x400008F0       ; func11         Image: start star
SF:40001100       4BFFFA21       b1       0x40000BF0       ; func11         Image: SF:400011C8       R8       0       R16       0       R24       4000259A       S       Stack         R1       40007F80       R9       FFFFFFFF       R17       0       R25       40001B88       E         R2       4000AB70       R10       40000000       R18       0       R26       40001830       E
B::Register         Image: Constraint of the state
RO         400011C8         R8         0         R16         0         R24         4000259A         S         Stack           R1         40007F80         R9         FFFFFFF         R17         0         R25         40001D88           R2         4000AB70         R10         40000000         R18         0         R26         40001830
R0         400011C8         R8         0         R16         0         R24         4000259A         S         Stack           R1         40007F80         R9         FFFFFFF         R17         0         R25         40001D88           R2         4000AB70         R10         40000000         R18         0         R26         40001830
R2 4000AB70 R10 4000000 R18 0 R26 40001830
R4 0 R12 R20 0 R28 1
R5 0 R13 4000C010 R21 0 R29 0C R6 0 R14 0 R22 40001960 R30 0
R7 0 R15 0 R23 0E56 R31 0B
TBL 150A8504 XER 0 CR 20304404 LR 400011C8
TBU F31A9668 USPRG05A242EC9 CTR D1 IP 400011C8
B::
components trace Data Var List PERF other previous
SF:400011C8 \\diabc\diabc\diabc\main+0x16C (CTS) CTS (-1997651.0.)

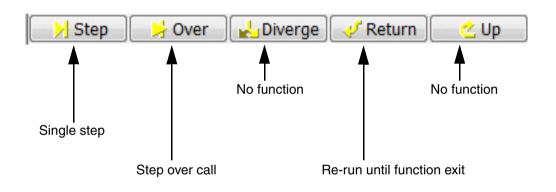
B::
components trace Data Var List PERF
SF:400011C8 \\diabc\diabc\main+0x16C (CTS) CTS (-1997651.0.)

TRACE32 PowerView displays the state of the target as it was when the instruction of the trace record -1997651.0 was executed

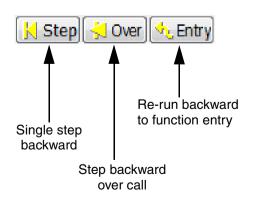
Now you can start to re-run the recorded program section within TRACE32 PowerView by forward or backward debugging.

Forward debugging com	mands	Backward debugging commands
🔄 [B::List]		
Step Vver Diverge	🞺 Return 📃 🕐 Up	📕 Step 👫 Over 🔩 Entry 💶 Off 🔛 😹
addr/line_code labe	l mnemonic	comment
621 SF:400011C8 48FFF5D1	func10(); bl 0x4	0000798 ; func10
623	func11( 5 );	

### **Forward Debugging**



#### **Backward Debugging**



This feature only makes send for the IEEE-ISTO 5001-2008 and the IEEE-ISTO 5001-2012 standard.

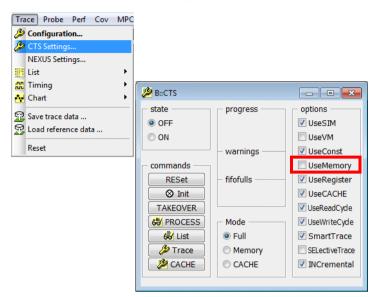
# Setup

In order to re-run the program and watch the variables, the following Nexus setups are recommended:

- Enable Branch Trace Messaging (BTM ON / BTM ON + HTM ON / BTM ON + HTM ON + BL\_HTM ON)
- Enable Data Trace Messages for read/write accesses, but suppress Data Trace Messages on overflow threat.

; Configuration example	
NEXUS.BTM ON	
NEXUS.HTM ON	
NEXUS.BL_HTM ON	
NEXUS.DTM ReadWrite	
NEXUS.SupprTHReshold 3/4	; Advise Nexus to suppress specified ; messages when Nexus FIFO is 3/4 filled
NEXUS.SpenDTM ON	; Advise Nexus to suppress Data Trace ; Messages when the specified filling ; level is reached

Un-check **UseMemory** in the CTS configuration window. A full explanation on this is given later in the chapter "CTS Technique", page 99.



#### **CTS.state**

**CTS.UseMemory OFF** 

Specify the starting point for the trace re-run by selecting **Set CTS** from the Trace pull-down menu. The starting point in the example below is the read access to the variable **mstatic1** in function **func2d**.

[B::Trace.List]									×
🖉 Setup 🔒 🕻	Goto	Chart	Profile	MIPS	♦ More	Le	SS		
	un address	cycle	data	s	ymbol	_		ti.back	
-0011009066	F:4000	0x400002A8 02A8 ptrace	;	func2d	\diabc\dia	abc\fur	ic2d	0.865us	A E
219	void funca {			-1 10(-1)					*
-0011009065	stwu D:40007 mflr	r1,-0x10(r1) 7F70 wr-long r0		40007F80 \	\diabc\Glo	bal\	SP_TEST+0x578	0.615us	
-0011009063	stw D:40007	r30,0x8(r1) 7F78 wr-long r31.0x0C(r1)		regvar,8(r 4000402C \ autovar,12	\diabc\Glo	bal\	SP_TEST+0x580	0.740us	
-0011009061	D:40007	7F7C wr-long		000000A \	\diabc\Glo	bal\	SP_TEST+0x584	0.740us	
-0011009060		r0,0x14(r1) 7F84 wr-long uto short aut					SP_TEST+0x58C	0.615us	
	re	egister short	t regvar;	/* short	register	variab	ile */		
223	a	utovar = reg	/ar = mst	atic1;					
	WZ	r30,-0x7FB8		regvar,mst	atic1(r13)	)	-		
-0011009058	D:40004	1058 rd-long r31,r30		2843043E \ autovar,re			Trace Set Ref	0.740us	
224	addi a	utovar++; r31,r31,0x1		autovar.au	-	z	Set Zero		
	dddr	1 51,1 51,011	3	uucovar , ao	covar ji		Toggle Bookmark		-
						8	Set CTS	,	h. d
						Q	View		
							List		
						<u> </u>	Chart		
							Ignore in Statistic		
							Use in Statistic		
							First in Statistic		
						±	Last in Statistic		

Full Statistic here

۲

Selecting Set CTS has the following effect:

• TRACE32 PowerView will use the preceding trace packet as starting point for the trace re-run.

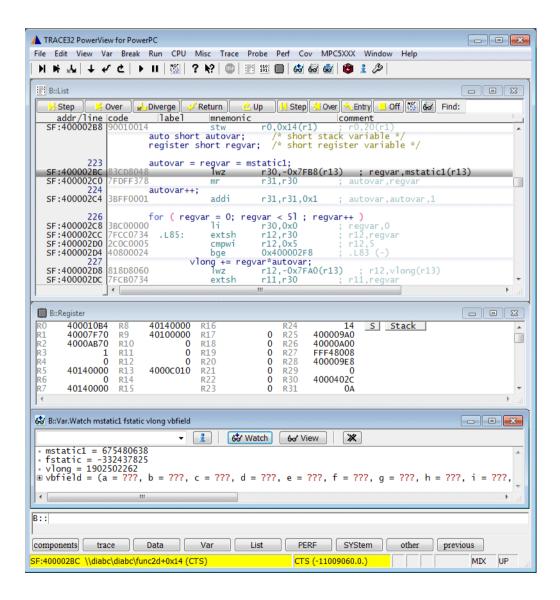
& Setup	Coto	# Find	Chart	Profile		♠ More	T Less	1		
<u> </u>		address				vmbol	LC35	J	ti.back	<u> </u>
record					autovar,12					
-0011009061		D:40007F	7C wr-long		000000A \	\diabc\Glo	obal\SP.	_TEST+0x584	0.740us	
0011000000	5		0,0x14(r1)	;	r0,20(r1)	\		TECT: 0ERC		+
-0011009060			84 wr-long	ovar:	/* short	Valabc Val	ppar\sP	_TEST+0x58C	0.615us	
					/* short					
								·		
223			ovar = regv							
-0011009058		wz r	50,-0X/FB80	(r13) ;	regvar,mst 2843043E	diabc/di	) abc\mstat:	ic1	0.740us	•
0011005050		r r	31,r30	:	autovar, re	dvar	abe (instat		0.74003	
224		auto	ovar++;			-				
	3	addi r	31,r31,0x1	;	autovar, au	tovar,1				
226		for	( negyar -	- 0. rem	/ar < 51 ;	requart.	)			
220	1				regvar,0	regvarre ,	,			
	6	extsh ri	L2,r30		r12, regvar					
			L2,0x5		r12,5					
227		oge O:	x400002F8		.L83 (-) ar*autovar;					
221	1	wz r	120x7FA0	r= regva	r12,vlong(	r13)				-
				(125) )	T LE   T Tong (					

The TRACE32 PowerView GUI does no longer show the current state of the target system, but it shows the target state as it was, when the starting point instruction was executed. This display mode is called CTS View.

•

#### CTS View means:

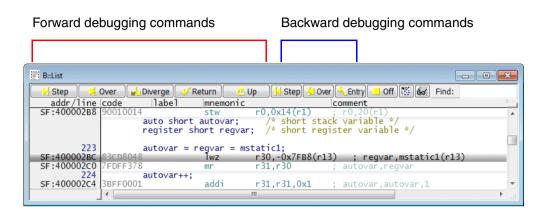
- The instruction pointer is set to the values it had when the starting point instruction was executed. This is done for all cores if an SMP system is under test.
- The content of the core registers is reconstructed (as far as possible) to the values they had when the starting point instruction was executed. This is done for all cores if an SMP system is under test. If TRACE32 can not reconstruct the content of a register it is displayed as empty.
- The contents of the variables changed by the recorded program section are reconstructed (as far as possible) to the values they had when the starting point instruction was executed. If TRACE32 can not reconstruct the content of a variable ??? are displayed.
- TRACE32 PowerView uses a yellow look-and-feel to indicate CTS View.
- The **Off** button in the Source Listing can be used to switch off the CTS View.



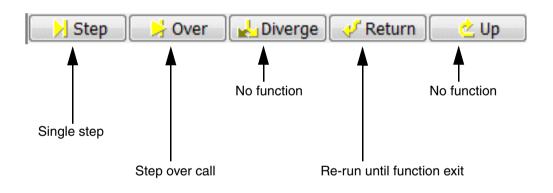
B::	
components trace Data Var	List PERF SYStem
SF:400002BC \\diabc\diabc\func2d+0x14 (CTS)	CTS (-11009060.0.)
	•

TRACE32 PowerView displays the state of the target as it was when the instruction of the trace record -11009060.0 was executed

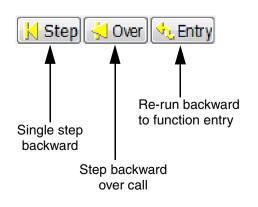
Now you can start to re-run the recorded program section within TRACE32 PowerView by forward or backward debugging.



### **Forward Debugging**



#### **Backward Debugging**



The technology used for Trace-Based Debugging allows additionally to display a full HLL trace.

69 69 69	Configuration CTS Settings NEXUS Settings			<b>1</b> 2
	List			Default
ŝ	Timing	≁		All
A.	Chart	►	₽	Tracking with Source
9 P	Load reference data		<del>&amp;</del>	List Context Tracking System
	Reset			

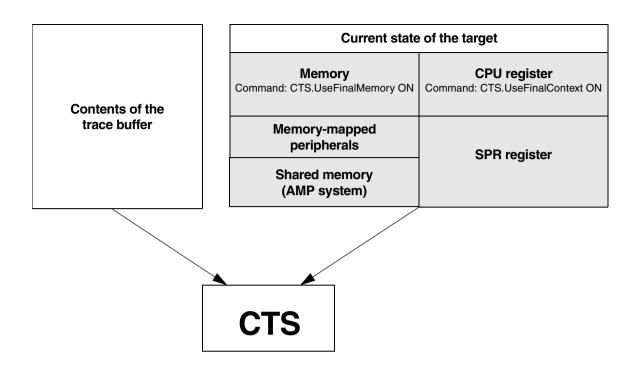
B::CTS.List	CTS   Goto   🎁 Find   🛒 TREE   😽 Chart   👾 Chart   🜩 More   🗶 Less	
record		
0000100195	<pre>vdouble = 1.60000000000001 regvar = 0.0 autovar = 1.081815803e+9</pre>	- 1.913us
215 0000100192	<pre>regvar = 0.0</pre>	— 21.340us
214 0000100147	<pre>for ( regvar = 0.0; regvar &lt; 5.0 ; regvar += 1.0 ) regvar = 1.0 vdouble = 1.600000000000001 regvar = 1.0 autovar = 1.081815803e+9</pre>	— 11.207us
215 0000100125	vdouble += regvar*autovar; vdouble = 1.0818158045999999e+9 regvar = 1.0	— 49.452us
214	for ( regvar = 0.0; regvar < 5.0 ; regvar += 1.0 ) regvar = 2.0	34.262us

For each HLL step the following information is displayed:

- The values of the local and global variables used in the HLL step
- The result of the HLL step
- The time needed for the HLL step

# **CTS.List**

List pure HLL trace.



CTS reads and evaluates the current state of the target together with the information recorded to the trace memory by default.

The following commands are used to configure CTS properly:

CTS.UseFinalMemory ON Default setting within TRACE32

If **CTS.UseFinalMemory** is ON and TRACE32 detects that a memory address was not changed by the recorded program section, TRACE32 PowerView displays the current content of this memory in CTS display mode.

- If Data Trace Messaging is disabled (NEXUS.DTM OFF), TRACE32 can not detect which memory content was changed. This is the reason why CTS.UseFinalMemory has to be set to OFF.
- If Data Trace Messaging is enabled (NEXUS.DTM ReadWrite) it is not guaranteed, that all read/write accesses are recorded. This is the reason why CTS.UseFinalMemory has to be set to OFF.

Please be aware, that CTS ignores all read/write cycles that can not be assigned to its instruction (displayed in red).

🔑 Setup 🚺	🕽 Goto	Find	Chart	📕 Profile	MIPS	More Less		
record	run add	ress	cycle	data		symbol	ti.ba	ack
	add add	li r1,	r1,0x10	; r	1,r1,16			
0003731312 0003731310		D:40004070 D:40004070	rd-long	E	45DBAF7	\\diabc\Global\vlon \\diabc\Global\vlon	ğ 0.	. 615us . 990us
0003731308 0003731306		D:40004070 D:40004070		6	88E5C0E	\\diabc\Global\vlon \\diabc\Global\vlon	g 0. g 0.	.740us .740us

MAP.VOLATILE <range> Declare specified address range as volatile.

CTS supposes by default that memory is only written by the core(s) for which trace information is recorded into the trace memory. But other bus master such as the DMA controller or other, not recorded cores, can change memory too. And external interfaces can change memory mapped peripheral registers.

All memory ranges, that are not only changed by the core(s) for which trace information is recorded, have to be excluded from the CTS memory/variable reconstruction.

MAP.VOLATILE 0xF0000000-0xFFFFFFFF	; exclude peripheral register ; address space from the CTS ; reconstruction
MAP.VOLATILE 0x400180000x4001BFFF	; exclude memory that is ; changed not only by the ; recorded core(s) ; from the CTS reconstruction

If Data Trace Messaging is disabled (NEXUS.DTM OFF) and **CTS.UseFinalMemory** is switch OFF, but your target memory contains constants, you can configure TRACE32 to use these constants for the CTS reconstruction by the following commands:

MAP.CONST <address\_range> CTS.UseConst ON

CTS.UseFinalContext ON Default setting within TRACE32

If **CTS.UseFinalContext** is ON and TRACE32 detects that a register was not changed by the recorded program section, TRACE32 PowerView displays the current content of this register in CTS View mode.

CTS.UseFinalContext has to be set to OFF, if you used Stack mode for tracing recording.

Break Run CPU Misc Tr Set Bist Ist Implementation	B::Break.Set	sion -		
Trigger Bus	address / expres	ssion		▼ <u>∡</u> 🗆 HLL
Trigger Reset	type Program ReadWrite Read Write default Ok	options EXclude NOMARK DATA Add	Temporary DISable DISableHIT LISableHIT LISableHIT LISableHIT	action stop Spot Alpha Beta Charly Delta
				Echo WATCH TraceEnable TraceON TraceOFF TraceTrigger BusTrigger BusCount

TraceEnable, TraceData, TraceON and TraceOFF are so-called filters. **Filters** can be used advise the NEXUS module to generate trace information only for events of interest.

**TraceEnable:** Advise the NEXUS module to generate trace messages only for the specified instruction(s) or read/write accesses.

**TraceData:** Advise the NEXUS module to generate trace messages for all executed instructions and for the specified read/write accesses.

TraceON: Advise the NEXUS module to start the generation of trace messages at the specified event.

TraceOFF: Advise the NEXUS module to stop the generation of trace messages at the specified event.

TraceTrigger, BusTrigger and BusCount are so-called triggers. **Triggers** can be used to advise the NEXUS module to signal the occurrence of an event. TRACE32 can react on this occurrence by stopping the trace recording, by counting the event ....

TraceTrigger: Stop the trace recording at the specified event.

BusTrigger: Generate a pulse on the trigger bus at the specified event.

**BusCount:** Count the specified event.

The MPC5xxx provides the following resources for filter and trigger:

• Data Trace Control Register (DTC): to filter Data Trace Messages (2-4 address ranges)

🗢 B::NEXUS.Register	×
⊞ <u>Nexus Port Controller (NPC)</u>	_
□ <u>Core Trace Control and Nexus Registers</u>	_
⊞ General Trace Control Registers	_
⊞ Watchpoint Trigger Register	=
□ <u>Data Trace Control Register</u> DTC <b>C0000000</b> RWT1 <b>DRWT</b> RWT2 <b>dis</b> RC1 <b>inr</b> RC2 <b>inr</b> DI1 data DI2 data	-
DTSA1 0000000 DTSA2 0000000	
DTEA1 FFFFFFF DTEA2 0000000	Ŧ
	۲. «

NEXUS.Register

Watchpoint Trigger Register: to activate a trace action on a specified event. The source for the specified event are the Watchpoints that are also used for the on-chip breakpoints.

Core type:	On-chip Breakpoints	Instruction Address Breakpoints	Data Address Breakpoints	Data Value Breakpoints	
e200z0 e200z0h	4 instruction 2 read/write no counters	4 single breakpoints or 2 breakpoint ranges	2 single breakpoints or 1 breakpoint range	none	
e200z0Hn3	4 instruction 2 read/write 2 data value no counters	4 single breakpoints or 2 breakpoint ranges	2 single breakpoints or 1 breakpoint range	2 single breakpoints (associated with data address BPs)	
e200z1 e200z3 e200z6 e200z650 e200z750	4 instruction 2 read/write 2 counters	4 single breakpoints or 2 breakpoint ranges	2 single breakpoints or 1 breakpoint range	none	

Core type:	On-chip Breakpoints	Instruction Address Breakpoints	Data Address Breakpoints	Data Value Breakpoints
e200z335	4 instruction 2 read/write 2 data value 2 counters	4 single breakpoints or 2 breakpoint ranges	2 single breakpoints or 1 breakpoint range	2 single breakpoints (associated with data address BPs)
e200z446 e200z4d e200z760	8 instruction 2 read/write 2 data value 2 counters	8 single breakpoints or 2 breakpoint ranges and 4 single breakpoints	2 single breakpoints or 1 breakpoint range	2 single breakpoints (associated with data address BPs)
e200z210 e200z215 e200z225 e200z420 e200z425 e200z720 e200z4201 e200z4203 e200z4204 e200z4251 e200z7260	8 instruction 4 read/write 2 data value no counters	8 single breakpoints or 4 breakpoint ranges	4 single breakpoints or 2 breakpoint ranges	2 single breakpoints (associated with data address BPs)

• The **MPC57xx** provides also means to control Program Trace Messaging and Data Trace Messaging from the application.

# Nexus Development Control Register:

PTMARK Bit 1	Program Trace Messaging when PMM bit is set
DTMARK Bit 1	Data Trace Messaging when PMM bit is set

#### Machine Status Register:

PMM Bit	Performance monitor mark bit.
	PMM Bit 1, PTMARK Bit 1 -> Program Trace Messaging is enabled
	PMM Bit 1, DTMARK Bit 1 -> Data Trace Messaging is enabled

The table below summarizes the influence of the filter/ trigger on the messaging.

		DTM	DTM	0714	DOM
	WTM Watchpoint Trace Messages	<b>BTM</b> Branch Trace Messages	<b>DTM</b> Data Trace Messages	<b>OTM</b> Ownership Trace Messages	<b>DQM</b> Data Acquisition Messages
TraceEnable on single instruction	Watchpoint Hit Message for instruction	Disabled	Unaffected	Unaffected	Unaffected
TraceEnable on instruction range	Unused	Filter applies	Filter applies	Unaffected	Unaffected
TraceEnable on read/write access	Unused	BTM disabled	DTM enabled	Unaffected	Unaffected
			Filter applies		
TraceData	Unused	Unaffected	DTM enabled	Unaffected	Unaffected
			Filter applies		
Global TraceON/ TraceOFF	Unused	Filter applies	Filter applies	Unaffected	Unaffected
Program TraceON/	Unused	BTM enabled	Unaffected	Unaffected	Unaffected
TraceOFF		Filter applies			
Data TraceON/ TraceOFF	Unused	Unaffected	Filter applies	Unaffected	Unaffected
TraceTrigger BusTrigger BusCount	WHM for instruction or data address/da ta value	Unaffected	Unaffected	Unaffected	Unaffected

# **Examples for TraceEnable on Instructions**

### Resource: Watchpoints

#### Controlled message types

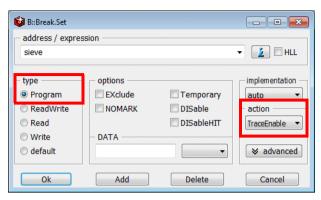
WTM Watchpoint Trace Messages	<b>BTM</b> Branch Trace Messages	<b>DTM</b> Data Trace Messages	<b>OTM</b> Ownership Trace Messages	<b>DQM</b> Data Acquisition Messages
Watchpoint Hit Message(s) is generated for the specified instruction(s)	Disabled	Unaffected	Unaffected	Unaffected

Disable message types, that are unaffected by the filter and not required for your analysis.

nexus     OFF     ON	- selection BTM	Option Option Option Option Option	PortSize	CLIENT1
RESet		OFF •	PortMode	- MODE
Prace List	DQM DTM OFF	suppression SpenDQM SpenWTM	DDR	- CLIENT2 - SELECT NONE
	– PTCM PID_MSR BL_HTM	SpenPTM SpenDTM SpenOTM		OFF +
	TLBNEW	- SupprTHReshold		

**Example 1:** Advise the NEXUS module to generate only trace information for the entries to the function sieve.

1. Set a Program breakpoint to the start address of the function sieve and select the action TraceEnable.



- 2. Start the program execution and stop it.
- 3. Display the result.

B::Trace.Li	st NEXUS DEFault		
Setup record	📭 Goto 🎁 Find 🕂 Chart 📕 Profile nexus	run address cycle data symbol	ti.back
-00000017	TCODE=OF SPI=O WHM S=0001	stwu r1,-0x18(r1) ; r1,-24(r1) 	89.480us
	char flags[SIZE+1];		*
681	int sieve() {	<pre>/* sieve of erathostenes */</pre>	
-00000016	TCODE=0F SPI=0 WHM S=0001	TRACE ENABLE C:40001388 ptrace //diabc_int/diabc/sieve	89.340us
	char flags[SIZE+1];		
681	int sieve() { ∢	/* sieve of erathostenes */	• •

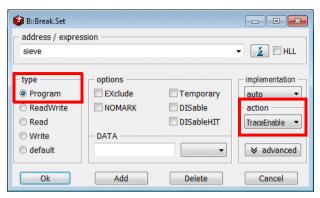
The following **Trace.STATistic** command calculates the time intervals for a program address event. The program address event is here the entry to the function sieve:

Trace.STATistic.AddressDIStance sieve

Ξ	B::Trace.STATis	tic.AddressDIS	Stance siev	e								×
	🖢 Setup 🛛 📗	Chart 🗧 🖨	Zoom	XZ	oom	A Move	🛛 🐺 Mov	e				
		samp]es:		59.	avr:	33.599				95.507ms		
		total:	2.30	)0s	in:	1.982	s out:	318.026ms	ratio:	86.174%		
	up to	count	ratio	<b>b</b>	1%	2%	5%	10%	20%	50%	100	
<	0.000us	0	). 0.0	000%								
	41.943ms	54									-	
	83.886ms			000%								
	125.829ms			00%								
	167.772ms			00%								
	209.715ms			00%								
	251.658ms			00%								
	293.601ms			000%								E
	335.544ms 377.487ms			00%								
	419.430ms			74%								
	461.373ms			000%								
	503.316ms			000%								
	545.260ms			000%								
	587.203ms			000%								
	629.146ms			000%								
	671.089ms			00%								-
		4		110000							6555777	Þ
1		4										

**Example 2:** Advise the NEXUS module to generate trace information for the entries to the function sieve and for the exits of the function sieve.

1. Set a Program breakpoint to the start address of the function sieve and select the action TraceEnable.



2. Set a Program breakpoint to the exit address of the function sieve and select the action TraceEnable.

📦 B::Break.Set			- • •
address / expres			▼ 🗾 🗆 HLL
<ul> <li>type</li> <li>Program</li> <li>ReadWrite</li> <li>Read</li> <li>Write</li> <li>default</li> </ul>	options	Temporary DISable DISableHIT	implementation auto action TraceEnable
Ok	Add	Delete	Cancel

**sYmbol.EXIT**(*<symbol>*) Returns the exit address of the specified function

3. Start the program execution and stop it.

4. Display the result.

B::Trace.Lis	it			×
		ata symbol	ti.back	
-00000015	C:40001388 ptrace	\\diabc_int\diabc\sieve	1.680us	
	<pre>char flags[SIZE+1];</pre>			*
681	int sieve() { stwu r1,-0x18(r1)	<pre>/* sieve of erathostenes */ : r124(r1)</pre>		
-00000014	TRACE ENABLE C:40001444 ptrace blr TRACE ENABLE	\\diabc_int\diabc\sieve+0xBC	87.820us	-
-00000013	C:40001388 ptrace	\\diabc_int\diabc\sieve	1.680us	
	<pre>char flags[SIZE+1];</pre>			_
681	int sieve() { stwu r1,-0x18(r1)	<pre>/* sieve of erathostenes */ ; r1,-24(r1)</pre>		
	TRACE ENABLE		I	►

The following **Trace.STATistic** command calculates the time intervals between two program address events A and B. The entry to the function sieve is A in this example, the exit from the function is B.

Trace.STATistic.AddressDURation sieve sYmbol.EXIT(sieve)

ø	Setup]	Chart	<b>\$</b> Zo	oom 🚺	Zo	om ]	1 N	love	🗧 Mo	ve	]					
		samples	:	30		avr:		745u			87.660us	max:		840us		
		total:		1.332s		in:	2.	632m	s out	•	1.330s	ratio	b: 0	.197%		
	up to	count		ratio	1	.%	2%		5%		10%	20%		50%	100	L
	87.660us	l	0.	0.000												1
	87.680us		15.	50.000	1%  -											
	87.700us		0.	0.000												
	87.720us		0.	0.000												
	87.740us		0.	0.000												
	87.760us		0.	0.000												
	87.780us		0.	0.000												
	87.800us		0.	0.000												
	87.820us 87.840us		0. 8.	0.000												
	87.860us		7.	23.333												
	87.880us		ó.	0.000												
	87.900us		ŏ.	0.000												
	87.920us		ŏ.	0.000												
	87.940us		0.	0.000												
	87.960us		0.	0.000												
	87.980us		0.	0.000	1%											
			0.	0.000												

Resource: Watchpoints, limited to one instruction address range

## Controlled message types

WTM	ВТМ	DTM	ОТМ	DQM
Unused	Filter applies if BTM is enabled	Filter applies if DTM is enabled	Unaffected	Unaffected

Enable BTM. This filter requires that Branch History messaging is disabled.

Enable DTM if you are interested in the read/write accesses performed by the specified instruction address range.

Disable message types, that are unaffected by the filter and not required for your analysis.

**Example:** Advise the NEXUS module to generate trace information for all taken branches within the function func9.

1. Enable Branch Trace messaging, but don't enable Indirect Branch History messaging.

Disable Data Trace messaging.

B::NEXUS				
nexus ○ OFF ● ON ■ RESet ■ List	selection BTM HTM OTM OTM DQM DTM OFF  PID_MSR BL_HTM TLBNEW TLBINV	option POTD STALL OFF Suppression SpenDQM SpenVTM SpenVTM SpenDTM SpenOTM SuprTHReshold 1/4	Configuration PortSize PortMode 1/2 DDR	CLIENT1 SELECT MODE OFF CLIENT2 SELECT NONE MODE OFF

2. Set a Program breakpoint to the complete address range of the function func9 (HLL check box ON) and select the action TraceEnable.

📦 B::Break.Set			
– address / expre	ssion		
func9			🕶 📝 🗹 HLL
- type	– options –		-implementation
Program	EXclude	Temporary	auto
ReadWrite	NOMARK	DISable	- action
© Read		DISableHIT	TraceEnable 🔻
Write	DATA		
Ø default		<b></b>	Sector Secto
L			
Ok	Add	Delete	Cancel

3. Start the program execution and stop it.

	📭 Goto 🎁 Find 🚰 Chart 🔳 Profile 🔳 MIPS 🔷 More 🗶 Less		x
record	run address cycle data symbol	ti.back	
154	<pre>static void func1( intptr ) /* static function */ int * intptr;</pre>		* =
154	1 stwu r1,-0x10(r1) ; r1,-16(r1)		
-00000059	TRACE ENABLE     P:4000004C ptrace     \\diabc_int\diabc\func1	3.680us	_
	<pre>static void func1( intptr ) /* static function */ int * intptr;</pre>		
154	{   stwu r1,-0x10(r1) ; r1,-16(r1) 		
-00000057	P:40000744 ptrace /\diabc_int\diabc\func9+0x34 cmpw r30,r31 ; reg2,reg1	3.820us	
-00000056	bge 0x40000774 ; .L155 (-) P:40000774 ptrace \\diabc_int\diabc\func9+0x64	0.680us	
334	for ( reg1 = 0 ; reg1 < 2 ; reg1++ ) + addi		
-00000055	b 0x40000730 ; .L158 P:40000730 ptrace \\diabc_int\diabc\func9+0x20	0.500us	
	cmpwi r31,0x2 ; reg1,2 bge 0x4000077C ; .L152 (-)		Ŧ
_	] 4	)	н. *

# Resource: DTC Register

# Controlled message types

WTM	втм	DTM	ОТМ	DQM
Unused	BTM is disabled by filter	DTM is enabled by filter	Unaffected	Unaffected
		Filter applies		

Disable message types, that are unaffected by the filter and not required for your analysis.

B::NEXUS				
Nexus OFF ON RESet	Selection  V BTM  HTM OTM  DM DM OFF  PTCM  PTCM  BL_HTM  TLBNEW  TLBINV	option POTD STALL OFF SpenDQM SpenDVTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4	Configuration PortSize MDO12 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE * OFF * CLIENT2 SELECT NONE * MODE OFF *

**Example:** Disable Branch Trace messaging and advise the NEXUS module to only generate trace information for the write accesses to the variable flags[3].

1. Set a Write breakpoint to the variable flags[3] and select the action TraceEnable

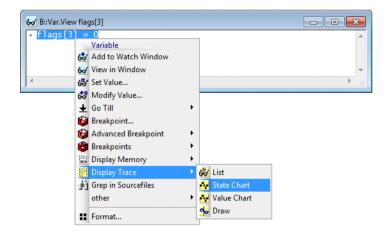
🔯 B::Break.Set			
address / expres	ssion		
flags[3]			🔻 🗾 🗹 HLL
- type	options		implementation —
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	TraceEnable 💌
Write	DATA		
O default		<b>•</b>	
L			
Ok	Add	Delete	Cancel

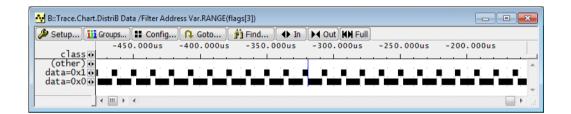
- no data value possible (limitation of DTC Register)
- accessing instruction not possible (limitation of DTC Register)
- 2. Start the program execution and stop it.
- 3. Display the result.

B:: Trace.List NEXUS VAR DEFault			- • •
🥔 Setup 🔼 Goto 🎽 Find 🦳 👬 Chart 📜 Profile 🛛 🧮 MIPS 🗎 🖨 More 🗶 Less			
record nexus	var run address cycle data	symbol	ti.back
-00000027 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=000000000000000000000		01 \\diabc_int\Global\flags+0x3	67.500us 🔺
-00000026 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=000000000000000000000		00 \\diabc_int\Global\flags+0x3	22.000us 🖃
-00000025 TCODE=05 SPI=0 DT-DM4 S=0000 A=00000000 V=000000000000001 1		01 \\diabc_int\Global\flags+0x3	67.500us 🗒
-00000024 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=000000000000000000000		00 \\diabc_int\Global\flags+0x3	21.820us
-00000023 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=0000000000000001		01 \\diabc_int\Global\flags+0x3	67.500us 🔺
-00000022 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=000000000000000000000		00 \\diabc_int\Global\flags+0x3	22.000us
-00000021 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=000000000000000000000		01 \\diabc_int\Global\flags+0x3	67.500us 👝
-00000020 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=000000000000000000000		00 \\diabc_int\Global\flags+0x3	21.840us 💷
-00000019 TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=0000000000000001	Flags[3] = 1 D:40005623 wr-byte	01 \\diabc_int\Global\flags+0x3	67.500us 🔻
4			E al

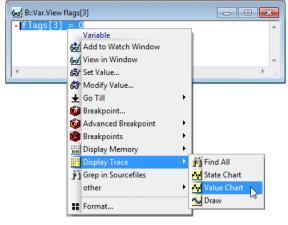
The Variable pull-down provides various way to analyze the variable contents over the time.

```
; open a window to display the variable Var.View flags[3]
```



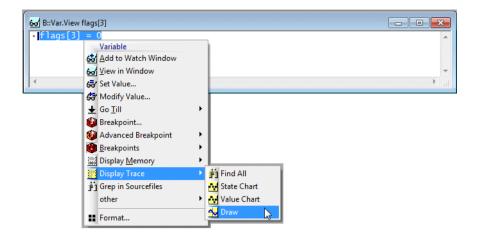


Display the value changes of a variable graphically Trace.Chart.DistriB Data /Filter Address Var.RANGE(<var>)



🕂 B::Trace.Chart.VarState /Filter Address \	/ar.RANGE(flags[3])	
Setup Config Coto 400000s range	Pind         M Chart         M In         M Out         KN Full           -1.047380000s         -1	-1.047360000s
flags[3] () 11 0		
► 4 III ► 4		

Display variable contents over the time numerically Trace.Chart.VarState



B::Trace.DRAW.Var %E	)EFault flags[3]				- • •
🌽 Setup 🔃 Goto	. 👘 Find	Chart	♣ In ► Out	KN Full 🔷 In 🗶 Out 引	Full
-	854.700ms	- 854	4.600ms	-854.500ms	-854.400m
			1		· · · · · · · ·
0.					

Display variable contents over the time graphically **Trace.DRAW.Var %DEFault** <*var>* 

# Resource: DTC Register

# Controlled message types

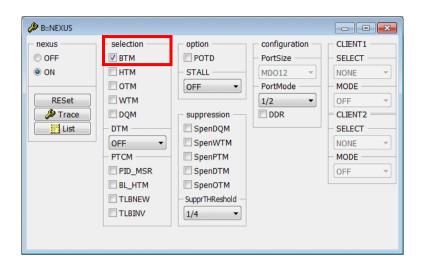
WTM	втм	DTM	ОТМ	DQM
Unused	Unaffected	DTM is enabled by filter	Unaffected	Unaffected
		Filter applies		

Disable message types that are unaffected by the filter and not required for the analysis.

B::NEXUS				
nexus OFF ON RESet Trace List	Selection BTM HTM OTM DQM DTM OFF PTCM PID_MSR BL_HTM TLBNEW TLBINV	option POTD STALL OFF Suppression SpenDQM SpenWTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4	Configuration PortSize MDO12 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE * OFF * CLIENT2 SELECT NONE * MODE OFF *

**Example:** Advise the NEXUS module to generate trace information for the write accesses to flags[12] and to generate trace information for all executed instructions.

1. Enable Branch Trace messaging.



2. Set a Write breakpoint to the variable flags[12] and select the action TraceData.

🙆 B::Break.Set			
address / expres	ssion		
flags[12]			🔻 👔 🗸 HLL
·			
type	options		implementation —
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
© Read		DISableHIT	TraceData 💌
Write	- DATA		
default		<b>•</b>	Sector Secto
L			
Ok	Add	Delete	Cancel

- 3. Start the program execution and stop it.
- 4. Display the result.

B::Trace.Lis	st NEXUS DEFault				
	📭 Goto ) 🎁 Find ) 🛃 Chart ) 🌃 Profile ) 📓 MIPS ) 🖨 More 🗶 Less				
record	nexus	run address	cycle data		ti.back
		addi li stbx	r12,r12,0x5620 r11,0x0 r11,r12,r29	; r12,r12,22048 ; r11,0 ; r11,r12,k	Ē
698	k += primz;	add b	r29,r29,r30 0x400013F8	; k,k,primz ; .L533	<b>^</b>
-06166088	TCODE=05 SPI=0 DT-DWM S=0000 A=00000000 V=000000000000000000000		562C wr-byte 13F8 ptrace	00 \\diabc_int\Global\flags+0x0C \\diabc_int\diabc\sieve+0x70	1.160us 0.340us
695	while ( k <= SIZE )	cmpwi bgt	r29,0x12 0x40001418	; k,18 ; .L532 (-)	
697	t flags[k] = FALSE;	lis addi li	r12,0x4000 r12,r12,0x5620 r11,0x0	; r12,16384 ; r12,r12,22048 ; r11,0	
698	k += primz;	stbx add	r11,r12,r29 r29,r29,r30	; r11,r12,k ; k,k,primz	-

Please be aware that in the case of a TraceData filter a correlation of the data access and the instruction is in most cases not possible.

# **Global TraceON/Trace OFF**

**Resource:** Watchpoints

## Controlled message types

WTM	втм	DTM	ОТМ	DQM
Unused	Filter applies	Filter applies	Unaffected	Unaffected

Enable Branch Trace Messaging and Data Trace Messaging if this information is required for your analysis.

Disable messages types that are unaffected and not required for the analysis.

nexus	<ul> <li>selection ——</li> </ul>	- option	<ul> <li>configuration</li> </ul>	CLIENT1
OFF	🗷 ВТМ	SmartTrace	- PortSize	- SELECT
ON	🔲 НТМ	POTD	MDO4 -	NONE -
	🔲 ОТМ	- STALL	- PortMode	- MODE
RESet	🗆 WTM	OFF 🔻	1/2 🔻	OFF -
🌽 Trace	🔲 DQM	suppression	DDR	
List	– DTM –––––	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	- PTCM	SpenPTM		NONE -
	PID_MSR	SpenDTM		- MODE
	BL_HTM	SpenOTM		OFF 👻
	TLBNEW	- SupprTHReshold -		
	TLBINV	1/4 🔻		

## Example:

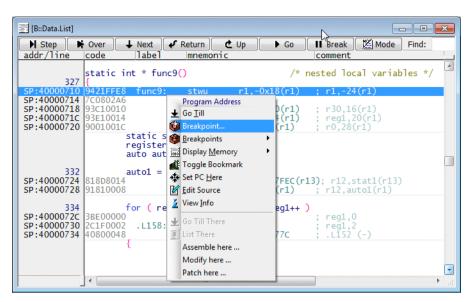
Advise the NEXUS module to start Branch Trace messaging and Data Write Messages at the entry to the function func9.

Advise the NEXUS module to stop Branch Trace messaging and Data Write Messages at the exit of the function func9.

nexus	<ul> <li>selection ———</li> </ul>	option	configuration —	CLIENT1
OFF	🖉 BTM	SmartTrace	- PortSize	- SELECT
ON	HTM	POTD	MDO4 -	NONE
	🔲 ОТМ	- STALL	- PortMode	MODE
RESet	WTM	OFF 🔻	1/2 🔻	OFF
🥬 Trace	DQM	suppression	DDR	
List	– DTM –––––	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	- PTCM	SpenPTM		NONE
	PID_MSR	SpenDTM		MODE
	BL_HTM	SpenOTM		OFF
	TLBNEW	- SupprTHReshold		
	TLBINV	1/4 🔹		

1. Enable Branch Trace Messages and Data Write Messages.

2. Set a Program breakpoint to the entry of the function func9 and select the action TraceON.



🔯 B::Break.Set SP:0)	«40000710 /DIALOG		
- address / expres			
SP:0x40000710			🔻 🗾 🗆 HLL
- type	options		- implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	action
Read		DISableHIT	TraceON 🔻
Write	- DATA		<b></b>
Ø default		-	Sector Secto
 Ok	Add	Delete	Cancel

3. Set a Program breakpoint to the exit of the function func9 and select the action TraceOFF.

🗾 [B::Data.List]		
► Step addr/line SP:40000790 SP:40000794		Break     Mode     Find:       comment
355 SP:40000798 SP:4000079C SP:400007A0 SP:400007A4	func10() ↓ Go Till 9421FFA8 func10: Breakpoint (r1) 7C0802A6 BDC10010 9001005C register register register register	; r1,-88(r1) ; v16,16(r1) ; r0,92(r1) , v8; 4, v15, v16, v17;
360 SP:400007A8 SP:400007A0 SP:400007B0 SP:400007B8 362 SP:400007B8 SP:400007B0 SP:400007C0	v17 = 0; 39800000 91810008 for (i 3BE00000 2C1F0003 . L184: 40800018 81810008 710 C11 Go Till There List There Assemble here Modify here Patch here Patch here 71) 71) 71)	; r12,0 ; r12,v17(r1) ; i,0 ; i,3 ; .L182 (-) ; r12,v17(r1) ; r12,v17(r1) ; r12,v17(r1)
SP:400007C0	7D8CFA14	; r12,r12,i

B::Break.Set SP:0			
SP:0x40000794	551011		▼ 🗾 🗆 HLL
- type Program	options	Temporary	implementation
<ul> <li>ReadWrite</li> <li>Read</li> </ul>	NOMARK	DISable	action
<ul> <li>Write</li> <li>default</li> </ul>	DATA		✓ advanced
Ok	Add	Delete	Cancel

🕲 B::Break.List					×
XDelete All O Disable All O Er	able All 🛛 🛛 🛛 Init	🧷 İmpl.	😨 Store	🔁 Load 🚺 Set	
address	types	impl	action		1
C:40000710		ONCHIP	TraceON	func9	
C:40000794	Program	ONCHIP	Trace0FF	func9\25+0x14	
	-				
					Ŧ
	•			4	

4. Start the program execution and stop it.

# 5. Display the result.

B::Trace.List			x
	욖 Goto 🎁 Find 🥂 Chart 📕 Profile 📕 MIPS 🗢 More 🗶 Less		
record	un address cycle data symbol	ti.back	_
-00129868	bge 0x4000077C ; .L152 (-) F:4000077C ptrace //diabc/func9+0x6C	0.360us	* •
351	return &stat1 r subi r3,r13,0x7FEC ; r3,r13,32748 TRACE ENABLE		Î
-00129866 -00129864 -00129863 -00129861 -00129859 -00129857	D:4007F78         wr-long         4000402c         \\diabc\Global\SP_TEST+0x580           D:40007F7C         wr-long         000000B         \\diabc\Global\SP_TEST+0x584           D:40007F84         wr-long         400011C8         \\diabc\Global\SP_TEST+0x586           D:40007F70         wr-long         48003B08         \\diabc\Global\SP_TEST+0x578           D:40007F70         wr-long         48003B08         \\diabc\Global\SP_TEST+0x578           D:40007F74         wr-long         48003B08         \\diabc\Global\SP_TEST+0x576           F:40000774         ptrace         \\diabc\diabc\diabc\func9+0x64	1.883ms 1.720us 1.480us 1.740us 1.720us 1.620us	
334 -00129856	<pre>for ( reg1 = 0 ; reg1 &lt; 2 ; reg1++ ) addi     r31,r31,0x1     ; reg1,reg1,1 b     0x40000730     ; .L158 F:40000730 ptrace</pre>	0.360us	
	t static stat2 = 0; register reg2; auto auto2;		
340 -00129855	auto2 = stat2; lwz r12,-0x7FE8(r13) ; r12,stat2(r13) stw r12,0x0C(r1) ; r12,auto2(r1) D:40007F74 wr-long 48003ACB \\diabc\Global\SP_TEST+0x57C	1.480us	
	*		

The event that switched the trace generation on is not visible in the trace.

## Resource: Watchpoints

# Controlled message types

WTM	ВТМ	DTM	ОТМ	DQM
Unused	BTM is enabled by filter	Unaffected	Unaffected	Unaffected
	Filter applies			

Disable messages types that are unaffected and not required for the analysis.

B::NEXUS				- • ×
B::NEXUS      Nexus      OFF     ON      RESet      Trace      List	- selection Ø BTM HTM OTM WTM DQM DTM Write PID_MSR	option SmartTrace POTD STALL OFF suppression SpenDQM SpenPTM SpenDTM	Configuration → PortSize MDO4 ✓ PortMode 1/2 ✓ DDR	CLIENT1 SELECT NONE OFF CLIENT2 SELECT NONE MODE
	E FLD_MSK BL_HTM TLBNEW TLBINV	SpenOTM SupprTHReshold - 1/4		OFF *

#### Example:

Advise the NEXUS module to generate trace information for all write accesses.

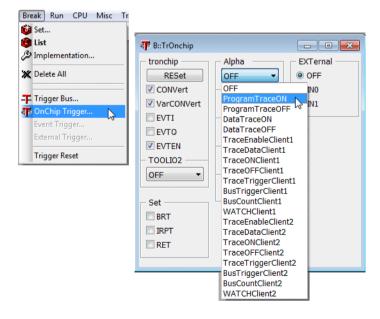
Advise the NEXUS module to start the Branch Trace messaging at the entry to the function func9.

Advise the NEXUS module to stop Branch Trace messaging at the exit of the function func9.

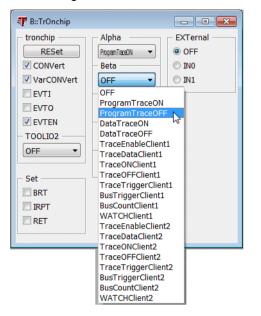
1. Enable Data Trace messaging for write accesses.

B::NEXUS				- • •
nexus	selection	option	configuration —	CLIENT1
OFF	BTM	SmartTrace	- PortSize	- SELECT
ON	HTM	POTD	MDO4 v	NONE -
	🔲 ОТМ	- STALL	- PortMode	MODE
RESet	WTM	OFF 🔻	1/2 🔹	OFF 👻
strace 🖉	DQM	suppression	DDR	
List	– DTM	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	PICM	SpenPTM		NONE -
	PID_MSR	SpenDTM		- MODE
	BL_HTM	SpenOTM		OFF -
	TLBNEW	- SupprTHReshold		
	TLBINV	1/4 🔹		

2. Open the TrOnchip window and select ProgramTraceON for Alpha.



3. Select ProgramTraceOFF for Beta.



4. Set a Program breakpoint to the entry of the function func9 and select Alpha.

🙆 B::Break.Set			
- address / expres	ssion		
func9			🔻 🗾 🗆 HLL
- type	options		implementation —
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	Alpha 🔻
Write	DATA		
O default		<b></b>	Sector Secto
L			·
Ok	Add	Delete	Cancel

5. Set a Program breakpoint to the exit of the function func9 and select Beta.

😝 B::Break.List					×
XDelete All ODisable All OEn	able All 🛛 🛛 🛇 Init	🥬 Impl.	💱 Store	🔁 Load 🔯 Set	
	types	impl	action		
F:40000710	Program	ONCHIP	Alpha	func9	
F:40000794	Program	ONCHIP	Beta	func9\25+0x14	
	4				h. A

6. Start and stop the program execution.

7. Display the result.

🌽 Setup 🔒	Goto 🛐 Find	🙀 Chart 🛛 🔛 Profile 🚺	MIPS 🔷 More 🗶 Less	
record ru	n address cvc	le data	symbol	ti.back
-02647066 -02647065 -02647063 -02647061 -02647059	D:40007F7C wr- D:40007F84 wr- D:40007F70 wr- D:40007F74 wr- F:40000774 ptr	long         000000           long         400011           long         48006B           long         48006A	0B \\diabc\Global\SP C8 \\diabc\Global\SP D1 \\diabc\Global\SP .C4 \\diabc\Global\SP \\diabc\diabc\func9	_TEST+0x58C 1.480us _TEST+0x578 1.740us _TEST+0x57C 1.720us
334	addi r31,r3 b 0x4000		reg1,1	
-02647058	F:40000730 ptr		\\diabc\diabc\func9	+0x20 0.360us
	cmpwi r31,0x bge 0x4000			
		static stat2 = 0; register reg2; auto auto2:		

Command line example

```
; establish a default start situation
Break.Delete /ALL
TrOnchip.RESet
; messaging setup
NEXUS.BTM ON
NEXUS.DTM Write
; filter settings
TrOnchip.Alpha ProgramTraceON
TrOnchip.Beta ProgramTraceOFF
Break.Set func9 /Program /Alpha
Break.Set sYmbol.EXIT(func9) /Program /Beta
Go
•••
Break
; display result
Trace.List
```

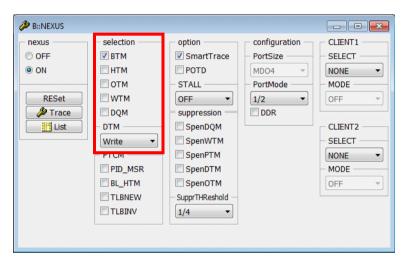
#### **Resource:** Watchpoints

### **Controlled message types**

WTM	втм	DTM	ОТМ	DQM
Unused	Unaffected	Filter applies	Unaffected	Unaffected

Enable Data Trace messaging as required for the analysis.

Disable messages types that are unaffected and not required for the analysis.



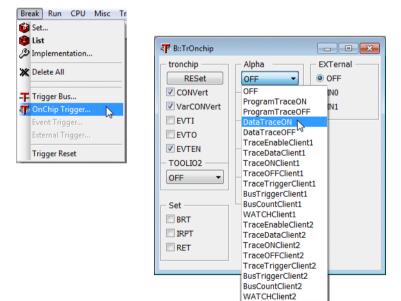
#### Example:

Enable Branch Trace messaging. Advise the NEXUS module to start the generation of Data Write Messages at the entry to the function func9. Advise the NEXUS module to stop the generation of Data Write Messages at the exit of the function func9.

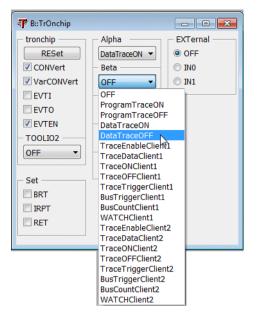
1. Enable Branch Trace messaging and Data Trace messaging for write accesses.

B::NEXUS				
BENEXUS  Nexus  OFF  OFF  ON  RESet  Trace  List	- selection V BTM HTM OTM WTM DQM DTM Vrite PICM PID MSR	option SmartTrace POTD STALL OFF suppression SpenDQM SpenVTM SpenPTM SpenDTM	Configuration PortSize MDO4 PortMode 1/2 DDR	CLIENT1 SELECT NONE MODE OFF SELECT NONE MODE
	BL_HTM TLBNEW	SpenOTM SupprTHReshold — 1/4 •		OFF *

2. Open the TrOnchip window and select DataTraceON for Alpha.



3. Select DataTraceOFF for Beta.



4. Set a Program breakpoint to the entry of the function func9 and select Alpha.

🔯 B::Break.Set			
address / expre	ssion		
func9			🔻 🗾 🗆 HLL
- type	options		implementation —
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	action
Read		DISableHIT	Alpha 👻
Write	DATA		
Ø default			
L	] [		
Ok	Add	Delete	Cancel

5. Set a Program breakpoint to the exit of the function func9 and select Beta.

😢 B::Break.List					×
X Delete All O Disable All O E	nable All 🛛 🛛 🛛 Init	🖉 Impl	😤 Store	😒 Load 🔞 Set	
address	types	impl	action		
F:40000710	Program	ONCHIP	Alpha	func9	
F:40000794	Program	ONCHIP	Beta	func9\25+0x14	
					-
J	•				h. 🔦

6. Start and stop the program execution.

7. Display the result.

B::Trace.Lis	t				- • ×	
🌽 Setup	🗘 Goto 🎁 Find	Chart	Nore Profile	Less		-
record		ycle dat			ti.back	5
323	li r11, lwz r0,0 insrwi r0,r stw r0,0 vbfiel lis r10, li r9,-	0x4000 -0x1 )x40F0(r12) )x40F0(r12) )x40F0(r12) d.m = -1; 0x4000 0x1 )x40F4(r10)	<pre>1 ; r0,r11,2,17 ; r0,16624(r12) ; r10,16384 ; r9,-1</pre>			
324	mtlr rÓ addi r1,r blr	)xOC(r1) ·1,0x8	; r1,r1,8			
-04040169	F:400011C4 p	otrace	\\diabc\diab	oc\main+0x168	16.420us	
619 -04040168	bl 0x40	func9(); 0000710 otrace	; func9 \\diabc\diab	oc\func9	0.380us	
327	static int * f { stwu r1,-		,	local variables */		
-04040167	D:40007F78 w mflr r0	2	4000402C \\diabc\Glob	al\SP_TEST+0x580	1.600us	
-04040165	D:40007F7C w	0x10(r1) /r-long 0x14(r1)	; r30,16(r1) 0000000B \\diabc\Glob : red1,20(r1)	al\SP_TEST+0x584	1.740us	
-04040164	D:40007F84 w	r-long	400011C8 \\diabc\Glob	al\SP_TEST+0x58C	1.480us	
-04040162	D:40007F70 w	0x1C(r1) wr-long stat1 = 0	48008022 \\diabc\Glob	oal\SP_TEST+0x578	1.720us	
	4				►) a	1

Resource: Watchpoints and logic in NEXUS Adapter (parallel trace only)

# Controlled message types

WTM	BTM	DTM	ОТМ	DQM
Watchpoint Hit Message(s) is generated for the specified instruction(s) or data address+data value	Unaffected	Unaffected	Unaffected	Unaffected

Disable messages types that are unaffected and not required for the analysis.

## Example:

Enable Branch Trace messaging. Advise the NEXUS module to generate a trigger for the trace if the function sieve is entered. Use this trigger to stops the trace recording.

1. Enable Branch Trace messaging.

nexus 🌮				
	Selection BTM OTM OTM DQM DTM OFF	option POTD STALL OFF Suppression SpenDQM SpenWTM	Configuration PortSize MDO12 ♥ PortMode 1/2 ♥ DDR	CLIENT1 SELECT NONE OFF CLIENT2 SELECT NONE
	PTCM PID_MSR BL_HTM TLBNEW TLBINV	SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4		OFF *

2. Set a Program breakpoint to the start address of the function sieve and select the action TraceTrigger.

🔯 B::Break.Set			
address / expres	ssion		
sieve			🔻 🗾 🗆 HLL
- type	– options –		- implementation
Program	EXclude	Temporary	auto 👻
ReadWrite	NOMARK	DISable	action
Read		DISableHIT	TraceTrigger 💌
Write	DATA		
O default		<b></b>	Sector Secto
Ok	Add	Delete	Cancel

#### 3. Start the program execution

The state of the trace changes from Arm to BRK when the trigger occurs.

B:: emulate trigger devices trace	Data Var List	other previous MIX UP
	State of the program execution (running)	State of the trace recording (Arm = recording)
B:: emulate trigger devices trace	Data Var List	other previous
		State of the trace recording (BRK = break by trigger, recording is stopped)

4. Display the result.

🔑 Setup	🗘 Goto 🁔	Find 🚺 🛃 Chart	Profile	MIPS 🔷 More	<b>X</b> Less		
record r	un address	cycle da		symbol	·	ti.back	
582 ]	mtlr addi blr	r31,0x1C(r1) r0,0x24(r1) r0 r1,r1,0x20	; r0,36(r1	L) 2			L L
00000017	P:4000	1358 ptrace		\\diabc_in	t\diabc\main+0x220	1.500us	
670	<mark>⊢ li</mark> cmpwi bge	<pre>for (j = 0; r31,0x0 r31,0x0A 0x4000114C</pre>	<b>j &lt; 10; j++</b> ; j,0 ; j,10 ; .L512 (-	) -)			
672	bl	{ 0x40001388	ve(); ; sieve				C

The trace generation is usually stopped before trace information is generated for the event that caused the trigger.

**Example:** Advise the NEXUS module to generate a trigger for the trace if a write access occurs to the variable flags[3]. Advise TRACE32 to fill another 10% of the trace memory before the trace recording is stopped.

1. Set a Write breakpoint to the variable flags[3] and select the action TraceTrigger.

🔯 B::Break.Set			
- address / expre	ssion		
flags[3]			▼ 📝 🕅 HLL
- type	- options		- implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	TraceTrigger 💌
Write	- DATA		
C default		-	Sector Secto
Ok	Add	Delete	Cancel

2. Define the trigger delay in the *Trace Configuration* Window.

				- • •
METHOD				
Analyzer O CA	nalyzer 🔘 Onchip	◎ ART ◎ LOGG	ER 🔘 SNOOPer 🔘 I	FDX 🔘 LA
			🔘 Integrator 🔘 I	Probe 💿 IProbe
state	used	ACCESS	- TDelay	
OISable		auto 🔻	13421772.	TrOnchip
OFF	0.		10% -	🥬 NEXUS
© Arm	- SIZE	CLOCK		
🔘 trigger	134217728.		THreshold	[]
🔘 break			<b>_</b>	Sector Secto
SPY	Mode	Mode	VCC	
	Fifo	O BusTrace	CLOCK	
commands	Stack	ClockTrace	autofocus	
RESet	🔘 Leash	FlowTrace		
Ø Init	STREAM		TERMination	
SnapShot	O PIPE	Prestore		
🗾 List	© RTS	SLAVE	TestFocus	
AutoArm			AutoFocus	
🗹 AutoInit			XX ShowFocus	
SelfArm				
L				

3. Start the program execution.

The state of the trace changes from Arm to TRG when the trigger occurs.

The state of the trace changes from TRG to BRK when the delay counter elapses.

emulate trigger devices trace Data Var List	other previous
	MIX UP
* *	
State of the State	e of the
program execution trace	e recording
	n = recording)
B::	
 	other previous
running	
	<b>L</b>
	State of the
	race recording
	TRG = trigger occurred,
	delay counter started)
	-
B::	
emulate trigger devices trace Data Var List	other previous
	State of the
	trace recording
	(BRK = delay counter elapsed,
	recording is stopped)

### 4. Display the result.

B:: Trace.List NEXUS DEFault			
🖉 Setup 🔁 Goto 🛛 🛱 Find 🛛 🚻 Chart 🖉 🌉 Profile 🖉 🌃 MIPS 🛛 🖨 More 🗶 Les		Trace Goto	
record nexus	run address cycle data syn	nbol	
-00000001 TCODE=05 SPT=0 DT-DWM S=0000 A=00000001 V=00000000000000000	stbx r11.r12.r31 : r11.r12.j	diabc_int\G	Goto
T00000000 TCODE=03 SPI=0 PT-DBM S=0007	P:400013AC ptrace	diabc_int\di Previous	First Trigger Zero
	- cmpwi r31,0x12 ; i,18 bgt 0x400013cC ; .1527 (-) l's r12,0x4000 ; r12,16384 addi r12,r12,0x5620 ; r12,r12,2204/ l'i r11,0x1 ; r11,1,1 stbx r11,r12,r31 ; r11,r12,;	8 Next	Last Ref Track Cancel
+00000001 TCODE=05 SPI=0 DT-DWM S=0000 A=00000003 V=0000000000000000000000000000	D:40005622 wr-byte 01 \\	diabc_int\Global\flags+0x2	0.500us
+00000002 TCODE=03 SPI=0 PT-DBM S=0007	addi r31,r31,0x1 : i,i,1 b 0x40013AC : i.152 Cmpwi r31,0x12 : i,18 bqt 0x40013C2 : i,257 (-) lis r12,0x40013C2 : 1.527 (-) lis r12,0x40005 : r12,12,6384 addi r12,r12,0x5620 : r12,12,034 li r11,0x1 : r11,0x1	diabc_int/diabc\sieve+0x24	0.340us
+00000004 TCODE=05 SPI=0 DT-DWM S=0000 A=00000001 V=0000000000000000000000000000	addi r31.r31.0x1 : i.i.1	diabc_int\Global\flags+0x3	0.840us
+00000005 TCODE=03 SPI=0 PT-DBM S=0007	b 0x400013AC ; .1529 P:400013AC ptrace \\ cmpwi r31,0x12 ; i,18	diabc_int\diabc\sieve+0x24	0.320us
			E. A

Push the *Trigger* button in the *Trace Goto* window to find the record, where TraceTrigger was detected by the trace (WHM message). Here the sign of the record numbers has changed. The TraceTrigger event is usually shortly after this point.

Resource: Watchpoints and logic in NEXUS Adapter (parallel trace only)

#### Controlled message types

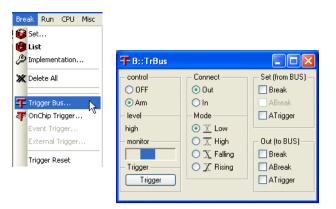
WTM	втм	DTM	ОТМ	DQM
Watchpoint Hit Message(s) is generated for the specified instruction(s) or data address+data value	Unaffected	Unaffected	Unaffected	Unaffected

Example: Generate a 100 ns high pulse on the trigger connector of POWERTRACE/ETHERNET or POWER DEBUG II when a write access to flags[9] occurs.

1. Set a write breakpoint to the variable flags[9] and select the action BusTrigger.

😻 B::Break.Set							
address / express flags[9]	ion		<b>V</b>	) 🗹 HLL			
type Program ReadWrite Read Write any Ok	options EXclude NOMARK DATA Set	Temporary DISable DISableHIT Delete	auto actio Bus1				
🕲 B::Break.Lis							
<u> </u>	isable	pes i	ect) 😭 9 mp 1 NCH I P	action BusTrigg	 Set	Global\fla	ags[9]
	<						

2. Start the program execution. 3. Open the *TrBus* window to watch the trigger.



Resource: Watchpoints and logic in NEXUS Adapter (parallel trace only). Only one event possible.

### Controlled message types

WTM	BTM	DTM	ОТМ	DQM
Watchpoint Hit Message(s) is generated for the specified instruction(s) or data address+data value	Unaffected	Unaffected	Unaffected	Unaffected

**Example 1:** Count how often the function sieve is called.

1. Set a Program breakpoint to the start address of the function sieve and select the action BusCount.

address / express	ion			
sieve			🖌 [ 🗍 🗍 HLL	
type Program ReadWrite Read Write any	options EXclude NOMARK DATA	Temporary DISable DISableHIT	implementation auto	
Ok	Set	Delete	Cancel	
B::Break.Lis				
🗶 Delete All 🚺 🖸 D	isable 💿 Enable address tu	⊘ Init 🖉 Selec pes im	t 😭 Store 😭 Lo pl action	oad 🚺 🔯 Set

2. Open the TRACE32 counter window and select EventHigh.

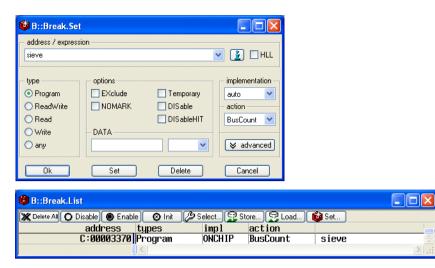
Misc Trace Perf Cov MPC Veco Oscillator Frequency Counter Pulse Generator	& B::Count					
Pulse Generator 2     Runtime     Memory Map	0	0.	evt_	_· IAC1	(*1)	LOW
Flash Programming     Choose Colors     Chorse Colors     Japaneese Menu	Mode Frequency Period PulsLow PulsHigh EventLow EventHigh	Gate           0.01 s           1 s           10 s           endless	Init AutoInit OUT	Select PODBUS iAC1 IAC2 IAC3 IAC4 DAC1 DAC1 DAC2 DCNT1 DCNT2 MCK0		

3. Start the program execution and display the result.

& B::Count	
0 38.605	evt IAC1 (*1)
Mode Gate Frequency 0.0 Period 0.1 PulsLow 1s PulsHigh EventLow end EventHigh EventHOld	AutoInit     OIAC1     OIAC2     OIAC3     OUT     OIAC4

**Example 2:** Measure the averaged time distance in which the function sieve is called.

1. Set a Program breakpoint to the start address of the function sieve and select the action BusCount.



2. Open the TRACE32 counter window and select Period.

isc Trace Perf Cov MPC						
Vco Oscillator						
Frequency Counter	_					
Pulse Generator 🛛 😽	& B::Count					
Runtime						
Memory Map			ns	IAC1	(*100)	LOW
Flash Programming	– Mode –	Gate	init	- Select		
Choose Colors	O Frequency	◯ 0.01 s	Init	O PODBUS		
🞾 Ethernet Config	<ul> <li>Period</li> </ul>	⊙ 0.1 s	AutoInit	<ul><li>● IAC1</li></ul>		
Japaneese Menu	RulsLow	O1s		O IAC2		
	O PulsHigh	◯ 10 s	- out	O IAC3		
	◯ EventLow	🔘 endless	ООТ	O IAC4		
	◯ EventHigh	– variable –		O DAC1		
	◯ EventHOld			O DAC2		
			PROfile	O DCNT1		
				O DCNT2		
				🔿 МСКО		

3. Start the program execution and display the result.

🖧 B::Count					
0 89	.200	us	IAC1	(*100)	
Mode Frequency Period PulsLow PulsHigh EventLow EventHigh EventHOld	Gate 0.01 s 0.1 s 1 s 0 l0 s 0 endless variable	Init AutoInit	Select PODBUS IAC1 IAC2 IAC3 IAC4 DAC1 DAC1 DAC2 DCNT1 DCNT2 MCK0		

Filters and Triggers are programmed to all cores that are controlled by the TRACE32 instance.

The fact that TRACE32 does not know on which core of the SMP system a program section is running has the consequence that the same filters/triggers are programmed to all cores. So, from the perspective of TRACE32, you can say the resources for filters/triggers are shared by all cores.

### Resource: Watchpoints

### Controlled message types

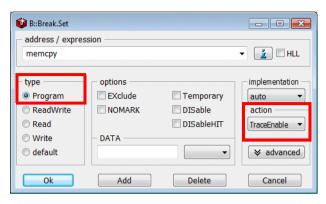
WTM Watchpoint Trace Messages	<b>BTM</b> Branch Trace Messages	<b>DTM</b> Data Trace Messages	<b>OTM</b> Ownership Trace Messages	<b>DQM</b> Data Acquisition Messages
Watchpoint Hit Message(s) is generated for the specified instruction(s)	Disabled	Unaffected	Unaffected	Unaffected

Disable message types, that are unaffected by the filter and not required for your analysis.

B::NEXUS				
B::NEXUS      nexus     OFF     OF     ON      RESet     D'Trace     List	Selection ♥ BTM HTM OTM WTM DQM DTM OFF ♥TCM PID_MSR BL_HTM TLENEW TLENEW	option POTD STALL OFF Suppression SpenDQM SpenVTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4	Configuration PortSize MDO12 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE MODE OFF CLIENT2 SELECT NONE OFF MODE OFF

**Example 1:** Advise the NEXUS module to generate only trace information for the entries to the function memcpy.

1. Set a Program breakpoint to the start address of the function memcpy and select the action TraceEnable.



- 2. Start the program execution and stop it.
- 3. Display the result.

📰 B::Trace.List	
Ø Setup A Goto A Goto A Find A Goto A	
record run address cycle data symbol ti.back	
-000000129 0   V:000019C6 ptrace \\sample1\Global\memcpy 2.904ms	4
-000000122 1 TRACE ENABLE -000000122 1 v:000019C6 ptrace \\sample1\Global\memcpy 5.800ms 1 se_mtar r8,r5 TRACE ENABLE	*
-0000000120 0 V:000019C6 ptrace \\sample1\Global\memcpy 5.800ms 0 se_mtar r8,r5 TRACE ENABLE	
-0000000117 1 V:000019C6 ptrace \\sample1\Global\memcpy 2.958ms 1 se_mtar r8,r5 TRACE ENABLE	
-0000000116 0 V:000019C6 ptrace \\sample1\Global\memcpy 2.896ms	•

Break.Delete /ALL	;	delete all breakpoints
Break.Set memcpy /Program /TraceEnable	;	program filter
Go	;	start program execution
Break	;	stop program execution
Trace.List	;	display result

The following **Trace.STATistic** command calculates the time intervals for a program address event. The program address event is here the entry to the function memcpy. The core information is discarded for this calculation.

Trace.STATistic.AddressDIStance memcpy

ß	Setup	Chart 🔶	Zoom	Zoom	Full						
		samples: total:	7495. 41.653s	avr: in:	5.509ms 41.287s		176.970us 366.496ms		41.548ms 99.120%		
		LOLAT:	41.0005	101:	41.20/5	out:	500.490ms	ratio:	99.120%		
		count		1%	2%	5%	10%	20%	50%	100	
	0.000us	0.	0.000%								
	20.000ms	7378.	98.438%							_	
	40.000ms	0.	0.000%								
	60.000ms	0.	0.000%								
	80.000ms	0.	0.000%								
	100.000ms	0.	0.000%								
	120.000ms	0.	0.000%								
	140.000ms	0.	0.000%								
	160.000ms	0.	0.000%								
	180.000ms 200.000ms	0.	0.000%								
	220.000ms	0.	0.000%								
	240.000ms	70.		4							
	260.000ms	47.									
	280.000ms	0.									
	300.000ms	0.	0.000%								
	320.000ms	0.	0.000%								
	520.000005	ŏ.	0.000%								

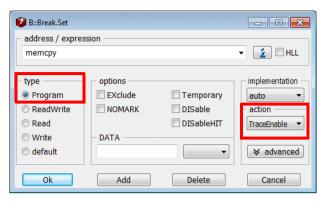
If you need the result per core, use the following command:

Trace.STATistic.AddressDIStance memcpy /CORE 0

	E	B::Trace.STATistic.AddressDIStance memcpy /CORE 0									
300.000ms         0.         0.000%           320.000ms         0.         0.000%		Up to 0.000us 20.000ms 40.000ms 60.000ms 100.000ms 120.000ms 140.000ms 160.000ms 160.000ms 180.000ms 220.000ms 240.000ms 240.000ms	Count	Image: Coordination of the system         Image: Coordination of the system<	oom avr: in:	11.019ms 41.287s	out:	366.744ms	ratio: 9	1.796ms 99.119%	
	>	280.000ms 300.000ms	0. 0.	0.000%							Ŧ

**Example 2:** Advise the NEXUS module to generate trace information for the entries to the function memcpy and for the exits of the function memcpy.

1. Set a Program breakpoint to the start address of the function memcpy and select the action TraceEnable.



2. Set a Program breakpoint to the exit address of the function memcpy and select the action TraceEnable.

🔯 B::Break.Set			
address / expres	ssion		
memcpy+0x78			🔻 🗾 🗖 HLL
- type	options		- implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	action
Read		🔲 DISableHIT	TraceEnable 🔻
Write	- DATA		
🔘 default		-	
Ok	Add	Delete	Cancel

**sYmbol.EXIT**(*<symbol>*) Returns the exit address of the specified function

3. Start the program execution and stop it.

### 4. Display the result.

🌽 Setup 🔒 Goto 🎁 Find 🦳 🚰 Chart 🛛 🚺 Profile	MIPS	
record run address cycle data	symbol	ti.back
	\\sample1\Global\memcpy+0x78	6.445us
0000000026 0 V:000019C6 ptrace se_mtar r8,r5	\\sample1\Global\memcpy	5.797ms
-0000000025 0 V:00001A3E ptrace 0 se_blr TRACE ENABLE	\\sample1\Global\memcpy+0x78	3.995us
0000000022 1 V:000019C6 ptrace se_mtar r8,r5	\\sample1\Global\memcpy	2.952ms
0000000021 1 V:00001A3E ptrace 1 se_blr TRACE ENABLE	\\sample1\Global\memcpy+0x78	7.605us
0000000020 0 V:000019C6 ptrace se_mtar r8,r5	\\sample1\Global\memcpy	2.893ms
0000000019 0 V:00001A3E ptrace 0 se_blr	<pre>\\sample1\Global\memcpy+0x78</pre>	4.900us

The following **Trace.STATistic** command calculates the time intervals between two program address events A and B. The entry to the function memcpy is A in this example, the exit from the function is B. The core information is discarded for this calculation.

Trace.STATistic.AddressDURation memcpy memcpy+0x78

ø	Setup	Chart 🔶	Zoom 🛛 🗶 Z	Zoom	Full					
		samples:	6672.	avr:	5.620us	min:	3.990us	max:	7.610us	
		total:	36.861s	in:	37.500ms	out:	36.824s	ratio:	0.101%	
	up to	count	ratio	1%	2%	5%	10%	20%	50%	100
	3.500us	0.	0.000%							
	4.000us	1660.	24.880%							
	4.500us	440.	6.594%			_				
	5.000us	718.								
	5.500us	518.	7.763%				•			
	6.000us	0.	0.000%							
	6.500us	1995.	29.901%							
	7.000us	105.	1.573%							
	7.500us	0.	0.000%							
	8.000us	1236.	18.525%					•		
	8.500us	0.	0.000%							
	9.000us	0.	0.000%							
	9.500us	0.	0.000%							
	10.000us	0.	0.000%							
	10.500us	0.	0.000%							
	11.000us	0.	0.000%							
	11.500us	0.	0.000%							
		0.	0.000%							

If you need the result per core, use the following command:

Trace.STATistic.AddressDURation memcpy memcpy+0x78 /CORE 0

E B::Trace.STATistic.AddressDURation memcpy memcpy+0x78 /CORE 0											
🔑 S	etup 🛛 🏨	Chart 🔶	Zoom 🔀 Z 3336.	loom avr:	Full 4.350us	min:	3.990us	max:	5.030us		_
		total:	36.861s	in:	14.513ms	out:	36.847s	ratio:	0.039%		
	up to	count	ratio	1%	2%	5%	10%	20%	50%	100	
<	3.900us	0.	0.000%	1							
	4.000us	1660.	49.760%								
	4.100us	440.	13.189%								
	4.200us	0.	0.000%								
	4.300us	0.	0.000%								
	4.400us	0.	0.000%								
	4.500us	0.	0.000%								
	4.600us	0.	0.000%								
	4.700us	0.	0.000%								
	4.800us	0.	0.000%								
	4.900us	235.									
	5.000us	483.	14.478%								
	5.100us	518.	15.527%								
	5.200us	0.	0.000%								
	5.300us	0.	0.000%								
	5.400us	0.	0.000%								
	5.500us	0.	0.000%								
>		0.	0.000%								
		•								•	

# **Examples for TraceEnable on Instruction Range**

Resource: Limited to one instruction address range

#### Controlled message types

WTM	ВТМ	DTM	ОТМ	DQM
Unused	Filter applies if BTM is enabled	Filter applies if DTM is enabled	Unaffected	Unaffected

Enable BTM. This filter requires that Branch History messaging is disabled.

Enable DTM if you are interested in the read/write accesses performed by the specified instruction address range.

Disable message types, that are unaffected by the filter and not required for your analysis.

**Example:** Advise the NEXUS module to generate trace information for all taken branches within the function OSInterruptDispatcher1.

1. Enable Branch Trace messaging, but don't enable Indirect Branch History messaging.

Disable Data Trace messaging.

nexus 🌽				
nexus	<ul> <li>selection ———</li> </ul>	option	configuration —	CLIENT1
OFF	🗹 ВТМ	POTD	- PortSize	- SELECT
ON	🔳 НТМ	- STALL	MDO12 -	NONE -
	OTM	OFF 👻	- PortMode	MODE
RESet	WTM		1/2 🔻	OFF 👻
🌽 Trace	DQM	- suppression	DDR	CLIENT2
List	– DTM	SpenDQM		- SELECT
	OFF 👻	SpenWTM		NONE -
	– PTCM –––––	SpenPTM		MODE
	PID_MSR	SpenDTM		OFF 👻
	BL_HTM	SpenOTM		
	TLBNEW	- SupprTHReshold		
		1/4 🔹		
	L]	L		

2. Set a Program breakpoint to the complete address range of the function OSInterruptDispatcher1 (HLL check box ON) and select the action TraceEnable.

🔯 B::Break.Set			
- address / expre	ssion		
OSInterruptDisp	atcher1		🕶 📝 🗹 HLL
- type	options		- implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
© Read		DISableHIT	TraceEnable 💌
Write	DATA		
O default		<b></b>	Sector Secto
Ok	Add	Delete	Cancel

- 3. Start the program execution and stop it.
- 4. Display the result.

[B::Trace.Lis	st /t]											×
🌽 Setup	🔒 Goto.	🎒 Find 🛛	Chart	📕 Profile	MIPS	6 🔶 More	Less					
		address	cycle	data		symbol				t	i.back	
-000001916		V:00007C04	ptrace			\\sample1	osisr\OSI	nterruptDis	patcher1+0x426		3.095us	*
		nsync lwzx r8, stbx r22 stwx r26 e_lmw r19 e_lwz r0, se_mtlr r0 e_addi r1, se_blr	r27,r25 ,r21,r31 ,r8,r28 ,0x0C(r1) 0x44(r1) r1,0x40		curApp,r oldPri,r r19,12(r r0,68(r1 r1,r1,64	1)						•
-000001916		V:0000912C				\\sample1	Global\	ghs_eofn_0S	5_StartNonAutosa	rCore+0x14	2.835us	
		e_lmvsrrw 0x4	8(r1)	;	72(r1)							
-000001916	3 1 1 6 1 1	e_add16i r12	,r13,-0x7 r24,r25		r24,r13, r12,r13,	-32736	osisr\OSI	nterruptDis	patcher1+0x72			
		e_add16i r21 lbzx r22 e_andi. r11 e_rlwinm r5,	.,r13,-0x7 ,r21,r31 ,r5,0x1F r5,0x2,0x ,r5,r0	19,0x1D;	r11, r5, 3	21,coreId 1 ,25,29						
-000001916		e_b1 0x4 v:00004032	032 ptrace	;	05CheckS		ostsk\05Cl	heckStack			3.740us	•

### Resource: DTC Register

### Controlled message types

WTM	втм	DTM	ОТМ	DQM
Unused	BTM is disabled by filter	DTM is enabled by filter	Unaffected	Unaffected
		Filter applies		

Disable message types, that are unaffected by the filter and not required for your analysis.

B::NEXUS				
Nexus OFF ON RESet	Selection  V BTM  HTM OTM  DM DM OFF  PTCM  PTCM  BL_HTM  TLBNEW  TLBINV	option POTD STALL OFF SpenDQM SpenDVTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4	Configuration PortSize MDO12 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE * OFF * CLIENT2 SELECT NONE * MODE OFF *

**Example:** Disable Branch Trace Messaging and advise the NEXUS module to generate trace information for the write accesses to the variable hookNmb.

1. Set a Write breakpoint to the variable hookNmb and select the action TraceEnable

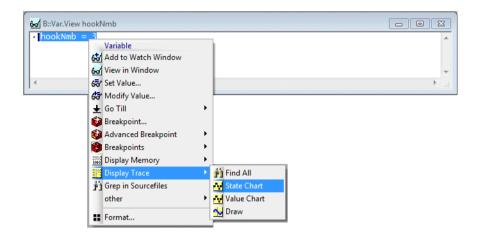
🔯 B::Break.Set			
address / expres	ssion		
hookNmb			👻 📝 HLL
type	- options		- implementation
© Program	EXclude	Temporary	auto 👻
ReadWrite	NOMARK	DISable	- action
🔘 Read		DISableHIT	TraceEnable 🔻
Write	DATA		
Ø default		<b>•</b>	Sector Secto
Ok	Add	Delete	Cancel

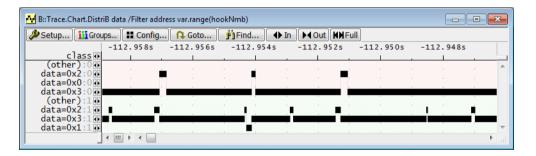
- no data value possible (limitation of DTC Register)
- accessing instruction not possible (limitation of DTC Register)
- 2. Start the program execution and stop it.
- 3. Display the result.

ſ	B::Trace.List						×
	Setup 📭 Goto	. 👔 Find 🕅 🚻 🕻	Chart Profile	MIPS Anore	Less		
	record run a		cle data	symbol	-1 -1 -1	ti.back	
	-0000000031 0 -0000000030 1	D:40000A58 wr- D:40000A58 wr-	-long 00	0000002 \\sample1 0000003 \\sample1	\Global\hookNmb	149.110us 180.555us	
	-000000028 0	D:40000A58 wr-	-long 00	0000003 \\sample1	Global\hookNmb	226.955us	н
	-000000025 1	D:40000A58 wr-		000002 \\sample1		2.710ms	Ť
	-000000023 1	D:40000A58 wr- D:40000A58 wr-	-long 00	0000001 \\sample1 0000002 \\sample1		52.840us 2.707ms	
	-000000020 1	D:40000A58 wr-	-long 00	000003 \\sample1	Global\hookNmb	174.130us	
	-000000018 0	D:40000A58 wr-	-long OC	0000003 \\sample1	\Global\hookNmb	116.780us	-
	I					F	

The Variable pull-down provides various ways to analyze the variable contents over the time.

; open a window to display the variable Var.View hookNmb

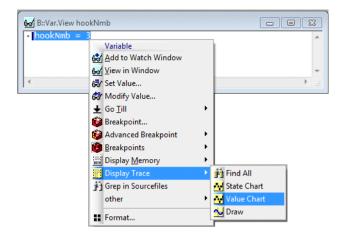




Display the value changes of a variable graphically - value changes per core Trace.Chart.DistriB Data /Filter Address Var.RANGE(<var>) [/SplitCORE]

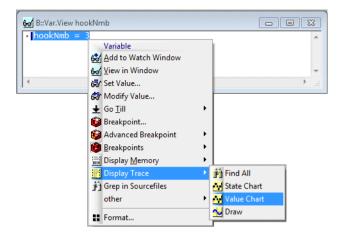
🚰 B::Trace.Chart.DistriB Data /Filter Address Var.RANGE(hookNmb) /JoinCORE							
Setup 🚻 Groups 🔡 Confi	g 🚺 Goto	🛉 Find 📢	In MOut K	Full			
	-2.624s	-2.6225	-2.620s	-2.618s	-2.616s	-2.614s	
class 💀							
(other)							
data=0x2	II.				. 📕		
data=0x0							
data=0x3							
data=0x1 💀							
	• • • • •					in 4	

Display the value changes of a variable graphically - value changes of all cores Trace.Chart.DistriB Data /Filter Address Var.RANGE(<var>) /JoinCORE



B::Trace.Chart.VarState /Filter address var.range(hookNmb)		
🌽 Setup 🔡 Config 🚺 Goto 🎁 Find Mart 🚺 🕪 In 🕅 Mout KM	Full	
112.965750000s -	112.965740000s -112	.965730000s
range 🚯 🚬 👘 👘 👘		
hookNmb 🚯 1	2	3
		<b>T</b>
		n. [4

Display variable contents over the time (numerically) - the core information is discarded **Trace.Chart.VarState** 



B::Trace.Df	RAW.Var %def hookNmb				- • •
🌽 Setup	Goto 🛐 Find.		In MURT		
	.1105	-91.105s	-91.100s	-91.095s	-91.09
	3.				
	1.		.		🗆
	0				• • • •

Display variable contents over the time (graphically) - the core information is discarded **Trace.DRAW.Var %DEFault** *<var>* 

### Resource: DTC Register

### Controlled message types

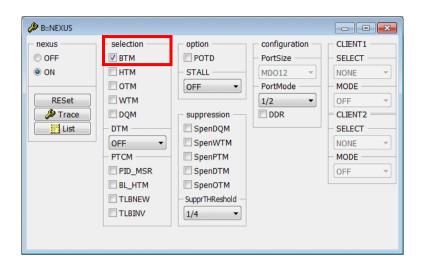
WTM	втм	DTM	ОТМ	DQM
Unused	Unaffected	DTM is enabled by filter	Unaffected	Unaffected
		Filter applies		

Disable message types that are unaffected by the filter and not required for the analysis.

B::NEXUS				
nexus OFF ON RESet Trace List	Selection BTM HTM OTM DQM DTM OFF PTCM PID_MSR BL_HTM TLBNEW TLBINV	option POTD STALL OFF Suppression SpenDQM SpenWTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4	Configuration PortSize MDO12 ▼ PortMode 1/2 ▼ DDR	CLIENT1 SELECT NONE * OFF * CLIENT2 SELECT NONE * MODE OFF *

**Example:** Advise the NEXUS module to generate Data Trace Messages for all write accesses to the variable hookNmb and to generate trace information for all executed instructions.

1. Enable Branch Trace messaging.



2. Set a Write breakpoint to the variable hookNmb and select the action TraceData.

🔯 B::Break.Set			- • •
- address / expres	ssion		
hookNmb			🔻 👱 🗸 HLL
type	options		- implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
🔘 Read		DISableHIT	TraceData 💌
Write	- DATA		
🔘 default			Sector Secto
Ok	Add	Delete	Cancel

- 3. Start the program execution and stop it.
- 4. Display the result.

B::Trace.List							
	Goto 🎁 Find	Chart	🌃 Profile 📔 📕 M	IPS Amore Less			
record r	un address	cycle d	ata	symbol	ti	.back	
52 1 1	{	askHook( vo	id )	/* Routine	to call before	entering task	context */
54 1	hookNmb=						· · · · · · · · · · · · · · · · · · ·
1	se_li r0	,0x2	; r0,2 3) ; r0,hoo				<u> </u>
-0001837177 1	e_stw r0 D:40000A5	8 wr-long	000000	02 \\sample1\Global\ho	okNmb	1.545us	
55 1 1 -0001837175 1 1		9,0x0(r30)	; r19,0(	h\OSTaskInternalDi 30)	spatch+0x148	0.900us	
1	mtpid r2	0					-

Please be aware that in the case of a TraceData filter a correlation of the data access and the instruction is in most cases not possible.

**Resource:** Watchpoints

## **Global TraceON/Trace OFF**

### Controlled message types

WTM	втм	DTM	ОТМ	DQM
Unused	Filter applies	Filter applies	Unaffected	Unaffected

Enable Branch Trace Messaging and Data Trace Messaging if this information is required for your analysis.

Diable messages types that are unaffected and not required for the analysis.

nexus 🔑 B::NEXUS				
nexus	<ul> <li>selection ——</li> </ul>	- option	- configuration	CLIENT1
OFF	🗷 ВТМ	SmartTrace	- PortSize	- SELECT
ON	🔳 HTM	POTD	MDO4 -	NONE -
	🔲 ОТМ	- STALL	- PortMode	- MODE
RESet	🗆 WTM	OFF 🔻	1/2 🔻	OFF 👻
strace 🖉	🔲 DQM	suppression	DDR	
List	– DTM –––––	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	- PTCM	SpenPTM		NONE -
	PID_MSR	SpenDTM		- MODE
	BL_HTM	SpenOTM		OFF 👻
	TLBNEW	- SupprTHReshold -		
	TLBINV	1/4 🔻		

#### Example:

Advise the NEXUS module to start Branch Trace messaging and Data Write Messages at the entry to the function OSInterruptDispatcher1.

Advise the NEXUS module to stop Branch Trace messaging and Data Write Messages at the exit of the function OSInterruptDispatcher1.

B::NEXUS				
nexus	<ul> <li>selection ——</li> </ul>	option	configuration —	CLIENT1
OFF	🗹 ВТМ	SmartTrace	- PortSize	- SELECT
ON	HTM	POTD	MDO4 -	NONE -
	🔲 ОТМ	- STALL	PortMode	MODE
RESet	WTM	OFF 👻	1/2 -	OFF -
strace 🌽	DQM	suppression	DDR	
List	– DTM	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	- PTCM	SpenPTM		NONE -
	PID_MSR	SpenDTM		MODE
	BL_HTM	SpenOTM		OFF -
	TLBNEW	- SupprTHReshold -		
	TLBINV	1/4 👻		

1. Enable Branch Trace Messages and Data Write Messages.

2. Set a Program breakpoint to the entry of the function OSInterruptDispatcher1 and select the action TraceON.

B::List OSIr	terruptDispa	tcher1]							
► Step	• Over	↓ Next	🖋 Return	C Up	► Go	II Break	Mode	Find:	osisr.c
addr/1	ine code	labe	1		mnemonic		co	omment	
	546 ////	///////////////////////////////////////	///////////////////////////////////////	//////	1//////////////////////////////////////	///////////////////////////////////////	///////////////////////////////////////	///////	///////////////////////////////////////
SV:00007			terruptDisp	oat cher:	1: e_stwu	r1,-0x40		r1,-64(r	
SV:00007		)90C		F	rogram Address	×00	:(r1) ;	r19,12(r	1)
SV:00007				+ 0	Go Till				
SV:0000/	7E8 54010	044			_	44 (	(r1)	r0,68(r1	
					Breakpoint	//			
	549			/// 🕲 🗉	reakpoints	• //			///////////////////////////////////////
	7EC 7FFE4	245	(//////////////////////////////////////	- i 🗐 🛙	)isplay <u>M</u> emory	► pr	///////////////////////////////////////	coreId,p	
SV:00007							0x2,0x0E		r25.coreId.2.14
	7F4 1F6D8				Bookmark	13			13,-32696
SV:00007	7F8 739FE	7F5		۲ 🍋 🛛	loggle Bookmark			r28720	
SV:00007	7FC 7D1BC	82E		- <b>*</b> S	et PC <u>H</u> ere	7,r	-25		
	800 1F9C8				dit Source	28	-0x7FF8		28,-32760
SV:00007	804 7F48E	06E				8,r	-28 ;	oldPri,r	8,r28
	////			/// 🛓 ۷	/iew <u>I</u> nfo				
	563 ////	[[[[[[]]]]]]	///////////////////////////////////////			—— V//	[[[[[[[]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]	///////	///////////////////////////////////////
	808 7 3 A OE				io Till There			r29,6553	
SV:00007	80C 51680 810 1FBD9			澤  L	ist There			r11,8(r8	
SV:00007				4	ssemble here				29,-27796 r12,r11,31,1,30
	818 7C0CF					2,1		, over ,	112,111,01,1,00
	81C 50E80			L L	Aodify here			r7,0(r8)	
	820 72E00			F	atch here	xò		r23,0	-
	1			111					▶
1	_								

🔯 B::Break.Set SV:0	x77DE /DIALOG		
address / expres	ssion		▼ 2 □ HLL
- type	options		implementation
Program	EXclude	Temporary	auto 👻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	TraceON 💌
Write	- DATA		
Ø default		•	Sector Secto
Ok	Add	Delete	Cancel

3. Set a Program breakpoint to the exit of the function OSInterruptDispatcher1 and select the action TraceOFF.

(B::List)				
Step Nover	Next 🖌 Return 🚺	Up 🕨 Go	🚺 Break 🕅 Mode	Find:
addr/line code	label mnemoni		comment	
SV:00007C14 1A610800 SV:00007C18 50010044		r19,0x0C(r1)	; r19,12(r1)	-
SV:00007C18 50010044 SV:00007C1C 0090	e_iwz se_mtlr	r0,0x44(r1)	; r0,68(r1)	
SV:00007C1E 18218040	) e_addi	r1,r1,0x40	; r1,r1,64	
SV:00007C22 0004	se blr	am Address		<b>.</b>
	→ Go Til	And the second se		⊫ . •
	💏 Break			
	Break			
		y Memory		
	<b>s</b> ∰ Bookr			
	Micore .	e Bookmark		
	🔆 Set PC			
	Edit S			
	🖌 View I			
	🛨 Go Til			
	List TI			
		nble here		
		y here		
	Patch	here		
B::Break.Set SV:0x7C22 /DIAL	DG			
- address / expression				
SV:0x7C22		👻 🚺 🗆 HLL		
- type options		- implementation -	_	
<ul> <li>Program</li> <li>EXclu</li> </ul>		auto 👻		
ReadWrite     NOM		action		
© Read	DISableHIT	TraceOFF 🔻		
O Write DATA -				
© default	-			
Ok Ad	d Delete	Cancel		

😢 B::Break.L	.ist					×
X Delete All		able All 🔘 Enab	le All 💿 Init	t 🖉 Impl	Store SLoad Set	
V:0000	)77DE	Program Program	ONCHIP	TraceON TraceOFF	OSInterruptDispatcher1 OSInterruptDispatcher1\39+0x3E8	*
v.0000	// C22	ri ogram	ONCHIP	II accorr		Ŧ
1		] <			4	

4. Start the program execution and stop it.

### 5. Display the result.

B::Trace.List								- • ×
🌽 Setup 🔃 🔒 Go	to 🎁 Find	Chart	Profile	MIPS	♦ More	Less		
record rur	n address	cycle	data		symbol	· · · · · · · · · · · · · · · · · · ·		ti.back
-0003380107 0	V:00007C04	ptrace			\\sample1	osisr\OSInter	ruptDispatcher1+0x426	3.095us 🔺
-0003380106 0 -0003380105 0 0 0 0 0 0	stbx r22 D:40000A4C stwx r26 D:FFF48008 e_lmw r19 e_lwz r0, se_mtlr r0	,r8,r28 wr-long ,0x0C(r1) 0x44(r1)	; 0 ; r ; r	03 1dPri,r8 0000000 19,12(r1 0,68(r1)	,r28	Global\OsAppI	D_	1.160us ^ 0.645us
-0003380103	e_addi ri, se_blr V:0000912C - TRACE ENABLE		; r	1,r1,64	\\sample1\	Global\ghs_	eofn_05_StartNonAutosarCore+0x14	1.030us
-0003380100 0 -0003380098 0 -0003380097 0 -0003380094 0 -0003380094 0 -0003380090 0 -0003380089 0 -0003380089 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	D:40001EF8 D:40001F88 D:40001F00 D:40001F00 D:40001F10 D:40001F20 D:40001F20 D:40001F20 D:40001F20 D:40001F21 D:40000F21 D:40000F21 D:400	wr-long wr-quad wr-quad wr-quad wr-quad wr-quad wr-quad wr-long ptrace ,r13,-0x7 r12,r25 ,r13,-0x7 r12,r25 ,r13,-0x7 ,r21,r31 ,r5,0x1F	000000000 000000000 000000144 400009E80 000000000 000000000 000000000 0000000	0000000 0000000 FF48008 0000000 0000012C 24,r13,- 12,r13,- urApp,r2 11,r5,31 r5,r5,2, srPtr,r5	<pre>\\sample1 \\sample1 \\sample1 \\sample1 \\sample1 \\sample1 \\sample1 \\sample1 32736 32688 32620 1,coreId 25,29 r0</pre>	Global\_Osort Global\_Osort Global\_Osort Global\_Osort Global\_Osort Global\_Osort Global\_Osort osisr\OSInter	iStackStart+0xEB4 iStackStart+0xED8 iStackStart+0xEC0 iStackStart+0xEC8 iStackStart+0xED0 iStackStart+0xED0 iStackStart+0xEE0 iStackStart+0xEEC ruptDispatcher1+0x72	1.882ms 0.900us 0.645us 0.905us 1.160us 1.160us 0.645us 0.645us
-0003380087 0		882			·· ·	Global\OsIsrA	rrayCore0	0.640us
-0003380085	e_b1 0x4 V:00004032	032 ptrace	; 0	SCheckSt		ostsk\0SCheck	Stack	0.645us -

The event that switched the trace generation on is not visible in the trace.

### Controlled message types

WTM	ВТМ	DTM	ОТМ	DQM
Unused	BTM is enabled by filter	Unaffected	Unaffected	Unaffected
	Filter applies			

Diable messages types that are unaffected and not required for the analysis.

B::NEXUS				- • •
nexus	<ul> <li>selection ——</li> </ul>	option	- configuration -	CLIENT1
OFF	🗹 BTM	SmartTrace	- PortSize	- SELECT
ON	🗆 НТМ	POTD	MDO4 -	NONE -
	🔲 ОТМ	- STALL	- PortMode	- MODE
RESet	🗆 WTM	OFF -	1/2 👻	OFF -
strace 🖉	🔲 DQM	suppression	DDR	
List	– DTM	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	- PTCM	SpenPTM		NONE -
	PID_MSR	SpenDTM		- MODE
	BL_HTM	SpenOTM		OFF 👻
	TLBNEW	- SupprTHReshold -		
		1/4 🔻		

#### Example:

Advise the NEXUS module to generate trace information on all write accesses.

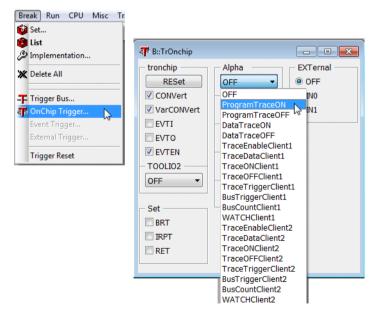
Advise the NEXUS module to start the Branch Trace messaging at the entry to the function memcpy.

Advise the NEXUS module to stop Branch Trace messaging at the exit of the function memcpy.

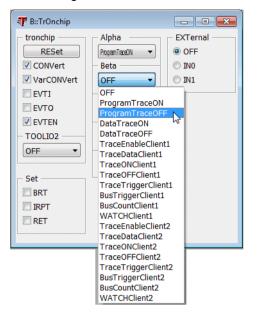
1. Enable Data Trace messaging for write accesses.

B::NEXUS				- • •
nexus	selection	option	configuration —	CLIENT1
OFF	BTM	SmartTrace	- PortSize	- SELECT
ON	HTM	POTD	MDO4 v	NONE -
	🔲 ОТМ	- STALL	- PortMode	MODE
RESet	C WTM	OFF 🔻	1/2 🔹	OFF 👻
strace 🖉	DQM	suppression	DDR	
List	– DTM	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	PICM	SpenPTM		NONE -
	PID_MSR	SpenDTM		- MODE
	BL_HTM	SpenOTM		OFF -
	TLBNEW	- SupprTHReshold		
	TLBINV	1/4 🔹		

2. Open the TrOnchip window and select ProgramTraceON for Alpha.



3. Select ProgramTraceOFF for Beta.



4. Set a Program breakpoint to the entry of the function memcpy and select the action Alpha.

🔯 B::Break.Set			- • •
address / expres	ssion		
тетсру			🔻 🗾 🗆 HLL
			in allow and all an
- type	options		implementation —
Program	EXclude	Temporary	auto 👻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	Alpha 👻
Write	DATA		
O default		-	Sector Secto
L			
Ok	Add	Delete	Cancel

5. Set a Program breakpoint to the exit of the function at memcpy and select the action Beta.

🕲 B::Break.List				×
Address types		Impl	Store Store	]
address types V:000019C6 Program V:00001A3E Program	ONCHIP ONCHIP	Alpha Beta	memcpy memcpy+0x78	*
•				<del> </del>   ▶

6. Start and stop the program execution.

### 7. Display the result.

B::Trace.List							
setup 👔	🕽 Go	oto 🏽 🎒 Find 🖉 Char	t 📔 🎦 Prof	file 📔 📕 MIF	S Amore	Less	
record	rur	n address cycle	data		symbol		ti.back
-0000322899		D:FFF48008 wr-lo		0000000			4.255us 🔺
-0000322897		D:FFF2400E wr-by		0			0.905us
-0000322895	0	V:00001A22 ptrac			\\sample1\	\Global\memcpy+0x5C	2.835us 🚽
	0	e_cmpli 0x0,r8,0x		; 0,r8,0			
	0	se_addi r7,0x4		; r7,RCHW			<b>^</b>
	0	se_addi r4,0x4		; r4,RCHW	2		
	0	se_beq 0x1A3E					
	0	se_mfar r5,r8					
	0	se_mtctr r5		<b>F 1</b>			
	0	se_subi r5,0x1		; r5,1			
	0	lbzx r6,r5,r4 stbx r6,r5,r7					
-0000322894	0	stbx r6,r5,r7 D:40000985 wr-by	+ a	6		Global\OsStacks+0x185	2.065us
-0000322094	6	e bdnz 0x1A30	Le	0	_ \\sampier\	GIODAI (OSSLACKS+0X105	2.065us
-0000322892		V:00001A30 ptrac	~		\\cample1\	Global\memcpy+0x6A	0.640us
-0000322032	6	se subi r5.0x1		; r5,1	(/samprer)	GTODAT (memcpy+0x0A	0.04005
	ŏ	1bzx r6,r5,r4		, 13,1			
	ŏ	stbx r6,r5,r7					
-0000322891	ŏ	D:40000984 wr-by	te	0	7 \\sample1	Global\OsStacks+0x184	0.645us
0000522051	ŏ	e_bdnz 0x1A30		, v	( (Samprer )	(droball (obbedeks) oxio	0101545
	ŏ	seblr					
-0000322890	ŏ	V:00003D4E ptrac	ρ		\\sample1\	osioc\05_0SIocReadAcros	ss+0xD4 0.645us
		TRACE ENABLE	There are a second		, (		
	1						E. 4

; default start situation Break.Delete /ALL TrOnchip.RESet ; messaging setup NEXUS.BTM ON NEXUS.DTM Write ; filter settings TrOnchip.Alpha ProgramTraceON TrOnchip.Beta ProgramTraceOFF Break.Set memcpy /Program /Alpha Break.Set memcpy+0x78 /Program /Beta

Go

•••

#### Break

; display result **Trace.List** 

### Controlled message types

WTM	втм	DTM	ОТМ	DQM
Unused	Unaffected	Filter applies	Unaffected	Unaffected

Enable the Data Trace Messaging as required for the analysis.

Disable messages types that are unaffected and not required for the analysis.

B::NEXUS				
B::NEXUS     Nexus     OFF     OFF     ON     RESet     D'Trace     List	Selection BTM HTM OTM WTM DQM DTM Write FICM PID_MSR	option SmartTrace POTD STALL OFF suppression SpenDQM SpenVTM SpenPTM SpenDTM	Configuration PortSize MDO4 ↓ PortMode 1/2 ↓ DDR	CLIENT1 SELECT NONE OFF SELECT CLIENT2 SELECT NONE MODE
	BL_HTM TLBNEW TLBINV	SpenOTM - SupprTHReshold - 1/4		OFF v

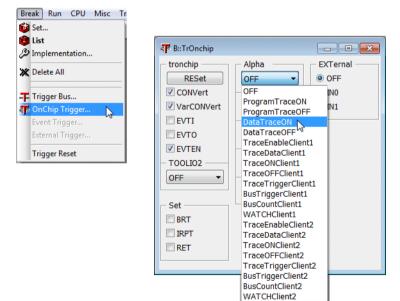
#### Example:

Enable Branch Trace messaging. Advise the NEXUS module to start the generation of Data Write Messages at the entry to the function OSInterruptDispatcher1. Advise the NEXUS module to stop the generation of Data Write Messages at the exit of the function OSInterruptDispatcher1.

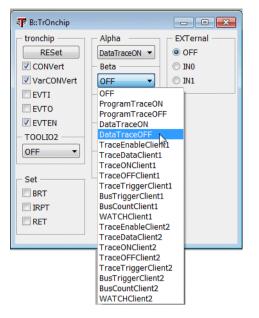
1. Enable Branch Trace messaging and Data Trace messaging for write accesses.

				- • ×
nexus	<ul> <li>selection ——</li> </ul>	option	configuration —	CLIENT1
OFF	🗷 BTM	SmartTrace	- PortSize	- SELECT
ON	HTM	POTD	MDO4 👻	NONE -
	🔲 ОТМ	- STALL	- PortMode	- MODE
RESet	WTM	OFF 🔻	1/2 🔹	OFF 👻
🥬 Trace	DQM	suppression	DDR	
🔣 List	– DTM	SpenDQM		CLIENT2
	Write 👻	SpenWTM		- SELECT
	- PICM	SpenPTM		NONE -
	PID_MSR	SpenDTM		- MODE
	BL_HTM	SpenOTM		OFF 👻
	TLBNEW	- SupprTHReshold -		
	TLBINV	1/4 💌		

2. Open the TrOnchip window and select DataTraceON for Alpha.



3. Select DataTraceOFF for Beta.



4. Set a Program breakpoint to the entry of the function OSInterruptDispatcher1 and select the action Alpha.

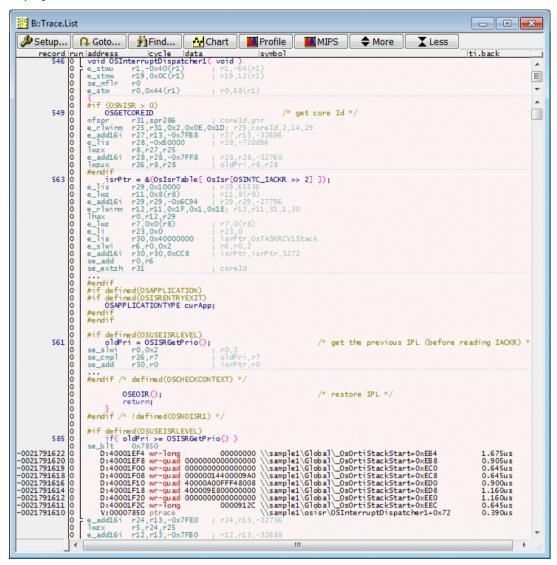
🔮 B::Break.Set			
– address / expre	ssion		
OSInterruptDisp	🔻 🗾 🗆 HLL		
- type	options		implementation —
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	Alpha 🔻
Write	- DATA		
Ø default			Sector Secto
L			L]
Ok	Add	Delete	Cancel

5. Set a Program breakpoint to the exit of the function OSInterruptDispatcher1and select the action Beta.

B::Break.List	icablo All	e All 🛛 🛇 Ini	t 🖉 Impl	Store SLoad 🔞 Set	~
				Store	
address	types	impl	action		
V:000077DE	Program	ONCHIP	Alpha	OSInterruptDispatcher1 OSInterruptDispatcher1\17+0x28	
V:00007830	Program	ONCHIP	Beta	0SInterruptDispatcher1\17+0x28	5
					-
		1			10000

6. Start and stop the program execution.

7. Display the result.



Resource: Watchpoints and logic in NEXUS Adapter (parallel trace only)

#### Controlled message types

WTM	втм	DTM	ОТМ	DQM
Watchpoint Hit Message(s) is generated for the specified instruction(s) or data address+data value	Unaffected	Unaffected	Unaffected	Unaffected

Disable messages types that are unaffected and not required for the analysis.

**Example:** Enable Branch Trace messaging. Advise the NEXUS module to generate a trigger for the trace if the function memcpy is entered. Use this trigger to stops the trace recording.

1. Enable Branch Trace messaging.

				- • ×
B::NEXUS      Nexus     OFF     OF     ON     RESet     D'Trace     Iist	Selection Selection BTM OTM OTM DQM DTM OFF PTCM PID_MSR BL_HTM TLBNEW TLBINV	option POTD STALL OFF SpenDQM SpenDVTM SpenDTM SpenOTM SuprTHReshold 1/4	Configuration → PortSize → MDO12 ↓ PortMode → 1/2 ↓ DDR	CLIENT1 - SELECT - MODE OFF - CLIENT2 - SELECT - NONE - MODE OFF

2. Set a Program breakpoint to the start address of the function memcpy and select the action TraceTrigger.

📦 B::Break.Set			
address / expres	ssion		
memcpy			🔻 🚺 🗆 HLL
- type	– options –		-implementation
<ul> <li>Program</li> </ul>	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	action
Read		DISableHIT	TraceTrigger 🔻
Write	DATA		
Ø default		-	
L			L
Ok	Add	Delete	Cancel

3. Start the program execution.

B:: emulate trigger devices trace	Data Var List	other previous
	State of the program execution (running)	State of the trace recording (Arm = recording)
B:: emulate trigger devices trace	Data Var List	other previous
		State of the trace recording (BRK = break by trigger, recording is stopped)

4. Display the result.

B::Trace.List List.T	ASK DEFault				- • •
🌽 Setup  📭 Go	oto 🗍 🏥 Find	d 🚺 🚮 Chart 🛛 🌉 Pr	ofile	MIPS	
record ru		cycle data		symbol	ti.back
-000000014 0		287C ptrace		\\sample1\ossch\OSTaskForceDispatch+0x8E	0.385us 🔺
0	lwzx	r0,r29,r31			E
0	se_bclri	r0,0x8	;	r0,8	-
0	stwx	r0,r29,r31			
0	e_add16i		;	r11,r13,-32720	^
0	lwzx	r0,r11,r31		C 43 33C73	
0	e_add16i	r6,r13,-0x7FA0	;	r6,r13,-32672	
0	lwzx cntlzw	r7,r6,r31 r10.r0			
0	e_slwi	r10,r0		r10,r10,2	
0	lwzx	r5,r10,r7	3	r10,r10,2	
0	e addi	r3.r1.0x8		r3,r1,8	
l o	se stw	r3,0x18(r28)		r3.24(r28)	
ŏ	stwx	r5,r27,r31	3	13,24(120)	
ŏ	e bl	0x913F		OSSetJmp	
-0000000013		BBEE ptrace	,	\\sample1\osioc\OS_OSIocWriteAcrossRef+0xB	A 8.640us
1	se lwz	r3.0x0(r31)	:	commId,0(r31)	
1	se_1hz	r5,0x4(r31)		r5,4(r31)	
1	e_b1	0x19C6		memcpy	-
4					
, ,					

The trace generation is usually stopped before the trace information for the event that caused the trigger is exported.

#### Example:

Advise the NEXUS module to generate a trigger if a write access to the variable hookNmb occurs. Advise TRACE32 to fill another 10% of the trace memory before the trace recording is stopped.

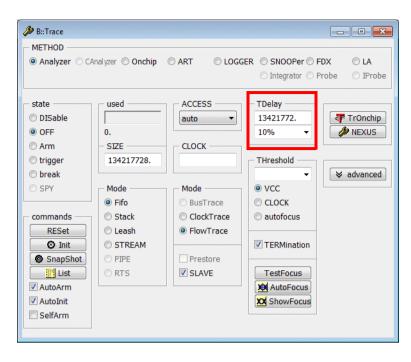
1. Enable Branch Trace messaging and Data Trace messaging for write accesses.

B::NEXUS			
nexus       selection         OFF       Ø BTM         ON       HTM         OTM       WTM         Ø Trace       DQM         III List       DTM         PID_MSR       BL_HTM         TLBNEW       TLBINV	option SmartTrace POTD STALL OFF Suppression SpenDQM SpenWTM SpenPTM SpenDTM SpenOTM SupprTHReshold 1/4	Configuration PortSize MDO4 V PortMode 1/2 V DDR	CLIENT1 SELECT NONE MODE OFF SELECT NONE MODE OFF

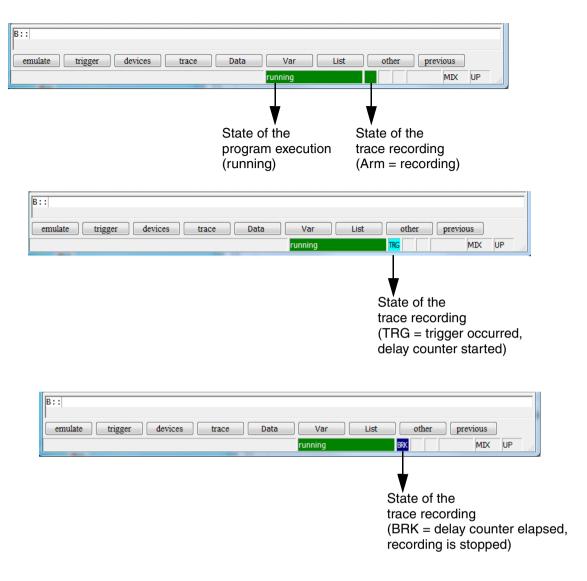
2. Set a Write breakpoint to the variable hookNmb and select the action TraceTrigger.

👹 B::Break.Set			- • • ×
address / expres	ssion		
hookNmb			🔻 🚺 🗹 HLL
type	options		implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	TraceTrigger 💌
Write	- DATA		
🔘 default		<b>•</b>	Sector Secto
L			
Ok	Add	Delete	Cancel

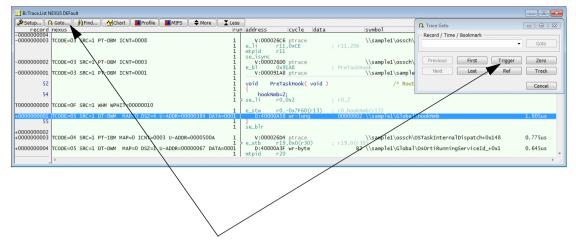
3. Define the trigger delay in the *Trace Configuration* Window.



4. Start the program execution.



### 5. Display the result.



Push the *Trigger* button in the *Trace Goto* window to find the record, where the trigger occurred (WHM message). Here the sign of the record numbers changed. The specified event is usually exported shortly after this point.

Resource: Watchpoints and logic in NEXUS Adapter (parallel trace only)

#### Controlled message types

WTM	втм	DTM	ОТМ	DQM
Watchpoint Hit Message(s) is generated for the specified instruction(s) or data address+data value	Unaffected	Unaffected	Unaffected	Unaffected

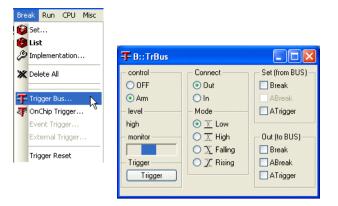
**Example:** Generate a 100 ns high pulse on the trigger connector of the POWER TRACE / ETHERNET or POWER DEBUG II when 3 is writes to hookNmb.

1. Set a write breakpoint to the variable flags[9] and select the action BusTrigger.

😻 B::Break.Set			
address / express [flags[9]	ion		V 👔 V HLL
type Program ReadWrite Read Write any	options EXclude NOMARK - DATA	Temporary	implementation auto
Ok	Set	Delete	Cancel
🕲 B::Break.Lis	t		

😫 B::Break.List			
Delete All     O Disable     O Enable     address	e 🛛 Init 🔑 Select tupes impl	Store SLoad	😻 Set
C:00007409			\\Diabp826\Global\flags[9]
	<		

- 2. Start the program execution.
- 3. Open the *TrBus* window to watch the trigger.



Resource: Watchpoints and logic in NEXUS Adapter (parallel trace only). Only one event possible.

### Controlled message types

WTM	BTM	DTM	ОТМ	DQM
Watchpoint Hit Message(s) is generated for the specified instruction(s) or data address+data value	Unaffected	Unaffected	Unaffected	Unaffected

**Example 1:** Count how often the function sieve is called.

Set a program breakpoint to the start address of the function sieve and select the action 1. BusCount.

address / express	ion ———			
sieve			🖌 🛐 🗌 HLL	
type	- options		_ implementation _	
<ol> <li>Program</li> </ol>	EXclude	Temporary	auto 💌	
◯ ReadWrite	NOMARK	DISable	- action	
🔿 Read		DISableHIT	BusCount 💌	
🔿 Write	- DATA			
🔿 any		<b>~</b>		
	Set	Delete	Cancel	
Ok	Set	Delete	Lancer	
B::Break.Lis	t			
🖉 Dalata All	isable 🔘 Enable	😡 Init 🖉 Selec	t 🔁 Store 🗟 Lo	ad 🛛 📸 Set

2. Open the TRACE32 counter window and select EventHigh.

Misc Trace Perf Cov MPC Veco Oscillator Frequency Counter Pulse Generator	& B::Count					
Pulse Generator 2     Runtime     Memory Map	0	0.	evt_	_· IAC1	(*1)	LOW
Flash Programming     Choose Colors     Chorse Colors     Japaneese Menu	Mode Frequency Period PulsLow PulsHigh EventLow EventHigh	Gate           0.01 s           1 s           10 s           endless	Init AutoInit OUT	Select PODBUS iAC1 IAC2 IAC3 IAC4 DAC1 DAC1 DAC2 DCNT1 DCNT2 MCK0		

3. Start the program execution and display the result.

🖧 B::Count		
0 38.605	evt IAC1	(*1)
Mode     Gate       Frequency     0.01 s       Period     0.1 s       PulsLow     1 s       PulsHigh     10 s       EventLow     endless       EventHigh     variable	init Select PDDBUS AutoInit IAC1 OUT IAC2 OUT IAC3 OUT IAC4 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1 OAC2 OAC1	

**Example 2:** Measure the period in which the function sieve is called.

1. Set a program breakpoint to the function sieve and select the action BusCount.

🗳 B::Break.Set				
- address / expressi sieve	ion ———		V 👔 🗆 HLL	
type Program ReadWrite Read Write any Ok	options EXclude NOMARK DATA Set	Temporary DISable DISableHIT	implementation auto v action BusCount v V advanced	
😵 B::Break.List	t			
X Delete All O D	isable) (© Enable) (© address types C:00003370 (Progr	s im	ct Store Store npl action NCHIP BusCour	

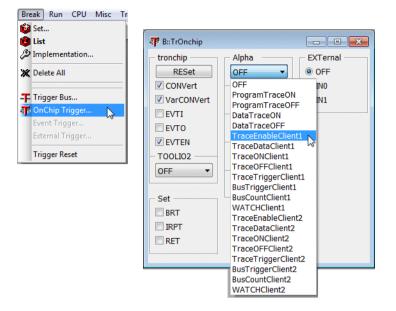
2. Open the TRACE32 counter window and select Period.

& B::Count					
of Britonin					
		ns	IAC1	(*100)	LOW
Mode	- Gate	init	Select		
Frequency	○ 0.01 s	Init	O PODBUS		
Period	⊙ 0.1 s	AutoInit	<ul> <li>IAC1</li> </ul>		
RulsLow			O IAC2		
O PulsHigh	O 10 s	_ out	O IAC3		
◯ EventLow	endless	ООТ	O IAC4		
◯ EventHigh	- variable		O DAC1		
◯ EventHOld			O DAC2		
			O DCNT1		
			O DCNT2		
			О мско		
	<ul> <li>Period</li> <li>VulsLow</li> <li>PulsHigh</li> <li>EventLow</li> <li>EventHigh</li> </ul>	Mode     Gate       Frequency     0.01 s       Period     0.1 s       SulsLow     1 s       PulsHigh     10 s       EventLow     endless       EventHigh     variable	Mode Frequency Period PulsHigh EventLow EventHigh Gate 0.01 s 0.1 s 1 s 0 ut 0 ut 0 UT 0 UT	ns IAC1 Mode Frequency Period PulsHigh EventLow EventHDId Gate 0.01 s 0.1 s 1 s 0 1 s 0 UT 0 LAC2 0 IAC3 0 UT 0 IAC4 0 DAC1 0 DAC2 0 DCNT1 0 DCNT2	ns IAC1 (*100) Mode Frequency Period QuisLow PulsHigh EventH0ld Gate 0.01 s 0.1 s 0.1 s 0.1 s 0.1 s 0.01 c 0.01 s 0.01 s 0.01 c 0.01 s 0.01 c 0.01 s 0.01 c 0.01 s 0.01 c 0.01 s 0.01 c 0.01 c 0.00

3. Start the program execution and display the result.

🖧 B::Count					
0 89	.200	us	IAC1	(*100)	
Mode Frequency Period PulsLow PulsHigh EventLow EventHigh EventHOld	Gate 0.01 s 0.1 s 1 s 0 l0 s 0 endless variable	Init AutoInit	Select PODBUS IAC1 IAC2 IAC3 IAC4 DAC1 DAC1 DAC2 DCNT1 DCNT2 MCK0		

The filter and trigger feature for the Trace Clients are provided via the TrOnchip window.



Trace Clients have their own resources in the NEXUS module. E.g. DMA client on MPC5554.

A TRACE32 PowerView for PowerPC 0 [Power Trace Ethernet @]	- <b>D</b> X
File Edit View Var Break Run CPU Misc Trace Perf Cov MPC5XXX Window Help	
H ⊨   ↓ √ Ċ   ▶ II   ⊠  % №   ◎     ≣ ⊞   📾 📾 🚳 🚳 ½ ≫	
S:NEXUS.Register	
⊕ <u>Nexus Port Controller (NPC)</u>	^î
⊞ <u>Core Trace Control and Nexus Registers</u>	
□ DMA_A Trace Control Registers DC1 A0000012 OPC full MDIV 1/2 EOC WP	
WEN         dis         EIC         DR         DT-TM         ena           DC2         00000000         EWC3         no         EWC2         no           WT         00000000         DTE         dis         DTE         dis           DTC         C0000000         RWT1         r.w         RWT2         dis         RC1         in         RC2         in	
DTC C0000000 RWT1 rw RWT2 dis RC1 in RC2 in DTSA1 00000000 DTSA2 00000000	E
DTEA1 FFFFFFF DTEA2 0000000	
BWC1 00000000 BWE1 dis BRW1 read BWR1 off BWT1 res. BWC2 00000000 BWE2 dis BRW2 read BWR2 off BWT2 res.	
BWA1 0000000 BWA2 0000000	
♥ eTPUA Control Register	<b>→</b>
B::	
emulate trigger devices trace Data Var List other previ	ous
DBG:0113000E Data Trace Start Address 1 stopped	ILL UP

Example MPC5554: Sample only DMA reads from address 0x40001000++0FFF.

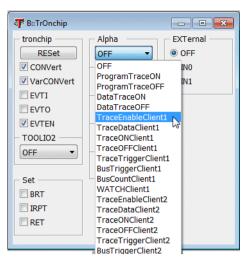
1. Select DMA a Trace Client1

B::NEXUS				
nexus OFF OFF ON RESet Trace List	Selection	option SmartTrace STALL	configuration         PortSize         MD012         PortMode         1/2         DDR         PCRCONFIG	CLIENT1 SELECT MODE ReadWrite • CLIENT2 SELECT NONE MODE OFF •

2. Set a Read breakpoint to the address range 0x40001000++0FFF and select Alpha as breakpoint action.

📦 B::Break.Set			
- address / expres	sion		
C:0x40001000+	+0xfff		▼ 🗾 🗆 HLL
type Program ReadWrite Read Write default	options	Temporary	implementation auto action Alpha advanced
Ok	Add	Delete	Cancel

3. Select TraceEnableClient1 for Alpha in the *TrOnchip* window.



- 4. Start and stop the program.
- 5. Display the result.

B::Trace.Lis	st				
🌽 Setup	Goto	Find	Chart	Profile MIPS 🔷 More	X Less
record	run addre	ess c	ycle da	ta symbol	ti.back
-00000042	D:4	400010C8 r	'd-dma	02102900	5.000us 🔺
-00000040	D:4	400010CC r	·d-dma	02102900	5.000us 📻
-00000038	D:4	400010D0 r	d-dma	02102900	5.000us 🖃
-00000036	D:4	400010D4 r	d-dma	02102900	5.000us
-00000034	D:4	400010D8 r	·d-dma	02102900	5.000us ^
-00000032	D:4	400010DC r	d-dma	02102900	5.000us
-00000030	D:4	400010E0 r	d-dma	02102900	5.000us
-00000028	D:4	400010E4 r	d-dma	02102900	5.000us -
	] (				E. ▲

## Activate the TRACE32 OS Awareness

TRACE32 includes a configurable target-OS debugger to provide symbolic debugging of operating systems.

Since most users use an AUTOSAR operating system, this is taken as an example here.

In order to provide AUTOSAR-aware tracing an ORTI file is required. The ORTI file is created by the AUTOSAR System Builder. It describes the structure and the memory mapping of the operating system objects.

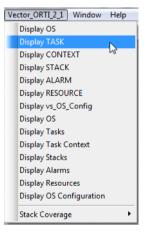
Setup command:

TASK.ORTI <ORTI\_file>

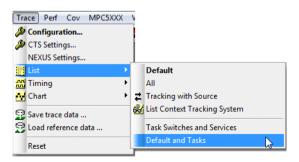
Load the ORTI file into TRACE32

Loading the ORTI file results in the following:

Symbolic debugging of the OSEK OS is possible. Debug commands are provided via an ORTI menu.



The **Trace** menu is extended for OS-aware trace display.



The **Perf** menu is extended for OS-aware profiling.

•

•

•

Pe	rf Cov MPC5XXX Vector_O	RT	I_		
ß	Perf Configuration		Ì		
	Perf List		ł.		
E	Perf List Dynamic		L		
	Function Runtime	۲			
	Distribution	۲	L		
	Duration A to B	۲	L		
	Distance trace records	۲	L		
	Task Runtime	Þ		Prepare	
	Task Services	۲	Ε	Show Numerical	
	Task ISR2s	۲	ė	Show as Timing	hr
	Task Function Runtime	۲	2	Tracking with Trace List	
	Task Status	۲	Г		
	Reset				

The manual of the OS Awareness for OSEK/ORTI is added to the Help menu.

1	Contents
1	index
ŀ	Find
ľ	Tree
1	TRACE32 PowerView User Manual
1	Processor Architecture Manual
1	Debugger User Guide
ľ	MCDS User Guide
ľ	MCDS Trigger Programming
k	Analyzer User Manual
(	ORTI RTOS Debugger Manual
	Timing Analyzer User Manual
F	Power Probe User Manual
	Stimuli Generator User Manual
1	Training Manuals
1	Demo Scripts
١	Welcome to TRACE32

The name of the current task is displayed in the **Task** field of the TRACE32 state line.

I						
B::						
emulate trigger devices trace	Data	Var List PERF	other	previous		
SP:00024638 \\PPC555x_ORTI_OTM_1\osektask\osGetTaskState+0x0C	TaskChain4	stopped (inside line)		HLL	UP	at
		-				
	Task	field				

There are two methods how task information can be generated by the NEXUS hardware module:

### By generating an Ownership Trace Messages

This method should be used if supported by the OSEK operating system. It is the only method for NEXUS Class 2 Modules.

### • By generating trace information for a specific write access

This method requires a NEXUS Class 3 Module. It should be used, if the OSEK operating system does not support Ownership Trace Messages.

# Exporting all Types of Task Information (OTM)

Ownership Trace Messages are generated when the OS updates

- the 8-bit Process ID register (PID0) all compliant standards
- NEXUS PID Register (NPIDR) IEEE-ISTO 5001-2012 compliant NEXUS module

PID0/NPIDR are updated by the OS on

- task switches
- entries and exits to service routines
- starts of ISR2 interrupt service routines and NO\_ISR information

AUTOSAR OSs perform this update since 10/2010.

If you are using a IEEE-ISTO 5001-2003/2008 compliant NEXUS module and your task ID is longer the 8-bit, the PID0 register has to be updated in several steps. This requires special support from your OS. If your OS does not provide this special support, Lauterbach can provide you patch information. Please contact **support@lauterbach.com** for details.

The generation of Ownership Trace Messages has to be enabled within TRACE32.

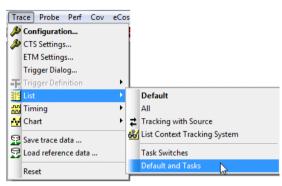
NEXUS.OTM ON ; enable the generation of Ownership Trace ; Messages

#### Example:

1. Advise the NEXUS hardware module to generate only Ownership Trace Messages.

NEXUS.BTM		disable the Branch Trace Messages
NEXUS.OTM		enable the Ownership Trace Messages

- 2. Start and stop the program execution to fill the trace buffer.
- 3. Display the result.



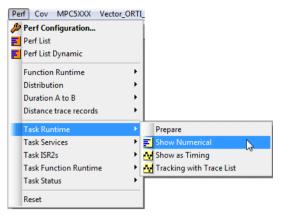
B::Trace.List L	ist.TASK DEFault
🌽 Setup 🚺	🕽 Goto 🏥 Find 🛃 Chart 🔛 Profile 🔛 MIPS 🔷 More 🗶 Less
record	run address cycle data symbol ti.back
	ISR2 = ISR2
-0000000064	
	ISR2 = NO_ISR
-0000000062	
	SERVICE = OSServiceId_ResumeAllInterrupts exit
-0000000060	
0000000000	SERVICE = OSServiceId_ChainTask entry
-000000058	
0000000000	SERVICE = OSServiceId_PostTaskHook entry service 00000085 19.250us
-0000000056	
-0000000054	SERVICE = OSServiceId_GetTaskID entry service 00000029 7.000us
-000000034	SERVICE = OSServiceId_GetTaskID exit
-0000000052	
-0000000032	SERVICE = OSServiceId_PostTaskHook exit
-0000000050	
000000000000000000000000000000000000000	task: TASK1 (0000008)
-0000000048	
0000000000	SERVICE = OSServiceId_ChainTask exit
-0000000046	service 00000024 14.000us -

cycle types	
owner	Ownership trace message for task switches
service	Ownership trace message for entries and exits to OSEK service routines
intr	Ownership trace message for start of OSEK interrupt service routine and NO_ISR information

TRACE32 allows to search for all available cycle types e.g. owner:

B::Trace.Li	st List.TASK DEFault	
	Q Goto     ♪ Find     ♪ Chart     ■ Profile     ■ MIPS     ♦ More)       Irun laddress     Irun laddress     Irun laddress     Irun laddress     Irun laddress	🛐 Trace Find 📃 💷
record	or r0,r1,r0,0x80 ; r0,r0,128 or r0,r1,r0 r1winm r8,r7,0x4,0x12,0x17 ; r8,r7,resetvector	○ Expert
-00681614	or r10,r0,r8 extrwi r6,r12,0x6,0x10 ; r6,r12,6,16 or r3,r10,r6 bl 0x2655C ; osSetRegisterPID24	address / expression
	- mtpid r3 nop srawi r3,r3,0x8 ; r3,r3,8 mtpid r3 nop	Cycle Data
	srawi r3,r3,0x8 ; r3,r3,8 mtpid r3 blr	Find Next Find First Find Here Find All Clear Cancel
-00681611 -00681610		RTI_OTM_1\tcb\osTimerInterrupt_cat2c+0xD4

The following two commands perform a statistical analysis of the task switches:



🎾 Setup 🚺 🏙	Groups 🔚 Cor	nfig) 🗾 Detail	led [ 🔚 Nesting	Chart	Profile		
	tasks: 6.	to	tal: 2.77	'5ms			
range	total	min	max	avr	count	ratio% 1%	2
(unknown)	192.720us	192.720us	192,720us	192.720us	0.	6.944%	
TASK0	267.440us	25.200us	25.800us	24.313us	11.	9.637%	_
TASK4	605.720us	60.100us	61.800us	60.572us	10.	21.826%	_
TASK3	599.360us	59.600us	60.560us	59.936us	10.	21.597%	_
TASK2	875.580us	87.160us	88.060us	87.558us	10.	31.551%	
TASK1	234.300us	23.040us	23.700us	23.430us	10.	8.442%	

TRACE32 assigns all trace information generated before the first task information to the (unknown) task.

record i	run address	cycle	data	sym	bol	ti.back
********						LI DUCK
-						
	GO					
·	SERVICE =		_StartOS ent	ry		
0000001814		service	0000003	-		
	ISR2 = NO_	ISR				
-0000001812		intr	00000005			810.290us
	SERVICE =	0SServiceId	_StartupHook	entry	-	
-0000001810			000000B7	-		115.505us
	SERVICE =	OSServiceId	_StartupHook	exit		
-0000001808		service	00000086			8.500us
-	task	: TASK0 (00	000009) ——			
-0000001806		owner	00000009			29.000us
	SERVICE =	OSServiceId	_StartOS exi	t		
	4					

Perf Cov MPC5XXX Vecto	or_ORTI	<u>i</u>
Perf Configuration		
Perf List		
Perf List Dynamic		
Function Runtime	•	
Distribution	•	
Duration A to B	•	
Distance trace records	•	
Task Runtime	Þ	Prepare
Task Services	Þ	🗾 Show Numerical
Task ISR2s	•	🙀 Show as Timing
Task Function Runtime	•	Tracking with Trace List
Task Status	•	
Reset		

🖗 Setup 🚺 Groups 🔡 Conf	g 📭 Goto	ji)Find ↓ In	MOut KNFull		
	-2.500ms	-2.000ms	-1.500ms	-1.000ms	
r ange 💀					
(unknown) 💀	<b>.</b> <u>.</u>	· · <u>·</u> · · <u>·</u>			
TASK0		<u>.</u>		· · · · · · · · · · · · · · · · · · ·	
TASK4 🖬	. 💻	<u> </u>			
TASK3	· · • • •	· · · · · ·		┫   . ■┫   . ■	
TASK2	💻 .				<b>—</b>
TASK1 🖬					

Trace.STATistic.TASK	Task runtime statistic
Trace.Chart.TASK	Task runtime time chart

The following two commands perform a statistical analysis of the OSEK service routines:

Pe	f Cov MPC5XXX \	/ector_ORTI			
۵	Perf Configuration		1		
F	Perf List				
F	Perf List Dynamic				
	Function Runtime	•			
	Distribution	+			
	Duration A to B	+			
	Distance trace records	•			
	Task Runtime	•			
	Task Services	•	Pre	epare	
	Task ISR2s	+	<b>S</b> h	ow Numerical	
	Task Function Runtime	•	😽 Sh	ow as Timing	~
	Task Status	•	M Tra	acking with Trace List	
	Reset				

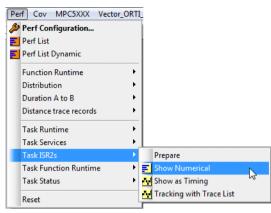
(unknown) represents the time in which the processor/core is not in an OSEK service routine

B::Trace.STATistic.TASKSRV							• ×
🖉 Setup 🚻 Groups 🔡 Config 🗾 Deta	ailed [ 💽 Nestir	ng Chart	Profile				
	srvs: 15.	to	otal: 2.77	'5ms			
range	total	min	max	avr	count	ratio% 1%	2%
(unknown)		-	799.460us	799.460us	0.	28.808%	A
OSServiceId_StartOS	195.820us	195.820us	195.820us	195.820us	1.	6.995%	_
OSServiceId_StartupHook	1.700us	1.700us	1.700us	1.700us	1.	0.061% +	
OSServiceId_PreTaskHook	235.960us	4.460us	4.940us	4.627us	51.	5.648%	=
OSServiceId_GetTaskID	158.340us	1.500us	1.660us	1.568us	101.	5.705%	_
OSServiceId_ActivateTask	158.260us	15.700us	16.100us	15.826us	10.	3.855%	
OSServiceId_PostTaskHook	260.740us	5.000us	5.400us	5.215us	50.	6.543%	
OSServiceId_SuspendAllInterrupts	64.020us	1.500us	2.560us	2.134us	30.	2.306%	
OSServiceId_ResumeAllInterrupts	409.460us	1.540us	20.360us	13.649us	30.	14.754%	<b>T</b>
	•		III				н. 4

Perf Cov MPC5XXX Vector	DRTI_		
🔑 Perf Configuration			
Perf List			
F Perf List Dynamic			
Function Runtime	•		
Distribution	+		
Duration A to B	•		
Distance trace records	•		
Task Runtime	•		
Task Services	Pre	pare	
Task ISR2s	🔸 🗾 Sho	ow Numerical	
Task Function Runtime	🕨 🖬 Sho	ow as Timing	N
Task Status	🕨 🚺 Tra	cking with Trace List	63
Reset			

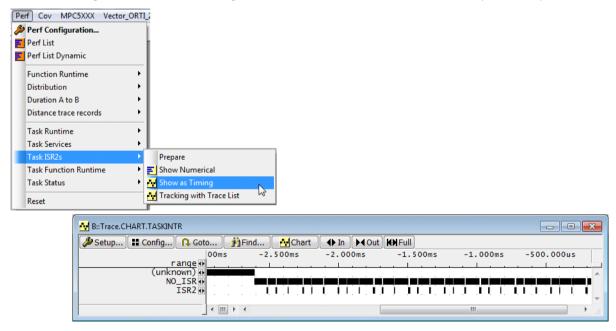
Setup III Config 📭 Goto 👔 Find	Chart	In ► Out	MNEW				
Setup ] To Coning ] Goto ] FIFind							
	-2.460ms	-2.440ms	-2.420ms	-2.400ms	-2.380ms	-2.360ms	-2.
r ange 🚯							
(unknown) 💀							
OSServiceId_StartOS							11
OSServiceId_StartupHook							
OSServiceId_PreTaskHook		· · · · · · · · · · · · · · · · · · ·					
OSServiceId_GetTaskID							
OSServiceId_ActivateTask							
OSServiceId_PostTaskHook							
SServiceId_SuspendAllInterrupts						· • • • •	
OSServiceId_ResumeAllInterrupts				· · · · · ·		· · · · · · ·	
USSer VICETU_ResumeATTITCerrupts	► 4 III			II II.■.			

The following two commands perform a statistical analysis of the OSEK interrupt service routines:



B::Trace.STATistic.TASKINTR							×
Setup	ig <mark>≣</mark> Detaile intrs: 3.		Chart tal: 2.77				
range (unknown) NO_ISR ISR2	162.060us 2.359ms	162.060us 53.060us		162.060us 78.634us	<u>count</u> 1. 30. 30.	ratio% 1% 5.839% 85.006% 9.154%	-
			m			۴	1

TRACE32 assigns all trace information generated before the first intr information to (unknown).



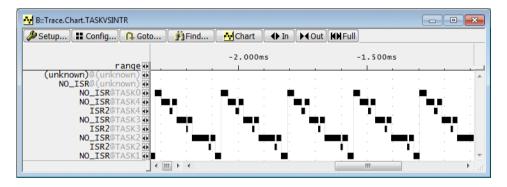
Trace.STATistic.TASKINTR Trace.Chart.TASKINTR Statistic on interrupt service routines

 Trace.Chart.TASKINTR
 Time chart on interrupt service routines

The following command allows to perform a statistical analysis of the OSEK interrupt service routines related to the active tasks.

Trace.STATistic.TASKVSINTRTask-related statistic on interrupt service routinesTrace.Chart.TASKVSINTRTime-chart for task related interrupt service routines

B::Trace.STATistic.TASKVSINTR							
	fig 🗾 Detaile	d [ Nesting	Chart	Nofile			
	intrs: 10.	to	otal: 2.77	5ms			
range	total	min	max	avr	count	ratio% 1%	2
(unknown)@(unknown)	162.060us	162.060us	162.060us	162.060us	1.	5.839%	A
NO_ISR@(unknown)	30.660us	30.660us	30.660us	30.660us	0.	1.104% -	
NO_ISR@TASK0	267.440us	25.200us	25.800us	24.313us	11.	9.637%	_
NO_ISR@TASK4	491.620us	17.500us	31.840us	24.581us	20.	17.715%	_
ISR2@TASK4	114.100us	11.040us	11.660us	11.410us	10.	4.111%	_
NO_ISR@TASK3	529.340us	17.160us	36.000us	26.467us	20.	19.074%	
ISR2@TASK3	70.020us	6.700us	7.260us	7.002us	10.	2.523%	_
NO_ISR@TASK2	805.660us	17.500us	63.000us	40.283us	20.	29.031%	
ISR2@TASK2	69.920us	6.800us	7.400us	6.992us	10.	2.519%	
NO_ISR@TASK1	234.300us	23.040us	23.700us	23.430us	10.	8.442%	
	4		III				50 F



intr information that was generated before the first task information is assigned to the @(unknown) task.

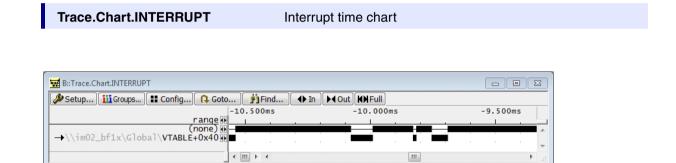
B::Trace.List Li	st.TASK DEFault	- • •
🖉 Setup 🚺	🖡 Goto  🏥 Find 🛛 🚻 Chart 🛛 🌉 Profile 🖉 MIPS 📄 🗢 More	Less
	run address cycle data symbol	ti.back
_*********	GO	*
	SERVICE = OSServiceId_StartOS entry	
-0000001814	service 00000003	
	ISR2 = NO_ISR	<u>^</u>
-0000001812	intr 00000005	810.290us 📃
0000001010	SERVICE = OSServiceId_StartupHook entry	115 505
-0000001810	service 000000B7	115.505us
-0000001808	SERVICE = OSServiceId_StartupHook exit   service 00000086 task: TASK0 (00000009)	8.500us
-0000001806	owner 00000009	29.000us
	SERVICE = OSServiceId_StartOS exit	*
	4	h.

# Exporting all Types of Task Information and all Instructions (OTM)

#### General setup:

NEXUS.BTM ON	; enable the Branch Trace ; Messages
NEXUS.OTM ON	; enable the Ownership Trace ; Messages
Trace.STATistic.InterruptIsFunction ON	; advise TRACE32 to regard the ; time between interrupt entry ; and exit as function

### **Statistic Analysis of Interrupts**



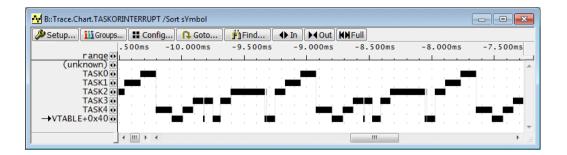
Trace.STATistic.INTERRUPT

Interrupt statistic

E:Trace.STATistic.INTERRUPT							
Setup iii Groups 🔡 Config.	📕 Detailed  📳 Nest	ing 🔤 🔂 Chart	🔼 Profile				
	funcs: 2.	tot	al: 22.625	5ms intr:	2.460ms		
	range total	min m	nax a	avr	count	intern% 1%	2%
→\\imO2_bf1x\Global\ <b>VTABL</b>	(none) 20.164ms E+0x40 2.250ms		20.164ms 71.005us	- 41.006us	- 60.	89.125% 10.874%	*
			III				

### Trace.Chart.TASKORINTERRUPT

Time chart of interrupts and tasks



### Trace.STATistic.TASKORINTERRUPT

Statistic of interrupts and tasks

	ups 🔡 Config tasks: 7.		Nesting		Profile				
	Lasks: /.		Juan: 22.02	1ms					
range	total	min	max	avr	count	ratio%	1%	2%	Ļ
(unknown)	9.705ms	9.705ms	9.705ms	9.705ms	0.	-	+		
TASKO	1.319ms	124.005us	129.005us	119.882us	11.	5.829%			_
TASK1	1.228ms	121.505us	124.255us	122.806us	10.	5.428%			_
TASK2	3.606ms	1.746us	278.010us	120.187us	30.	15.938%			_
TASK3	2.227ms	61.500us	82.005us	74.243us	30.	9.845%			_
TASK4	2.110ms	80.405us	130.755us	105.510us	20.	9.328%			_
♦VTABLE+0x40	2.426ms	11.000us	66.605us	40.436us	60.	10.725%			_

## Trace.Chart.TASKVSINTERRUPT

Time chart interrupts, task-related

🛿 Setup 🛛 🚻 Groups 📲 Config 🚺 Goto 🤅 🏥 F	ind 🚺 🕪 In	MOut KNFull					
	.400ms	-10.200ms	-10.000m	15	-9.800r	15	-9.6
range 💀			 		1		
(none)@(unknown)							
(none)@TASK0							
(none)@TASK1 🚯						· · · ·	
(none)@TASK2						· · · ·	
(none)@TASK3							
(none)@TASK4 🚯							
→\\im02_bf1x\Global\VTABLE+0x40@TASK2							
→\\im02_bf1x\Global\VTABLE+0x40@TASK3							
→\\im02 bf1x\G]oba]\VTABLE+0x40@TASK4 00							

### Trace.STATistic.TASKVSINTERRUPT

Statistic of interrupts, task-related

E: Trace.STATistic.TASKVSINTERRUPT /Sort sYmbol							x
🌽 Setup 👖 Groups 🔡 Config 🗾 Detailed 🖉 N	lesting 🔤 🚹 Ch	iart 🛛 🔣 Profil	e				
	funcs: 9.	to	tal: 22.62	1ms			
range	total	min	max	avr	count	intern% 1%	
(none)@(unknown)	0.000us	-	-	0.000us	0.(1/0)	- +	
(none)@TASK0	0.000us	-	-	0.000us	0. (1/0)	0.000%	
(none)@TASK1	0.000us	-	-	0.000us	0. (1/0)	0.000%	
(none)@TASK2	0.000us	-	-	0.000us	0. (1/0)	0.000%	
(none)@TASK3	0.000us	-	-	0.000us	0. (1/0)	0.000%	
(none)@TASK4	0.000us	-	-	0.000us	0. (1/0)	0.000%	
→\\imO2_bf1x\Global\VTABLE+0x40@TASK2	792.800us	13.250us	67.005us	39.640us	20.	3.504%	-
→\\imO2_bf1x\Global\VTABLE+0x40@TASK3	779.805us	11.500us	66.755us	38.990us	20.	3.447%	-
→\\im02_bf1x\Global\VTABLE+0x40@TASK4	678.035us	20.750us	70.750us	44.465us	20.	3.931%	-
	] ∢ [	·	III			•	. di

### **Task Switches**

Each operating system has a variable that contains the information which task is currently running. One way to export task switch information is to advise the NEXUS hardware module to generate trace information when a write access to this variable occurs.

The address of this variable is provided by the TRACE32 function TASK.CONFIG(magic).

```
PRINT TASK.CONFIG(magic)
```

; print the address of the variable ; that holds the task identifier

Example: Advise the NEXUS hardware module to generate only trace information on task switches.

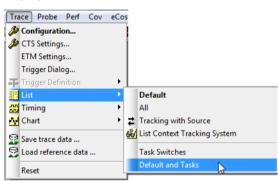
1. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic) and select the trace action TraceEnable.

🔯 B::Break.Set			- • ×
address / expres TASK.CONFIG(n			-
- type	options		implementation —
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
🔘 Read		DISableHIT	TraceEnable 💌
Write	- DATA		
🔘 default			
Ok	Add	Delete	Cancel

Break.Set TASK.CONFIG(magic) /Write /TraceEnable

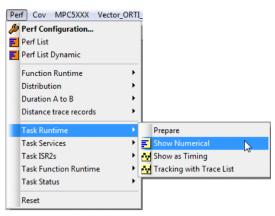
2. Start and stop the program execution to fill the trace buffer

#### 3. Display the result.



🌽 Setup	📭 Goto 🏽 🎒 Find 🛛 🚻 Chart 🛛 🛄 Profile	MIPS 🔶 More 🗶 Less	
record	run address cycle data	symbol	ti.back
-00000021	D:400008A0 wr-word	0008 \\PPC555x_ORTI_OTM_1\Global\os(	trlVars 111.160us
-00000020	D:400008A0 wr-word D:400008A0 wr-word	FFFF \\PPC555x_ORTI_OTM_1\Global\os	trlVars 697.320us
00000019	D:400008A0 wr-word TASK = NO_TASK	0008 \\PPC555x_ORTI_OTM_1\Global\os0	trlVars 110.980us
-0000018	D:400008A0 wr-word TASK = TaskChain4	FFFF \\PPC555x_ORTI_OTM_1\Global\osC	trlVars 697.320us
00000017	D:400008A0 wr-word	0008 \\PPC555x_ORTI_OTM_1\Global\os0	trlVars 111.160us
-00000016	D:400008A0 wr-word TASK = TaskChain4	FFFF \\PPC555x_ORTI_OTM_1\Global\osC	trlVars 697.300us
-00000015	D:400008A0 wr-word	0008 \\PPC555x_ORTI_OTM_1\Global\os0	trlVars 111.000us

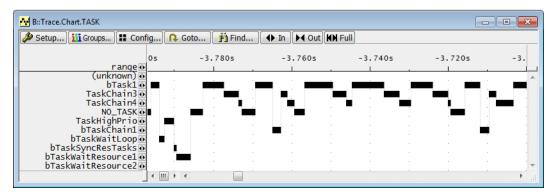
The following two commands perform a statistical analysis of the task switches:



🎾 Setup 🚺 Groups	🖁 Config) 🗾	Detailed [ N	esting 🔄 🚻 Ch	art 🛛 🏋 Profile	e		
	tasks: 13.	to	otal: 7.02	9s			
range			max	avr	count	ratio% 1%	2%
(unknown)	0.000us	0.000us	-	0.000us	0.	0.000%	
NO_TASK	561.324ms	103.660us	1.242ms	155.319us	3614.	7.985%	
bTask1	3.050s	559.320us	7.394ms	5.427ms	562.	43.390%	
bTaskChain1	342.671ms	317.140us	1.430ms	641.707us	534.	4.875%	
TaskChain3	1.446s	337.160us	2.301ms	1.197ms	1208.	20.565%	
TaskChain4	1.458s	242.660us	1.686ms	823.314us	1771.	20.743%	
bTaskWaitLoop	36.029ms	1.402ms	2.396ms	1.638ms	22.	0.512%	
TaskHighPrio	95.511ms	688.800us	1.020ms	782.880us	122.	1.358%	
bTaskSyncResTasks	16.849ms	623.640us	2.172ms	1.532ms	11.	0.239%	
bTaskWaitResource2	10.035ms	293.000us	1.487ms	771.912us	13.	0.142%	
oTaskWaitResource1	11.011ms	806.120us	1.301ms	1.001ms	11.	0.156%	
TaskSec	1.655ms	728.120us	927.120us	827.620us	2.	0.023%	
bTaskChain2	416.980us	416.980us	416.980us	416.980us	1.	0.005% +	

#### Perf Cov MPC5XXX Vector\_ORTI\_

Perf Configuration	
Perf List	
Perf List Dynamic	
Function Runtime	+
Distribution	•
Duration A to B	<b>+</b>
Distance trace records	+
Task Runtime	Prepare
Task Services	Show Numerical
Task ISR2s	Show as Timing
Task Function Runtime	Tracking with Trace List
Task Status	+ [
Reset	



### **OSEK Service Routines**

The time spent in OSEK service routines can be evaluated.

OSEK writes information on the entries and exits to OSEK service routines to a defined variable. One way to export information on OSEK service routines is to advise the NEXUS hardware module to generate trace information when a write access to this variable occurs.

The address of this variable is provided by the TRACE32 function TASK.CONFIG(magic\_service).

PRINT TASK.CONFIG(magic\_service)

- ; print the address of the variable
- ; that holds the service
- ; information

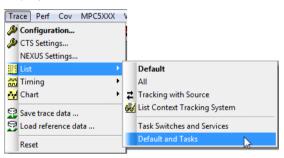
**Example:** Advise the NEXUS hardware module to generate only trace information for entries and exits to OSEK service routines.

1. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic\_service) and select the trace action TraceEnable.

🔯 B::Break.Set			- • •
address / expres			
TASK.CONFIG(m	agic_service)		▼ 🗾 🗆 HLL
- type	options		- implementation
Program	EXclude	Temporary	auto 👻
ReadWrite	NOMARK	DISable	- action
🔘 Read		DISableHIT	TraceEnable 🔻
Write	DATA		
O default		<b>•</b>	
Ok	Add	Delete	Cancel

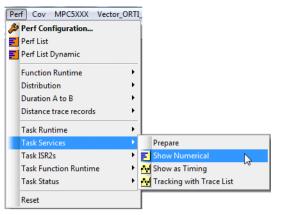
Break.Set TASK.CONFIG(magic\_service) /Write /TraceEnable

- 2. Start and stop the program execution to fill the trace buffer
- 3. Display the result.



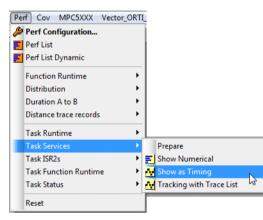
B::Trace.List List.TASK DEFaul	t	- • •
🌽 Setup 🔃 🗘 Goto 🎼	)Find 🛛 🚻 Chart 📔 🎦 Profile 🖉 🏧 MIPS 📔 🗢 More 🖉 🗶 Less	
record run address	cycle data symbol	ti.back
-00000028   D:400 SERVICE -00000027   D:400	= SetRe Alarm entry 1008E0 wr-byte 31 \\PPC555x_ORTI_OTM_1\Global\osORTICurrentServiceId = SetRe Alarm exit 30 \\PPC555x_ORTI_OTM_1\Global\osORTICurrentServiceId = ReleaseResource entry	51.500us
-00000026   D:400 SERVICE	<pre>- reasenessing of the service o</pre>	44.000us
		tte 4

The following two commands perform a statistical analysis of the OSEK service routines:



(unknown) represents the time in which the processor/core is not in an OSEK service routine

E B::Trace.STATistic.TAS	KSRV						
Setup 🚺 Groups	Config	E Detailed	Nesting	Chart 🛛 📕 Pro	ofile		
	srvs: 21.	to	tal: 2.42	8s			
range	total	min	max	avr	count	ratio% 1%	2%
(unknown)		-	976.870ms	976.870ms	0.	40.234%	
CancelAlarm	103.910ms	98.140us	811.120us	107.557us	968.	4.270%	
SetRelAlarm	109.591ms	83.160us	771.640us	91.776us	1197.	4.375%	
GetAlarm	70.302ms	72.320us	83.480us	72.626us	968.	2.895%	
GetResource	126.370ms	88.820us	8.198ms	139.101us	917.	4.272%	
ReleaseResource	117.924ms	76.480us	1.020ms	128.738us	916.	4.631%	
Schedule	730.347ms	93.320us	9.798ms	168.248us	4571.	24.279%	
ActivateTask	12.626ms	18.500us	140.500us	44.773us	282.	0.411%	
PostTaskHook	95.835ms	77.820us	89.660us	78.424us	1222.	3.044%	
GetTaskID	19.037ms	7.320us	18.000us	7.649us	2489.	0.784%	
	•		III				+



-2.184000000s -2.18300000 (unknown) CancelAlarm GetAlarm GetAlarm GetResource ReleaseResource ActivateTask DostTaskHook GetTask10 GetTask5tate PreTaskHook	🔡 Config 🔼 Goto	Find	Chart	<b>♦</b> In	►Out	KNFull			
(unknown) CancelAlarm SetRelalarm GetAlarm GetResource ReleaseResource Schedule ActivateTask PostTaskHook GetTaskState PreTaskHook		/	0000s				-2.18	300000	0s
CancelAlarmo SetRelAlarmo GetAlarmo GetResource ReleaseResource Schedule PostTaskHook GetTaskID GetTaskState PreTaskHook									. !
CancelAlarmo SetRelAlarmo GetResource ReleaseResource Schedule ActivateTaskto PostTaskHook GetTaskJDO GetTaskJtot	(unknown)								_
GetAlarmo GetResource ReleaseResource Schedule ActivateTaski PostTaskHooki GetTaskState PreTaskHooki	CancelAlarm	1		1 1					
GetResource ReleaseResource Schedule ActivateTaski PostTaskHook GetTaskID GetTaskState PreTaskHook		1		1 1			· ·		
ReleaseResource	GetAlarm	1		1 1		· •	l' İl		<u> </u>
Schedule of ActivateTaskin ActivateTaskin PostTaskInok of GetTaskIstate PreTaskHook of		1		1 1			· ·		
ActivateTask PostTaskHook GetTaskID GetTaskState PreTaskHook		1		1 1			· ·		
PostTaskHook GetTaskID GetTaskState PreTaskHook					_		<u> </u>		-
GetTaskID GetTaskState PreTaskHook						·   ·			
GetTaskState PreTaskHook									
PreTaskHook 🖸	GetTaskID								
Ennorthood									
ETTOTHOOK M	ErrorHook 🚯								

### **OSEK ISR2s**

The time spent in OSEK interrupt service routine can be evaluated.

OSEK writes information on the start of an interrupt service routine to a defined variable as well as the information NO\_ISR. One way to export information on OSEK interrupt service routine is to advise the NEXUS hardware module to generate trace information when a write access to this variable occurs.

The address of this variable is provided by the TRACE32 function TASK.CONFIG(magic\_isr2).

PRINT TASK.CONFIG(magic\_isr2)

- ; print the address of the variable
- ; that holds the interrupt service
- ; information

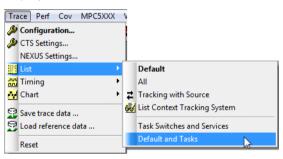
**Example:** Advise the NEXUS hardware module to generate only trace information on the start of an interrupt service routine as well as on the information NO\_ISR.

1. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic\_isr2) and select the trace action TraceEnable.

🔯 B::Break.Set			- • •
address / expres TASK.CONFIG(m			▼ 🗾 🗆 HLL
type	options		- implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
🔘 Read		DISableHIT	TraceEnable 🔻
Write	DATA		
🔘 default		-	
Ok	Add	Delete	Cancel

Break.Set TASK.CONFIG(magic\_isr2) /Write /TraceEnable

- 2. Start and stop the program execution to fill the trace buffer
- 3. Display the result.



🔑 Setup	📭 Goto 🏥 Find 🔂 Chart 🔛 Profile	e More Less	
	run address cycle data	symbol	ti.back
00000017	ISR2 = NO_ISR   D:400008D4 wr-word ISR2 = TestTimer1	FDFF \\PPC555x_ORTI_OTM_1\Global\osActiveISRID	224.500us
00000016	D:400008D4 wr-word ISR2 = N0 ISR	0000 \\PPC555x_ORTI_OTM_1\Global\osActiveISRID	1.243ms
00000015	D:400008D4 wr-word ISR2 = osTimerInterrupt	<pre>FDFF \\PPC555x_ORTI_OTM_1\Global\osActiveISRID</pre>	
00000014	D:400008D4 wr-word ISR2 = NO_ISR	0001 \\PPC555x_ORTI_OTM_1\Global\osActiveISRID	
00000013	D:400008D4 wr-word	<pre>FDFF \\PPC555x_ORTI_OTM_1\Global\osActiveISRID</pre>	96.000us

The following two commands perform a statistical analysis of the OSEK interrupt service routines:

Perf Cov MPC5XXX Vecto	or_ORTI	F	
Perf Configuration			
Perf List			
Perf List Dynamic			
Function Runtime	•		
Distribution	•		
Duration A to B	+		
Distance trace records	+		
Task Runtime	•		
Task Services	+		
Task ISR2s	Þ	Prepare	
Task Function Runtime	+	Show Numerical	N
Task Status	•	Show as Timing	43
Reset		🕂 Tracking with Trace List	

E: B:: Trace.STATistic.TASKINTR							
💋 Setup 🚺 Groups 📰 Config 💽 Detailed 📆 Nesting 🔂 🔂 Chart							
intrs: 4. total: 1.954s							
range	total	min	max	avr	count	ratio% 1%	2
(unknown)	0.000us	0.000us	-	0.000us		0.000%	~
osTimerInterrupt	75.453ms	95.820us	225.500us	128.540us	587.	3.861%	_
NO_ISR	1.575s	94.820us			1060.	80.621%	_
TestTimer1	303.201ms	389.640us	861.460us	639.664us	474.	15.517%	_
							*
	4		III				h. 1997

Perf	Cov MPC5XXX	Vector_ORT	2		
🔑 Per	rf Configuration				
🗾 Per	f List		ŧ.		
🗾 Per	f List Dynamic		L		
Fur	nction Runtime	•	L		
Dis	tribution	+	L		
Du	ration A to B	+	L		
Dis	tance trace record	s 🕨	L		
Tas	k Runtime	+	L		
Tas	k Services	• •	L		
Tas	ik ISR2s	•		Prepare	
Tas	k Function Runtim	ne 🕨	E	Show Numerical	
Tas	k Status	•	M	Show as Timing	N
Res	et		M	Tracking with Trace List	h

B::Trace.CHART.TASKINTR			- • •			
🌽 Setup 🛛 🗮 Config 🚺 Goto 👘 Find 🚺 Chart 🚺 🕪 In 📄 🛏 Out 🕅 Full						
	-1.850s	-1.845s	-1.840s			
range 💀 🚬						
(unknown) 🐽						
osTimerInterrupt 🚯						
NO_ISR						
TestTimer1	💻 .	—				
			<b>T</b>			
			E. 4			

Trace.STATistic.TASKINTR Trace.Chart.TASKINTR Statistic on interrupt service routines

Time chart on interrupt service routines

OSEK interrupt service routines that occur in multiple tasks can be displayed per task, if the following information is available:

- Task switch information
- ISR2 start and NO\_ISR information

#### Example:

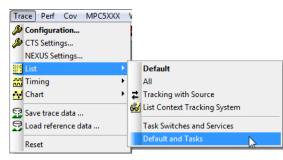
- 1. Advise the NEXUS hardware module to generate only trace information
  - on task switches
  - on the start of an interrupt service routine as well as on the information NO\_ISR

🙆 B::Break.Set			
address / expres			
TASK.CONFIG(m	agic)		▼ 🗾 🗆 HLL
	options		-implementation
Program	EXclude	Temporary	auto 👻
ReadWrite	NOMARK	DISable	- action
Read		DISableHIT	TraceEnable 🔻
Write	DATA		
🔘 default		-	
Ok	Add	Delete	Cancel
😻 B::Break.Set			- • •
address / expres	sion		
TASK.CONFIG(m	agic_isr2)		▼ 🗾 🗆 HLL
type	options		-implementation
Program	EXclude	Temporary	auto 🔻
ReadWrite	NOMARK	DISable	- action
🔘 Read		DISableHIT	TraceEnable 💌
Write	DATA		
🔘 default		-	
Ok	Add	Delete	Cancel

Break.Set TASK.CONFIG(magic) /Write /TraceEnable
Break.Set TASK.CONFIG(magic\_isr2) /Write /TraceEnable

2. Start and stop the program execution to fill the trace buffer.

#### 3. Display the result.



🌽 Setup	🔃 🖓 Goto 🎁 Find 🔂 Chart 🔛 Profile	MIPS Amore Less	
record	run address cycle data	symbol	ti.back
-00000025	ISR2 = NO_ISR   D:400008D4 wr-word TASK = NO_TASK	FFFF \\PPC555x_ORTI_OTM_1\Global\osActiveISRID	561.480us
-00000024	D:400008A0 wr-word TASK = TaskChain4	FFFF \\PPC555x_ORTI_OTM_1\Global\osCtrlVars	554.300us
-00000023	D:400008A0 wr-word TASK = NO_TASK	0008 \\PPC555x_ORTI_OTM_1\Global\osCtrlVars	111.180us
00000022	D:400008A0 wr-word TASK = TaskChain4	FFFF \\PPC555x_ORTI_OTM_1\Global\osCtrlVars	680.460us
-00000021	D:400008A0 wr-word ISR2 = osTimerInterrupt	0008 \\PPC555x_ORTI_OTM_1\Global\osCtrlVars	111.320us
00000020	D:400008D4 wr-word ISR2 = NO_ISR	0001 \\PPC555x_ORTI_OTM_1\Global\osActiveISRID	233.660us
-00000019	D:400008D4 wr-word	FFFF \\PPC555x_ORTI_OTM_1\Global\osActiveISRID	96.160us

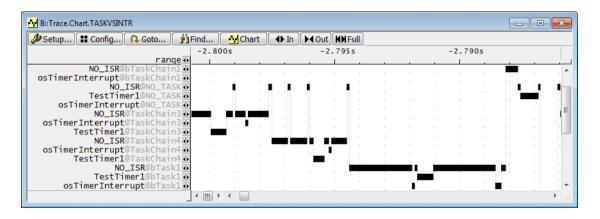
The following command allows to perform a statistical analysis of the OSEK interrupt service routines related to the active tasks.

Trace.STATistic.TASKVSINTR

Trace.Chart.TASKVSINTR

Task-related statistic on interrupt service routines Time-chart on task related interrupt service routines

B::Trace.STATistic.TASKVSINTR								
🌽 Setup 🚺 Groups 🔡 Config ⋛ Detailed	Nesting	Chart 🛛	Profile					
	intrs: 29.	to	otal: 1.94	0s				
range	total	min	max	avr	count	ratio% 1	% 2%	
(unknown)@(unknown)	0.000us	0.000us	-	0.000us	1.	0.000%		
TestTimer1@(unknown)	616.800us	616.800us	616.800us	616.800us	1.	0.031% +		
NO_ISR@(unknown)	932.800us	198.320us	734.480us	932.800us	1.	0.048% +		
osTimerInterrupt@(unknown)	95.980us	95.980us	95.980us	95.980us	1.	0.004%		
NO_ISR@NO_TASK	99.336ms	75.000us	114.480us	108.801us	913.	5.120%		===
NO_ISR@TaskChain3	393.778ms	94.980us	1.340ms	748.628us	526.	20.300%		_
TestTimer1@TaskChain3	73.057ms	393.980us	830.460us	603.780us	121.	3.766%		_
NO_ISR@TaskChain4	274.230ms	94.820us	781.140us	497.695us	551.	14.137%		
osTimerInterrupt@TaskChain4	11.723ms	95.980us	96.320us	96.087us	122.	0.604% 🗲		
NO_ISR@bTask1	670.648ms	94.820us	3.238ms	1.040ms	645.	34.573%		
TestTimer1@bTask1	141.549ms	402.660us	860.960us	711.300us	199.	7.297%		
osTimerInterrupt@bTask1	45.908ms	95.820us	348.480us	162.219us	283.	2.366%		
NO_ISR@bTaskChain1	79.392ms	159.480us	544.820us	441.066us	180.	4.092%		
osTimerInterrupt@TaskChain3	11.241ms	95.980us	96.320us	96.074us	117.	0.579% +		
NO_ISR@TaskHighPrio	24.776ms	153.320us	698.800us	527.146us	47.	1.277%	-	*
	•		III					16. A 222



ISR2 information that was generated before the first TASK information is assigned to the @(unknown) task.

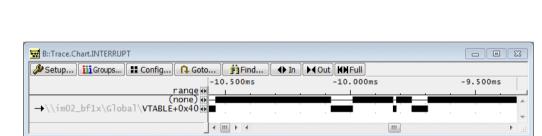
B::Trace.Lis	t List.TASK DEFault			×
🌽 Setup	📭 Goto 🏽 🛱 Find 🔂 Chart 🖉 Profile	MIPS More Less		
	run address cycle data	symbol	ti.back	
-********				*
	GO			
-00003975	ISR2 = TestTimer1   D:400008D4 wr-word ISR2 = NO ISR	0000 \\PPC555x_ORTI_OTM_1\Global\osActiveISRID		-
-00003973	D:400008D4 wr-word ISR2 = osTimerInterrupt	<pre>FFFF \\PPC555x_ORTI_OTM_1\Global\osActiveISRID</pre>	616.800us	_
-00003972	D:400008D4 wr-word ISR2 = NO ISR	0001 \\PPC555x_ORTI_OTM_1\Global\osActiveISRID	734.480us	
-00003971	D:400008D4 wr-word TASK = NO_TASK	<pre>FFFF \\PPC555x_ORTI_OTM_1\Global\osActiveISRID</pre>	95.980us	
-00003970	D:400008A0 wr-word	FFFF \\PPC555x_ORTI_OTM_1\Global\osCtrlVars	198.320us	Ψ.
	<		Þ	

General setup:

```
Break.Set TASK.CONFIG(magic) /Write /TraceData
; advise TRACE32 to regard the time between interrupt entry
; and exit as function
Trace.STATistic.InterruptIsFunction ON
```

## **Statistic Analysis of Interrupts**

Trace.Chart.INTERRUPT



Trace.STATistic.INTERRUPT

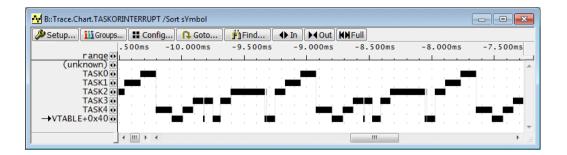
Interrupt statistic

Interrupt time chart

B::Trace.STATistic.INTERRUPT					
Setup iii Groups 🔡 Config	.) 📕 Detailed 🖉 Ne	esting 🛛 🛺 Chart 🗍	💶 Profile		
	funcs: 2	. tota	1: 22.625ms	intr: 2.460ms	
	range total	min ma:		count	intern% 1% 2%
→\\imO2_bf1x\Global\VTABL	(none) 20.164 E+0x40 2.250		20.164ms 71.005us 41	.006us 60.	- 89.125%
- ((THIO2_DI1X(GIODAT(VTABL	2.230	ms   11.250us   .	/1.005us   41	.00605   60.	.   10.8/4%
	•		III		

### Trace.Chart.TASKORINTERRUPT

Time chart of interrupts and tasks



### Trace.STATistic.TASKORINTERRUPT

Statistic of interrupts and tasks

<pre>   Setup II Groups E Config E Detailed IN Nesting A Chart Profile   tasks: 7. total: 22.621ms </pre>											
range i		min 9.705ms		avr 9.705ms	count	ratio%	1%	2%	_		
(unknown) TASK0	9.705ms 1.319ms	124.005us	129.005us	119.882us	0.	5.829%	+		_ ^		
TASK1	1.228ms	121.505us	124.255us			5.428%			-1		
TASK2	3.606ms	1.746us	278.010us		30.	15.938%			-		
TASK3	2.227ms	61.500us			30.	9.845%			-		
TASK4	2.110ms	80.405us			20.	9.328%			-		
→VTABLE+0x40	2.426ms	11.000us	66.605us	40.436us	60.	10.725%			-		

## Trace.Chart.TASKVSINTERRUPT

Time chart interrupts, task-related

🛿 Setup 🛛 🚻 Groups 📲 Config 🦳 Goto 🤅 🏥 F	ind 🚺 🕪 In	MOut KNFull					
	.400ms	-10.200ms	-10.000m	15	-9.800r	15	-9.6
range 💀			 		1		
(none)@(unknown)							
(none)@TASK0							
(none)@TASK1 🚯						· · · ·	
(none)@TASK2 🚯						· · · ·	
(none)@TASK3							
(none)@TASK4 🚯							
→\\im02_bf1x\Global\VTABLE+0x40@TASK2							
→\\im02_bf1x\Global\VTABLE+0x40@TASK3							
→\\im02 bf1x\G]oba]\VTABLE+0x40@TASK4 00							

### Trace.STATistic.TASKVSINTERRUPT

Statistic of interrupts, task-related

E: Trace.STATistic.TASKVSINTERRUPT /Sort sYmbol							x		
🥟 Setup 🚺 Groups 📳 Config 💽 Detailed 🛛 🋐 Nesting 🗋 🔂 Chart 🗋 🔤 Profile 🕽									
funcs: 9. total: 22.621ms									
range	total	min	max	avr	count	intern% 1%			
(none)@(unknown)	0.000us	-	-	0.000us	0.(1/0)	- +			
(none)@TASK0	0.000us	-	-	0.000us	0. (1/0)	0.000%			
(none)@TASK1	0.000us	-	-	0.000us	0. (1/0)	0.000%			
(none)@TASK2	0.000us	-	-	0.000us	0. (1/0)	0.000%			
(none)@TASK3	0.000us	-	-	0.000us	0. (1/0)	0.000%			
(none)@TASK4	0.000us	-	-	0.000us	0. (1/0)	0.000%			
→\\imO2_bf1x\Global\VTABLE+0x40@TASK2	792.800us	13.250us	67.005us	39.640us	20.	3.504%	-		
→\\imO2_bf1x\Global\VTABLE+0x40@TASK3	779.805us	11.500us	66.755us	38.990us	20.	3.447%	-		
→\\im02_bf1x\Global\VTABLE+0x40@TASK4	678.035us	20.750us	70.750us	44.465us	20.	3.931%	-		
	] ∢ [	·	III			•	. di		

The TRACE32 Instruction Set Simulator can be used for a belated OS-aware trace evaluation. To set up the TRACE32 Instruction Set Simulator for belated OS-aware trace evaluation proceed as follows:

1. Save the trace information for the belated evaluation to a file.

```
Trace.SAVE belated__orti.ad
```

2. Set up the TRACE32 Instruction Set Simulator for a belated OS-aware trace evaluation (here OSEK on a MPC5553):

```
SYStem.CPU MPC5553
                                            ; select the target CPU
SYStem.Up
                                            ; establish the
                                            ; communication between
                                            ; TRACE32 and the TRACE32
                                            ; Instruction Set
                                            ; Simulator
Trace.LOAD belated orti.ad
                                            : load the trace file
Data.Load.ELF my_app.out /NoCODE /GHS
                                            ; load the symbol and
                                            ; debug information
                                            ; load the ORTI file
TASK.ORTI my orti.ort
Trace.List List.TASK DEFault
                                            ; display the trace
                                            ; listing
```

# Exporting all Types of Task Information (OTM)

Ownership Trace Messages are generated when the OS updates

- the 8-bit Process ID register (PID0) IEEE-ISTO 5001-2003 compliant NEXUS module
- NEXUS PID Register (NPIDR) IEEE-ISTO 5001-2008 compliant NEXUS module and subsequent standards

PID0 respectively NPIDR is updated on

- task switches
- entries and exits to OSEK service routines
- start of OSEK interrupt service routines and start of NO\_ISR code

The ORTI standard support task-aware tracing via OTMs since October/2010.

If you are using a IEEE-ISTO 5001-2003 compliant NEXUS Class 2 module and your task ID is longer the 8-bit, the PID0 register has to be updated in several steps. This requires special support from your OSEK system. If your OSEK system does not provide this special support, Lauterbach can provide you patch information. Please contact support@lauterbach.com for details.

The generation of Ownership Trace Messages has to be enabled within TRACE32.

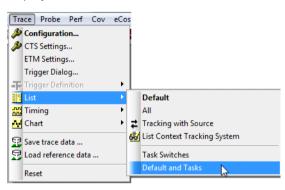
NEXUS.OTM ON ; enable the generation of Ownership Trace ; Messages

#### Example:

1. Advise the NEXUS hardware module to generate only Ownership Trace Messages.

NEXUS.BTM	OFF	'	disable the Branch Trace messaging
NEXUS.OTM	ON		enable the Ownership Trace Messages

- 2. Start and stop the program execution to fill the trace buffer.
- 3. Display the result.



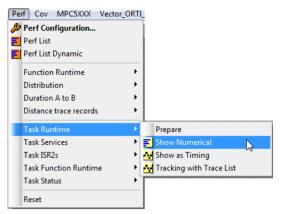
B::Trace.List Li	ist.TASK DEFault		×
setup	🖡 Goto  🐴 Find 🛛 🚻 Chart 🖉 🎬 Profile 🖉 🎽 MIPS 🛛 🗢 More 🗌	Less	
	run address cycle data symbol	ti.back	
-0000006824		6.780us	
	SERVICE = OSServiceId_StartOS entry		
-0000006822	0 service 00000003	5.425us	-
	ISR2 = NO_ISR		
-0000006820	0  intr 0000005	2.044ms	
	———— task: TASKRCV1 (00000001) —————		-
-0000006817	0 owner 00000001	365.180us	
	SERVICE = OSServiceId_StartOS exit		
-0000006815	0 service 00000002	6.535us	
	SERVICE = OSServiceId_PreTaskHook entry		
-0000006813	0 service 000000B3	2.340us	
	SERVICE = OSServiceId_PreTaskHook exit		
-0000006811	0  service 000000B2	3.080us	· •
	•		►
-0000006813	or SERVICE = OSSERVICE OUCOUSS SERVICE = OSSERVICEI_PreTasKHook exit 0  service 000000B2  ∢	2.3400s 3.080us	- ►

cycle types	
owner	Ownership trace message for task switches
service	Ownership trace message for entries and exits to OSEK service routines
intr	Ownership trace message for start of OSEK interrupt service routine and start of NO_ISR code

TRACE32 allows to search for all available cycle types e.g. owner:

B::Trace.List List.T/	ASK DEFault						
Setup 🔒 Go	to 👘 Find	Chart	Profile	MIPS	♦ More	Less	
	n address			sy	rmbol		ti.back
-0000006828 0	- SERVICE = 0x		000000D6 entry				1.725us 🔺
-0000006826 0		service	000000D9	💾 Trace Fi	nd		
-0000006824 0		service	000000D8	© Expert	Occupies Cycle	e 💿 Grou	p 🔘 Changes 🔘 Up
-0000006822 0	- SERVICE = OS						🔘 Signal 🛛 💿 Down
-0000006820 0	- ISR2 = NO_IS	R intr	00000005	address	/ expression –		
	task:	TASKRCV1					▼ <u>≟</u> 🗆 HLL
-0000006817 0	SERVICE = 05	owner	00000001 Start05 ex				
-0000006815 0		service	00000002	– Cycle –		Data	
-0000006813 0	- SERVICE = OS	ServiceId service		OWNER	-		▼
	SERVICE = OS						
-0000006811 0	SERVICE = 05	service ServiceId		Find Nex	t Find First	Find Here	Find All Clear Cancel
				I Ind Nex	i ind i i sc	( Ind field	rind /un cical cancer

The following commands perform a statistical analysis of the task switches:



B::Trace.STATistic.	TASK							x
🌽 Setup 🚺 Grou	ups 🔡 Config	g 🗾 Detailed	Nesting	Chart	Profile			
	tasks: 10.	to	tal: 7.06	2s				
range	total	min	max	avr	count	ratio% 1%	2%	
(unknown):0	32.946ms	32.946ms	32.946ms	32.946ms	0.	0.466% 🗲		
TASKRCV1:0	114.392ms	257.035us	591.755us	272.362us	420.	1.619%		
NO_TASK0:0	6.877s	117.565us	236.532ms	10.174ms	676.	97.379%		_
TASKRCV2:0	32.980ms	139.730us	139.770us	139.748us	236.	0.466% +		
TASKSTOP:0	4.782ms	214.380us	263.795us	239.081us	20.	0.067% +		
(unknown):1	33.434ms	33.434ms	33.434ms	33.434ms	0.	0.473%		
TASKSND1:1	84.235ms	209.985us	220.475us	210.587us	400.	1.192% -		
NO TASK1:1	6.806s	1.184ms	237.175ms	5.671ms	1200.	96.362%		
TASKSND2:1	76.925ms	85.335us	266.645us	192.313us	400.	1.089% -		
TASKCNT:1	62.284ms	145.030us	457.955us	155.709us	400.	0.881% +		- 100
	•		111					•

Trace.STATistic.TASK [/SplitCORE]

Task runtime statistic, result per core

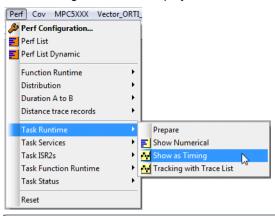
B::Trace.STATistic.	TASK /MergeCO	RE							×
🖉 Setup 🚺 Gro	ups 🔡 Config	j 🗾 Detailed	Nesting	Chart	Profile				
	tasks: 10.	to	tal: 7.06	52s					
range	total	min	max	avr	count	ratio%	1%	2%	
(unknown):0	32.946ms	32.946ms	32.946ms	32.946ms	0.	0.233%	+		
(unknown):1	33.434ms	33.434ms	33.434ms	33.434ms	0.	0.236%	+		
TASKRCV1	114.392ms	257.035us	591.755us	272.362us	420.	0.809%	+		
TASKSND1	84.235ms	209.985us	220.475us	210.587us	400.	0.596%	<b>+</b>		
NO_TASK0	6.877s	117.565us	236.532ms	10.174ms	676.	48.689%			_
NO_TASK1	6.806s	1.184ms	237.175ms	5.671ms	1200.	48.181%			_
TASKSND2	76.925ms	85.335us	266.645us	192.313us	400.	0.544%	+		
TASKRCV2	32.980ms	139.730us	139.770us	139.748us	236.	0.233%	<b>+</b>		
TASKCNT	62.284ms	145.030us	457.955us	155.709us	400.	0.440%			
TASKSTOP	4.782ms	214.380us			20.	0.033%			-
	•			•					F

#### Trace.STATistic.TASK /MergeCORE

Task runtime statistic, results of all cores merged

TRACE32 assigns all trace information generated before the first **task** information to the **(unknown)** tasks. The **(unknown)** tasks are always displayed per core.

The following commands display a time-chart of the task run-times:



Setup 🚻 Groups 📲 Config	📭 Go	to ]	<b>F</b>	Find	.][	) In		Out	HM	Full	]										
s	-5.	955s		-5.	950s		- 5	. 94	5 s		-5.	940	s		-5.	935	s		- 5	. 93	30s
r ange 💀								1 .			. 1									1	
(unknown): 🛛 💀																					
TASKRCV1:0																					
NO_TASK0:0																					
TASKRCV2 : 0															Ĩ						
TASKSTOP : 0																					
(unknown):10														5.18							
TASKSND1:10				10.01				1													
NO_TASK1:1				i di				1		1			Ĩ.	1							
TASKSND2 : 1			100	1000			225					100						1		1	
TASKCNT:10			0.2	1.00															1		

Trace.Chart.TASK [/SplitCORE]

Task runtime time chart, result per core

B::Trace.Chart.TASK /MergeCORE						
🎾 Setup 🛛 🏭 Groups 🖉 📲 Conf	ig 🖪 Goto	. Find	▲ In ▲Out	KNFull		
	-5.948s	-5.947s	-5.946s	-5.945s	-5.944s	-5.943s
r ange 💀						
(unknown):0 💀						
(unknown):1 💀						
TASKRCV1 🚸						
TASKSND1 💀						
NO_TASK0						
NO_TASK1 •						
TASKSND2		· · · <u>· · ·</u> · · ·				
TASKRCV2		· · · · · · · · · · · · · · ·				
TASKCNT		<mark></mark>	· · · · · · · · · · ·			
TASKSTOP			· · · · · · · · · · · · · · ·			
	4 <u>m</u> + 4					4

Trace.Chart.TASK /MergeCORE

Task runtime time chart, results of all cores merged

The following commands perform a statistical analysis of the OSEK service routines:

_		_			
Pe	f Cov MPC5XXX Vector_OR	П_			
ß	Perf Configuration				
E	Perf List				
E	Perf List Dynamic				
	Function Runtime				
	Distribution				
	Duration A to B	·			
	Distance trace records	·			
	Task Runtime		L		
	Task Services			Prepare	
	Task ISR2s		F	Show Numerical	N
	Task Function Runtime	•	~	Show as Timing	N
	Task Status		N	Tracking with Trace List	
	Reset				

(**unknown**) represents the time in which the processor/core is not in an OSEK service routine

B::Trace.STATistic.TASKSRV							
🌽 Setup 👖 Groups 🔡 Config 📰 Detailed 👔	Nesting C	hart 🛛 🔼 Prof	ile				
	srvs: 26.	to	otal: 4.07	'1s			
	_						
range		min		avr	count	ratio% 1%	2%
(unknown):0			4.005s	4.005s	0.	98.377%	*
OSServiceId_PreTaskHook:0	1.698ms	3.350us	4.645us	4.191us	405.	0.041% +	
OSServiceId_GetEvent:0	3.137ms	11.340us	12.505us	12.447us	252.	0.077% +	
OSServiceId_DisableAllInterrupts:0	124.240us	5.025us	5.415us	5.177us	24.	0.003% +	
OSServiceId_EnableAllInterrupts:0	110.575us	4.380us	5.030us	4.607us	24.	0.002% +	
OSServiceId_SetRelAlarm:0		32.345us	127.740us	36.880us	264.	0.239% +	
OSServiceId_StartScheduleTableRel:0	2.795ms	232.880us	232.920us	232.898us	12.	0.068% +	
OSServiceId_ClearEvent:0	2.198ms	8.630us	9.410us	8.723us	252.	0.053% +	
OSServiceId_WaitEvent:0	12.579ms	49.615us	54.780us	49.919us	252.	0.290% +	
OSServiceId_PostTaskHook:0	1.255ms	2.960us	3.480us	3.098us	405.	0.030% +	
OSServiceId_CancelAlarm:0	6.402ms	26.545us	26.685us	26.674us	240.	0.157%	
OSServiceId_GetResource:0	8.688ms	22.035us	24.105us	22.804us	381.	0.213% +	
OSServiceId_ReleaseResource:0	10.158ms	26.160us	27.590us	26.661us	381.	0.249% +	
OSServiceId_TerminateTask:0	6.433ms	41.755us	45.245us	42.043us	153.	0.146% +	
OSServiceId_StopScheduleTable:0	1.885ms	132.620us	181.620us	157.113us	12.	0.046% +	
OSServiceId_GetCounter2Info:0	54.135us	18.045us	18.045us	18.045us	3.	0.001% +	
OSServiceId_StartScheduleTableAsync:0	48.330us	16.110us	16.110us	16.110us	3.	0.001%	
(unknown):1	3.992s	-	3.992s	3.992s	0.	98.070%	
OSServiceId_PreTaskHook:1	3.345ms	4.635us	5.030us	4.646us	720.	0.082% +	
OSServiceId_TerminateTask:1	55.611ms	74.745us	81.470us	77.238us	720.	1.282%	
OSServiceId_PostTaskHook:1	3.415ms	4.635us	4.900us	4.743us	720.	0.083% +	
OSServiceId_GetCounterValue:1	6.724ms	47.680us	47.700us	47.691us	141.	0.150% +	
OSServiceId_ErrorHook:1	617.895us	4.380us	4.385us	4.382us	141.	0.015% +	
OSServiceId_IncrementCounter:1	11.648ms	43.820us	127.745us	48.531us	240.	0.286% +	
OSServiceId_GetCounter2Info:1	186.240us	30.800us	31.320us	31.040us	6.	0.004% +	
OSServiceId_StartScheduleTableAsync:1	1.042ms	173.605us	173.745us	173.695us	6.	0.025% +	
	•		III				h. 10000

Trace.STATistic.TASKSRV [/SplitCORE]

Statistic on service routines, result per core

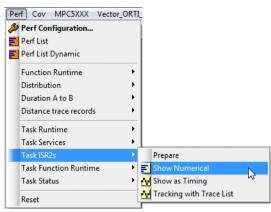
erf Cov MPC5XXX Vecto	or_ORTI	
Perf Configuration		
Perf List		
Perf List Dynamic		
Function Runtime	•	
Distribution	•	
Duration A to B	•	
Distance trace records	•	
Task Runtime	+	
Task Services	•	Prepare
Task ISR2s	•	🗾 Show Numerical
Task Function Runtime	•	🛺 Show as Timing 💦 💦
Task Status	•	Tracking with Trace List
Reset		

B::Trace.Chart.TASKSRV		
🛿 Setup 🔛 Config 📭 Goto 🎽 Find 🦳 🖓 Chart 🗌	♦ In ► Out KH Full	
	-3.28000000s	-3.27
r ange 🕢		
(unknown): 0 🚯		
OSServiceId_PreTaskHook:00		
OSServiceId_GetEvent: 0 🚯		
OSServiceId_DisableAllInterrupts:0		
OSServiceId_EnableAllInterrupts:000		
OSServiceId_SetRelAlarm:030		
OSServiceId_StartScheduleTableRel:0		
OSServiceId_ClearEvent:0		
OSServiceId_WaitEvent:0		
OSServiceId_PostTaskHook:0		
OSServiceId_CancelAlarm:0		
OSServiceId_GetResource:0		
OSServiceId_ReleaseResource:0		
OSServiceId_TerminateTask:0		
OSServiceId_StopScheduleTable:00		
OSServiceId_GetCounter2Info:00		
SServiceId_StartScheduleTableAsync:0		
(unknown):1		
OSServiceId_PreTaskHook:10		· · · · · · · · · · · · · · · · · · ·
OSServiceId_TerminateTask:100		
OSServiceId_PostTaskHook:100		
OSServiceId GetCounterValue:1	—	
OSServiceId_ErrorHook:10		
OSServiceId_IncrementCounter:10		
OSServiceId_GetCounter2Info:1		
SServiceId_StartScheduleTableAsync:10		
• • • • • •		

Trace.Chart.TASKSRV [/SplitCORE]

Time chart on service routines, result per core

The following commands perform a statistical analysis of the OSEK interrupt service routines:



B::Trace.STATistic.TAS	ANTR									x
🌽 Setup 🚻 Groups	Config	E Detailed	Nesting 🔤	Chart 🛛 🔣 Pr	ofile					
	intrs: 8.	to	tal: 28.33	8s						
range	total	min	max	avr	count	ratio%	1%	2%	5%	
(unknown):0	1.061ms	1.061ms	1.061ms	1.061ms	1.	0.003%				
SYSTEM_TIMER:0	426.352ms	28.990us	45.505us	29.091us	14656.	1.504%				
NO_ISR:0	27.800s	207.755us	1.905ms	1.609ms	17282.	98.103%				-
OS_TIMER:0	110.128ms	40.335us	42.930us	41.921us	2627.	0.388%	+			
(unknown):1	0.000us	0.000us	-	0.000us	1.	0.000%				
SECOND_TIMER:1	511.482ms	107.345us	109.580us	108.136us	4730.	1.804%		-		
NO_ISR:1	27.799s	58.890us	236.685ms	5.593ms	4970.	98.099%	-			-
SECOND_TEST:1	27.202ms	76.945us	180.345us	112.871us	241.	0.095%	+			-
	4			III					+	1. d

Trace.STATistic.TASKINTR [/SplitCORE] Statistic on interrupt service routines, result per core

TRACE32 assigns all trace information generated before the first intr information to (unknown).

Perf Cov MPC5XXX Vector	r_ORTI_2
🔑 Perf Configuration	
Perf List	
Perf List Dynamic	
Function Runtime	•
Distribution	•
Duration A to B	•
Distance trace records	+
Task Runtime	•
Task Services	•
Task ISR2s	Prepare
Task Function Runtime	Show Numerical
Task Status	Show as Timing
Reset	Tracking with Trace List

B::Trace.Chart.TASKINTR							, • 🗙	
Setup 🔡 Config 📭 Goto	Find	🕂 Chart 🛛 📢	In MOut KNF	ull				
		-8.	377500000s		-	8.377000	000s	
range 🕖						1		1
NO_ISR:0								
SYSTEM_TIMER:0								
OS_TIMER:0								
(unknown):0								
NO_ISR:1								
SECOND_TIMER:1								
SECOND_TEST:1								
(unknown):1					·			Ŧ
	+ -						١. ١	

Trace.Chart.TASKINTR [/SplitCORE]

Time chart on interrupt service routines, result per core

The following commands allow to perform a statistical analysis of the OSEK interrupt service routines related to the active tasks.

### Trace.STATistic.TASKVSINTR [/SplitCORE]

Task-related statistic on interrupt service routines, result per core

Trace.Chart.TASKVSINTR [/SplitCORE]

Time-chart for task related interrupt service routines, result per core

B::Trace.STATistic.TASKVSINTR							• •
Setup iii Groups II Config.	<b>F</b> Detailed	Nesting	Chart	Profile			
	intrs: 20.		tal: 28.3	385			
range	total	min	max	avr	count	ratio% 1%	2%
(unknown)@(unknown):0	1.061ms	1.061ms	1.061ms	1.061ms	1.	0.003% +	*
SYSTEM_TIMER@(unknown):0	29.000us	29.000us	29.000us	29.000us	1.	<0.001%	
NO_ISR@(unknown):0	639.295us	27.970us	611.325us	639.295us	1.	0.002%	
OS_TIMER@(unknown):0	42.790us	42.790us	42.790us	42.790us	1.	<0.001% +	
NO_ISR@TASKRCV1:0	458.241ms	256.975us	591.815us	272.277us	1683.	1.617%	-
NO_ISR@NO_TASK0:0	27.190s	24.615us	1.905ms	1.360ms	19988.	95.950%	
SYSTEM_TIMER@NO_TASK0:0	426.323ms	28.990us	45.505us	29.091us	14655.	1.504%	
OS_TIMER@NO_TASK0:0	110.085ms	40.335us	42.930us	41.921us	2626.	0.388% +	
NO_ISR@TASKRCV2:0	131.754ms	139.690us	139.740us	139.718us	943.	0.464% +	
NO_ISR@TASKSTOP:0	19.386ms	214.320us	263.735us	239.332us	81.	0.068% +	
(unknown)@(unknown):1	0.000us	0.000us		0.000us	1.	0.000%	
SECOND_TIMER@(unknown):1	107.620us	107.620us	107.620us	107.620us	1.	<0.001% +	
NO_ISR@(unknown):1	47.300us	47.300us	47.300us	47.300us	0.	<0.001% +	
NO_ISR@TASKCNT:1	246.488ms	103.235us	273.015us	149.932us	1644.	0.869% +	
NO_ISR@NO_TASK1:1	26.907s	47.295us	236.591ms	2.763ms	9740.	94.952%	
SECOND_TIMER@NO_TASK1:1	511.375ms	107.345us	109.580us	108.136us	4729.	1.804%	-
NO_ISR@TASKSND1:1	337.522ms	209.930us	220.435us	210.556us	1603.	1.191% -	
NO_ISR@TASKSND2:1	307.721ms	85.315us	266.590us	191.966us	1603.	1.085% -	
SECOND_TEST@NO_TASK1:1	23.938ms	76.945us	180.345us	119.097us	201.	0.084% +	
SECOND_TEST@TASKCNT:1	3.263ms	81.580us	81.600us	81.587us	40.	0.011% +	-
	4		III				н. <b>4</b>

Hart B:: Trace. Chart. TASKVSINTR																					[	_		•		×
Setup 🔡 Config 🔃 Goto		Ĥ	Fin	d	A.	C	nart		•	In			Dut	K	MF	ull										
		. 70	)9s		 -21	7	08	s		- 2	1.	707	7s			- 21	. 7	06	s	-2	21	. 70	) 5 s	5		
range o		ι.			 												ι.			 						
(unknown)@(unknown):00																										
SYSTEM_TIMER@(unknown):0																										
NO_ISR@(unknown):0																										
OS_TIMER@(unknown):0																										
NO_ISR@TASKRCV1:0																										
NO_ISR@NO_TASK0:0																					1					1
SYSTEM_TIMER@NO_TASK0:00			1																			1				
OS_TIMER@NO_TASK0:00																										
NO_ISR@TASKRCV2:0																										
NO_ISR@TASKSTOP:0																										
(unknown)@(unknown):1																										
SECOND_TIMER@(unknown):1																										
NO_ISR@(unknown):1																										
NO_ISR@TASKCNT:1																										
NO_ISR@NO_TASK1:1			1.1						1											Ľ.						
SECOND_TIMER@NO_TASK1:10		22						2R																		
NO_ISR@TASKSND1:10								1																	10.5	
NO_ISR@TASKSND2:10																										
SECOND_TEST@NO_TASK1:10																										
SECOND_TEST@TASKCNT:10																										-
	1.	Ím	1 F						Ċ																*	
J	1	-																								

intr information that was generated before the first task information is assigned to the @(unknown) task.

🌽 Setup 🚺	Goto	Find	Chart	Profile	MIPS	More	Less	]	
record	run add	ress	cvcle	data	sv	/mbo1	t	.back	
_*********	<u> </u>			·					_
_*********									1
	GO	-							
	GO								
-0000294748	1		intr	0000002					
-0000294746	1		intr	00000005				L07.620us	ſ
		— task: T	ASKCNT (	00000004) -					
-0000294744	1		owner	00000004				47.300us	
-0000294742	1		service	00000B3				11.080us	
-0000294740	1		service	00000B2				4.645us	
-0000294738	1		service	00000093				15.205us	
-0000294736	1		service	00000092				44.340us	
-0000294734	1		service	0000023				15.205us	
-0000294732	1		service	00000B5				28.230us	
-0000294730	ī			000000B4				4.635us	
		— task: N	D TASK1	(00000007)					
-0000294728	1		owner	00000007				22.300us	
-0000294726	1		service	00000022				19.720us	
-0000294723	0		intr	00000001					
-0000294721	0		intr	00000005				29.000us	
-0000294718	1		intr	0000002				1.130ms	
-0000294716	1		intr	00000005			1	L09.555us	
	· · · · · · · · · · · · · · · · · · ·	— task: T/	ASKSND1	(00000000)					
-0000294714	1		owner	00000000				47.300us	
-0000294712	1		service	00000B3				11.085us	
-0000294710	1		service	00000B2				4.640us	
-0000294708	0		intr	0000003			(	511.325us	
-0000294706	0		intr	00000005				42.790us	
-0000294704	1		service	0000023			1	L36.110us	
		— task: T	ASKRCV1	(00000001)					
-0000294702	0		owner	00000001				27.970us	
									*

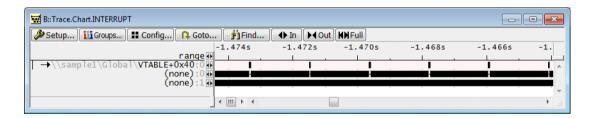
# Exporting all Types of Task Information and all Instructions (OTM)

#### General setup:

NEXUS.BTM ON	; enable the Branch Trace ; Messages
NEXUS.OTM ON	; enable the Ownership Trace ; Messages
Trace.STATistic.InterruptIsFunction ON	; advise TRACE32 to regard the ; time between interrupt entry ; and exit as function

#### **Statistic Analysis of Interrupts**

Trace.Chart.INTERRUPT [/SplitCORE]Interrupt time chart (default), results split<br/>up per coreTrace.Chart.INTERRUPT /CORE <n>Interrupt time chart for specified core



Trace.STATistic.INTERRUPT [/SplitCORE]

Interrupt statistic (default), results split up per core

Trace.STATistic.INTERRUPT /CORE <n>

Interrupt statistic for specified core

E B:: Trace. STATistic. INTERRUPT							- <u>- x</u>
Setup iii Groups Config Deta	ailed [ 🖪 Nestir	ng 🚺 👫 Chart	Profile				
	funcs: 3.	to	otal: 1.08	30s intr:	69.340ms		
range	total	min	max	avr	count	intern% 1%	2%
(none):0	1.011s	-	1.011s	-	-	46.791%	
(none):1	1.080s	-	1.080s	-		50.000%	
→\\sample1\Global\VTABLE+0x40:0	69.124ms	55.600us	730.970us	99.342us	698.	3.208%	
							-
	•		111				5 F
,							

#### Trace.Chart.TASKORINTERRUPT [/SplitCORE]

Trace.Chart.TASKORINTERRUPT /CORE <n>

Time chart for interrupts and tasks (default), results split up per core

Time chart for interrupts and tasks for specified core

🎾 Setup 🚺 Groups	E Config	🔒 Goto	Find	◆ In ► Out	KNFull		
	-1.8	75s -	1.874s	-1.873s	-1.872s	-1.871s	-1.870
r ange 👀							
(unknown):0 💀							
NO_TASK0:00							
TASKRCV1:0							
TASKRCV2:0							· · · · · i i
TASKSTOP:0							
→VTABLE+0x40:0							· · · · · · <b>·</b>
(unknown):1		<del>.</del>		· · · · · · · · · · · · · · · · · ·		· · · <del>·</del> · · · · · · · ·	

Trace.STATistic.TASKORINTERRUPT [/SplitCORE]

Statistic for interrupts and tasks (default), results split up per core

Trace.STATistic.TASKORINTERRUPT /CORE <n>

Statistic for interrupts and tasks for specified core

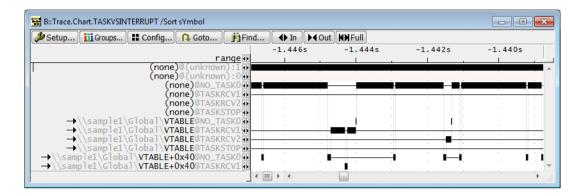
Setup Groups	tasks: 7.		Nesting Nesting 1.97	Chart Pro '8s	ofile	
range	total	min	max	avr	count	ratio% 1%
(unknown):0	5.544ms	5.544ms	5.544ms	5.544ms	0.	- +
NO_TASK0:0	1.854s	203.605us	1.795ms	1.475ms	1257.	93.759%
TASKRCV1:0	30.321ms	234.958us	434.675us	250.584us	121.	1.533%
TASKRCV2:0	9.016ms	128.725us	128.900us	128.797us	70.	0.455% +
TASKSTOP:0	1.131ms	169.810us	207.215us	188.511us	6.	0.057%
→VTABLE+0x40:0	77.415ms	55.181us	307.750us	61.343us	1262.	3.914%
(unknown):1	1.978s	And a second sec	1.978s	1.978s	0.	- 4

Trace.Chart.TASKVSINTERRUPT [/SplitCORE]

Trace.Chart.TASKVSINTERRUPT /CORE <n>

Interrupt time chart, task-related (default), results split up per core

Interrupt time chart task-related, for specified core



Trace.STATistic.TASKVSINTERRUPT [/SplitCORE]

Interrupt statistic, task-related (default), results split up per core

Trace.STATistic.TASKVSINTERRUPT /CORE <n>

Interrupt statistic, task-related, for specified core

B::Trace.STATistic.TASKVSINTERR	UPT /Sort sYmbo	bl					
Setup 🚻 Groups 📲 Cont	fig 🗾 Detaile	ed [ Nesting	Chart	Kan Profile			
	funcs: 12.	to	otal: 1.78	37s			
range	total	min	max	avr	count	intern% 1	%
(none)@(unknown):0	0.000us	-	-	0.000us	0.(1/0)	- 4	
(none)@(unknown):1	0.000us	-	-	0.000us	0.(1/0)	- +	
(none)@NO_TASK0	0.000us	-	-	0.000us	0.(1/0)	0.000%	
(none)@TASKRCV1	0.000us	-	-	0.000us	0.(1/0)	0.000%	
(none)@TASKRCV2	0.000us	-	-	0.000us	0.(1/0)	0.000%	
(none)@TASKSTOP	0.000us	-	-	0.000us	0. (1/0)	0.000%	
1\Global\VTABLE@NO_TASK0	60.841ms	234.720us	775.965us	318.539us	191.	1.702%	
1\Global\ <b>VTABLE</b> @TASKRCV1	0.000us	-	-	0.000us	0.	0.000%	
1\Global\ <b>VTABLE</b> @TASKRCV2	0.000us	-	-	0.000us	0.	0.000%	
1\Global\VTABLE@TASKSTOP	0.000us	-	-	0.000us	0.	0.000%	
bal\VTABLE+0x40@NO_TASK0	53.271ms	55.590us	55.625us	55.607us	958.	1.490%	_
bal\VTABLE+0x40@TASKRCV1	432.895us	71.750us	72.260us	72.149us	6.	0.012%	
					,		-
	4						- 4 / / /
-							

#### Task Switches

An SMP operating system has **one variable per core** that contains the information which task is currently running. One way to export task switch information is to advise the NEXUS hardware module to generate trace information when a write access to one of these variables occurs.

The address of these variables is provided by the TRACE32 functions TASK.CONFIG(magic[<core>]).

```
PRINT TASK.CONFIG(magic[0]) ; print the address of the variable
; that holds the task identifier
; for core 0
PRINT TASK.CONFIG(magic[1]) ; print the address of the variable
; that holds the task identifier
; for core 1
...
PRINT TASK.CONFIG(magic[n]) ; print the address of the variable
; that holds the task identifier
; for core n
```

**Example:** Advise the NEXUS hardware module to generate only trace information on task switches for a dual-core chip.

1. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic[0]) and select the trace action TraceEnable.

🔯 B::Break.Set			
address / expres TASK.CONFIG(m			-
type Program ReadWrite Read	options EXclude NOMARK	<ul> <li>Temporary</li> <li>DISable</li> <li>DISableHIT</li> </ul>	implementation → auto ▼ action TraceEnable ▼
<ul> <li>Write</li> <li>default</li> <li>Ok</li> </ul>	Add	Delete	♥ advanced     Cancel

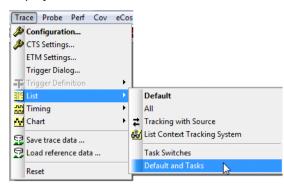
2. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic[1]) and select the trace action TraceEnable.

🙆 B::Break.Set			- • •						
	address / expression TASK.CONFIG(magic[1])								
<ul> <li>type</li> <li>Program</li> <li>ReadWrite</li> <li>Read</li> <li>Write</li> <li>default</li> </ul>	options EXclude NOMARK DATA	Temporary	implementation auto  action TraceEnable advanced						
Ok	Add	Delete	Cancel						

Break.Set TASK.CONFIG(magic[0]) /Write /TraceEnable
Break.Set TASK.CONFIG(magic[1]) /Write /TraceEnable

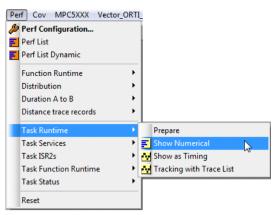
3. Start and stop the program execution to fill the trace buffer

#### 4. Display the result.



📕 B::Trace.List l	ist.TASK DEFault		
Setup 🚺	🕽 Goto 🦳 👬 Find 🛛 🚻 Chart 🖉 🎦 Profile 🕽	MIPS Amore Less	
record	run address cycle data	symbol	ti.back
-000000037	TASK = NO_TASK1 1   D:40000A45 wr-byte TASK = TASKSND1	07 \\sample1\Global\OsOrtiRunning_+0x1	145.660us
-000000034		00 \\sample1\Global\OsOrtiRunning_+0x1	1.307ms
-000000032		01 \\sample1\Global\OsOrtiRunning_	2.704ms
-000000030		07 \\sample1\Global\OsOrtiRunning_+0x1	209.975us
-000000028		06 \\sample1\Global\OsOrtiRunning_	258.440us
-000000025	1   D:40000A45 wr-byte	02 \\sample1\Global\OsOrtiRunning_+0x1	2.688ms
-000000023		07 \\sample1\Global\OsOrtiRunning_+0x1	85.325us
-0000000020	1   D:40000A45 wr-byte	04 \\sample1\Global\OsOrtiRunning_+0x1	1.365ms 🔫

The following commands perform a statistical analysis of the task switches:



Setup iii Grou	TASK Jps 🔡 Confid	<b>E</b> Detailed	Nesting	MChart	Profile					
· · ·	tasks: 10.		tal: 4.93		<b>_</b> )					$\top$
range	total	min	max	avr	count	ratio%	1%	2%	5%	
(unknown):0	0.000us	0.000us	-	0.000us	0.	0.000%				
TASKRCV1:0	81.231ms	257.245us	591.560us	273.504us	297.	1.645%				
NO_TASK0:0	4.830s	117.285us	236.480ms	10.126ms	477.	97.817%				
TASKRCV2:0	23.205ms	139.700us	139.875us	139.788us	166.	0.469%	+			
TASKSTOP:0	3.347ms	214.460us	263.740us	239.099us	14.	0.067%	+			
(unknown):1	533.680us	533.680us	533.680us	533.680us	0.	0.010%	+			
TASKSND1:1	59.367ms	209.940us	220.430us	210.523us	282.	1.202%	-			
NO_TASK1:1	4.780s	1.184ms	237.130ms	5.650ms	846.	96.801%				-
TASKSND2:1	54.154ms	85.315us	266.595us	192.036us	282.	1.096%	-			
TASKCNT:1	43.880ms	144.995us	457.855us	155.604us	282.	0.888%	+			
	4			III	'				+	

Trace.STATistic.TASK [/SplitCORE]

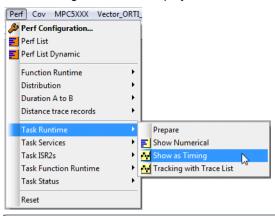
Task runtime statistic, result per core

B::Trace.STAT						To al					×
Setup				Nesting		Profile					
	t	asks: 10.	to	tal: 4.93	88s						
ra	nge  t	otal	min	max	avr	count	ratio%	1%	2%	5%	
(unknown	):0	0.000us	0.000us	-	0.000us	0.	0.000%				
(unknown	):1	533.680us	533.680us	533.680us	533.680us	0.	0.005%	+			
TASKR	CV1	81.231ms	257.245us	591.560us	273.504us	297.	0.822%	+			
TASKS	ND1	59.367ms	209.940us	220.430us	210.523us	282.	0.601%	+			
NO_TA	SK0	4.830s	117.285us	236.480ms	10.126ms	477.	48.908%				-
NO_TA	SK1	4.780s	1.184ms	237.130ms	5.650ms	846.	48.400%				-
TASKS	ND2	54.154ms	85.315us	266.595us	192.036us	282.	0.548%	+			
TASKR	CV2	23.205ms	139.700us	139.875us	139.788us	166.	0.234%	+			
TASK	CNT	43.880ms	144.995us	457.855us	155.604us	282.	0.444%	+			
TASKS	TOP	3.347ms	214.460us	263.740us	239.099us	14.	0.033%	+			-
		(			m					•	E at

Trace.STATistic.TASK /MergeCORE

Task runtime statistic, results of all cores merged

The following commands display a time-chart of the task run-times:



Setup 🚻 Groups 📲 Config	📭 Go	to ]	<b>F</b>	Find	.][	) In		Out	HM	Full	]										
s	-5.	955s		-5.	950s		- 5	. 94	5 s		-5.	940	s		-5.	935	s		- 5	. 93	30s
r ange 💀								1 .			. 1									1	
(unknown): 🛛 💀																					
TASKRCV1:0																					
NO_TASK0:0																					
TASKRCV2 : 0															Ĩ						
TASKSTOP : 0																					
(unknown):10														5.18							
TASKSND1:10				10.01				1													
NO_TASK1:1				i di				1		1			Ĩ.	1							
TASKSND2 : 1			100	1000			225					100						See		1	
TASKCNT:10			0.2	1.00															1		

Trace.Chart.TASK [/SplitCORE]

Task runtime time chart, result per core

B::Trace.Chart.TASK /MergeCORE												
🎾 Setup 🚺 Groups 🕻 🚼 Config 📭 Goto 🎁 Find 🚺 In 🕨 Out 🕅 KN Full												
	-5.948s	-5.947s	-5.946s	-5.945s	-5.944s	-5.943s						
r ange 💀												
(unknown):0 💀												
(unknown):1 💀												
TASKRCV1 🚸												
TASKSND1 💀												
NO_TASK0												
NO_TASK1 •												
TASKSND2		· · · <u>· · ·</u> · · ·										
TASKRCV2		· · · · · · · · · · · · · · ·										
TASKCNT		<mark></mark>	· · · · · · · · · · ·									
TASKSTOP			· · · · · · · · · · · · · · ·									
	4 <u>m</u> + 4					4						

Trace.Chart.TASK /MergeCORE

Task runtime time chart, results of all cores merged

The time spent in OSEK service routines can be evaluated.

OSEK writes information on the entries and exits to OSEK service routines to a defined variable per core. One way to export information on OSEK service routines is to advise the NEXUS hardware module to generate trace information when a write access to one of these variables occurs.

The address of these variables is provided by the TRACE32 functions **TASK.CONFIG(magic\_service[**<core>]).

```
PRINT TASK.CONFIG(magic_service[0]) ; print the address of the
; variable that holds the
; service information for core 0
PRINT TASK.CONFIG(magic_service[1]) ; print the address of the
; variable that holds the
; service information for core 1
...
PRINT TASK.CONFIG(magic_service[n]) ; print the address of the
; variable that holds the
; service information for core n
```

**Example:** Advise the NEXUS hardware module to generate only trace information for entries and exits to OSEK service routines for a dual-core chip.

1. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic\_service[0]) and select the trace action TraceEnable.

🔮 B::Break.Set								
address / expression TASK.CONFIG(magic_service[0])								
<ul> <li>─ type</li> <li>─ Program</li> <li>○ ReadWrite</li> <li>○ Read</li> </ul>	options	Temporary DISable	implementation auto action TraceEnable					
Write     default     Ok	Add	Delete	★ advanced     Cancel					

2. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic\_service[1]) and select the trace action TraceEnable.

🔯 B::Break.Set			
address / expres			
TASK.CONFIG(m	agic_service[1])		▼ 🛓 🗋 HLL
type	– options –––––		- implementation
Program	EXclude	Temporary	auto 👻
ReadWrite	NOMARK	DISable	- action
🔘 Read		DISableHIT	TraceEnable 💌
Write	DATA		
Ø default		•	
Ok	Add	Delete	Cancel

Break.Set TASK.CONFIG(magic\_service[0]) /Write /TraceEnable
Break.Set TASK.CONFIG(magic\_service[1]) /Write /TraceEnable

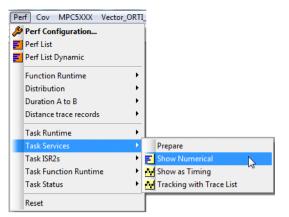
3. Start and stop the program execution to fill the trace buffer

### 4. Display the result.

Tra	ice Perf Cov MPC5XX	Х	٧	
	Configuration CTS Settings NEXUS Settings			
	List	•		Default
流	Timing	►		All
24	Chart	►	₽	Tracking with Source
ବ	Save trace data		67	List Context Tracking System
_	Load reference data			Task Switches and Services
	Reset			Default and Tasks

B::Trace.List L	ist.TASK DEFault	- • •
Setup 👔	🕽 Goto 🎽 Find 🛛 💾 Chart 🛛 🌌 Profile 🖉 🏧 MIPS 🛛 🜩 More 🖉 🗶 Less	
record		ti.back 🔄
-0000000128	SERVICE = OSServiceId_PostTaskHook entry 0   D:40000A3E wr-byte B5 \\sample1\Global\OsOrtiRunningServiceId_ SERVICE = OSServiceId_PostTaskHook exit 0   D:40000A3E wr-byte B4 \\sample1\Global\OsOrtiRunningServiceId_	17.015us
	SERVICE = OSServiceId_StartScheduleTableAsync exit	*
-0000000126	SERVICE = OSServiceId TerminateTask exit	176.465us
-0000000125	SERVICE = OSServiceId_TerminateTask entry	22.945us
-0000000124	1   D:40000A3F wr-byte 23sample1\Global\OsOrtiRunningServiceId_+0x1 SERVICE = OSServiceId_PostTaskHook entry	8.250us
-0000000122	SERVICE = OSServiceId_PostTaskHook exit	27.070us
-0000000121	SERVICE = OSServiceId_TerminateTask exit	5.545us
-0000000120	1   D:40000A3F wr-byte 22sample1\Global\OsOrtiRunningServiceId_+0x1	40.605us -

The following two commands perform a statistical analysis of the OSEK service routines:



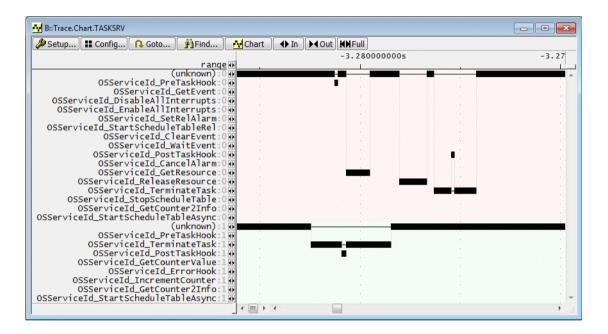
(unknown) represents the time in which the processor/core is not in an OSEK service routine

	10										
🖲 Setup 🚺 Groups 📰 Config 📰 Detailed	Nesting C	hart 📃 📕 Prof									
	srvs: 26.	to	tal: 4.07	'1s							
range total min max avr count ratio% 1% 2											
(unknown):0	4.005s	111111111111111111111111111111111111111	4.005s	4.005s	0.	98.377%					
OSServiceId_PreTaskHook:0	1.698ms	3.350us	4.645us	4.191us	405.	0.041% +					
OSServiceId_GetEvent:0	3.137ms	11.340us	12.505us	12.447us	252.	0.077% +					
OSServiceId_DisableAllInterrupts:0	124.240us	5.025us	5.415us	5.177us	24.	0.003% +					
OSServiceId_EnableAllInterrupts:0	110.575us	4.380us	5.030us	4.607us	24.	0.002% +					
OSServiceId_SetRelAlarm:0	9.736ms	32.345us	127.740us	36.880us	264.	0.239%					
OSServiceId_StartScheduleTableRel:0	2.795ms	232.880us	232.920us	232.898us	12.	0.068%					
OSServiceId_ClearEvent:0	2.198ms	8.630us	9.410us	8.723us	252.	0.053%					
OSServiceId_WaitEvent:0	12.579ms	49.615us	54.780us	49.919us	252.	0.290% +					
OSServiceId_PostTaskHook:0	1.255ms	2.960us	3.480us	3.098us	405.	0.030% +					
OSServiceId_CancelAlarm:0	6.402ms	26.545us	26.685us	26.674us	240.	0.157% +					
OSServiceId_GetResource:0	8.688ms	22.035us	24.105us	22.804us	381.	0.213%					
OSServiceId_ReleaseResource:0	10.158ms	26.160us	27.590us	26.661us	381.	0.249%					
OSServiceId_TerminateTask:0	6.433ms	41.755us	45.245us	42.043us	153.	0.146% +					
OSServiceId_StopScheduleTable:0	1.885ms	132.620us	181.620us	157.113us	12.	0.046%					
OSServiceId_GetCounter2Info:0	54.135us	18.045us	18.045us	18.045us	3.	0.001%					
OSServiceId_StartScheduleTableAsync:0	48.330us	16.110us	16.110us	16.110us	3.	0.001% +					
(unknown):1	3.992s		3.992s	3.992s	0.	98.070%					
OSServiceId_PreTaskHook:1	3.345ms	4.635us	5.030us	4.646us	720.	0.082% +					
OSServiceId_TerminateTask:1	55.611ms	74.745us	81.470us	77.238us	720.	1.282%					
OSServiceId_PostTaskHook:1	3.415ms	4.635us	4.900us	4.743us	720.	0.083% +					
OSServiceId_GetCounterValue:1	6.724ms	47.680us	47.700us	47.691us	141.	0.150% +					
OSServiceId_ErrorHook:1	617.895us	4.380us	4.385us	4.382us	141.	0.015% +					
OSServiceId_IncrementCounter:1	11.648ms	43.820us	127.745us	48.531us	240.	0.286% +					
OSServiceId_GetCounter2Info:1	186.240us	30.800us	31.320us	31.040us	6.	0.004% +					
OSServiceId_StartScheduleTableAsync:1	1.042ms	173.605us	173.745us	173.695us	6.	0.025% +					

Trace.STATistic.TASKSRV [/SplitCORE]

Statistic on service routines, result per core

Perf Configuration			
Perf List			
Perf List Dynamic			
Function Runtime	•		
Distribution	•		
Duration A to B	•		
Distance trace records	+		
Task Runtime	•		
Task Services	Þ	Prepare	
Task ISR2s	÷	Show Numerical	
Task Function Runtime	•	🙀 Show as Timing	N
Task Status	•	🙀 Tracking with Trace List	5



Trace.Chart.TASKSRV [/SplitCORE]

Time chart on service routines, result per core

The time spent in OSEK interrupt service routines can be evaluated.

OSEK writes information on the start of an interrupt service routine to a defined variable per core as well as the information NO\_ISR. One way to export information on OSEK interrupt service routines is to advise the NEXUS hardware module to generate trace information when a write access to these variables occurs.

The address of these variables is provided by the TRACE32 functions **TASK.CONFIG(magic\_isr2[**<core>]).

```
PRINT TASK.CONFIG(magic_isr2[0]) ; print the address of the variable
; that holds the interrupt service
; information for core 0
PRINT TASK.CONFIG(magic_isr2[1]) ; print the address of the variable
; that holds the interrupt service
; information for core 1
...
PRINT TASK.CONFIG(magic_isr2[n]) ; print the address of the variable
; that holds the interrupt service
; information for core n
```

**Example:** Advise the NEXUS hardware module to generate only trace information on the start of an interrupt service routine as well as on the information NO\_ISR for a dual-core chip.

1. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic\_isr2[0]) and select the trace action TraceEnable.

🔯 B::Break.Set			- • ×						
address / expression TASK.CONFIG(magic_isr2[0])									
vpe Program ReadWrite Read Write	options	<ul> <li>Temporary</li> <li>DISable</li> <li>DISableHIT</li> </ul>	implementation auto ▼ action TraceEnable ▼						
O default	Add	▼ Delete	♥ advanced     Cancel						

2. Set a Write breakpoint to the address indicated by TASK.CONFIG(magic\_isr2[1]) and select the trace action TraceEnable.

🔯 B::Break.Set			- • •					
address / expres			▼ 3 □ HLL					
TASK.CONFIG(m	TASK.CONFIG(magic_isr2[1])							
- type	options		- implementation					
Program	EXclude	Temporary	auto 🔻					
ReadWrite	NOMARK	DISable	- action					
🔘 Read		DISableHIT	TraceEnable 🔻					
Write	DATA							
© default		-						
Ok	Add	Delete	Cancel					

Break.Set TASK.CONFIG(magic\_isr2[0]) /Write /TraceEnable
Break.Set TASK.CONFIG(magic\_isr2[1]) /Write /TraceEnable

3. Start and stop the program execution to fill the trace buffer

### 4. Display the result.

Tra	ice Perf Cov MPC5XX	X	٧	
	Configuration CTS Settings NEXUS Settings			
	List	Þ		Default
52	Timing	Þ		All
$\sim$	Chart	Þ	₽	Tracking with Source
e.	Save trace data		6	List Context Tracking System
9	Load reference data			Task Switches and Services
	Reset			Default and Tasks

B::Trace.List List.TASK DEFault			83
🖉 Setup 🔁 Goto 🎁 Find 🥂 Chart 🖉 🎽 Prof	file   📕 MIPS   🜩 More   🗶 Less		
record run address cycle data	symbol	ti.back	
ISR2 = SECOND_TIMER -0000038840 1 D:40000A36 wr-word ISR2 = NO ISR	0002 \\sample1\Global\OSISRId_+0x2	1.342ms	-
-0000038838 1   D:40000A36 wr-word ISR2 = SYSTEM_TIMER	0005 \\sample1\Global\OSISRId_+0x2	108.155us	T A
-0000038836 0   D:40000A34 wr-word ISR2 = N0_ISR	0001 \\sample1\Global\OSISRId_	1.904ms	
-0000038834 0   D:40000A34 wr-word ISR2 = SECOND_TIMER	0005 \\sample1\Global\OSISRId_	29.520us	
-0000038832 1   D:40000A36 wr-word ISR2 = NO_ISR	0002 \\sample1\Global\OSISRId_+0x2	1.342ms	
-0000038830 1   D:40000A36 wr-word ISR2 = OS_TIMER	0005 \\sample1\Global\OSISRId_+0x2	110.215us	
-0000038829 0   D:40000A34 wr-word	0003 \\sample1\Global\OSISRId_	658.865us	<b>▼</b>

The following commands perform a statistical analysis of the OSEK interrupt service routines:

erf Cov MPC5XXX Vecto Perf Configuration		
Perf List		
Perf List Dynamic		
Function Runtime	+	
Distribution	•	
Duration A to B	•	
Distance trace records	+	
Task Runtime	+	
Task Services	•	
Task ISR2s	Þ	Prepare
Task Function Runtime	÷	🛃 Show Numerical 💦 📐
Task Status	•	🙀 Show as Timing
Reset		🙀 Tracking with Trace List

E B:: Trace.STATistic.TASKINTR								3	
🌽 Setup 🗓 Groups 🚼 Config 🗾 Detailed  👔 Nesting 🔂 Chart 🛛 📕 Profile									
	intrs: 8.	to	tal: 12.61	.3s					
range	total	min	max	avr	count	ratio% 1%	2%		
(unknown):0	0.000us	0.000us	-	0.000us	1.	0.000%			
SYSTEM_TIMER:0	193.169ms	29.510us	46.025us	29.609us	6524.	1.531%			
NO_ISR:0	12.370s	207.380us	1.904ms	1.606ms	7703.	98.072%			
OS_TIMER:0	49.992ms	40.725us	43.315us	42.366us	1180.	0.396% 🗲			
(unknown):1	42.774ms	42.774ms	42.774ms	42.774ms	1.	0.339% +			
SECOND_TEST:1	12.277ms	77.720us	180.995us	113.674us	108.	0.097%			
NO_ISR:1	12.327s	58.385us	236.699ms	5.525ms	2231.	97.732%			
SECOND_TIMER:1	230.917ms	107.875us	110.230us	108.718us	2124.	1.830%		-	
	•						F		

Trace.STATistic.TASKINTR [/SplitCORE] Statistic on interrupt service routines, result per core

Perf Cov MPC5XXX Vector	r_ORTI_2
🔑 Perf Configuration	
Perf List	
Perf List Dynamic	
Function Runtime	•
Distribution	•
Duration A to B	•
Distance trace records	+
Task Runtime	•
Task Services	•
Task ISR2s	Prepare
Task Function Runtime	Show Numerical
Task Status	Show as Timing
Reset	Tracking with Trace List

₩ B::Trace.Chart.TASKINTR										×
Setup 🚼 Config 📭 Goto	Find	Chart	🕀 In 🕞	Out	ull					
	-8.377500000s				-8.37700000s					
range 🕖								1		
NO_ISR:0										- A
SYSTEM_TIMER:00										
OS_TIMER:0										
(unknown):0										
NO_ISR:1										
SECOND_TIMER:1										
SECOND_TEST:1										
(unknown):1										-
l i iii	• •									•i

Trace.Chart.TASKINTR [/SplitCORE]

Time chart on interrupt service routines, result per core

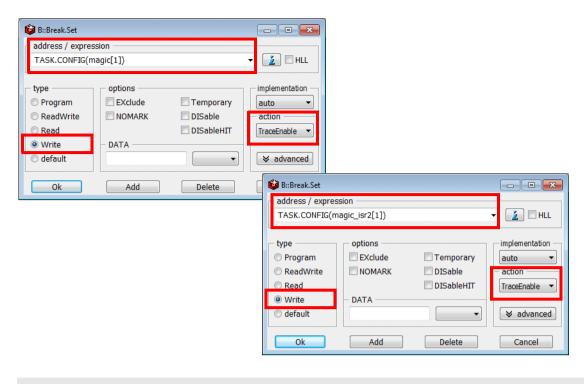
OSEK interrupt service routines that occur in multiple tasks can be displayed per task, if the following information is available:

- Task switch information per core
- ISR2 start and NO\_ISR information per core

#### Example:

- 1. Advise the NEXUS hardware module to generate the following trace information for a dual-core chip:
  - task switches per core
  - start of an interrupt service routine as well as on the information NO\_ISR per core.

🚳 B::Break.Set								
address / expres TASK.CONFIG(m			•	ź 🗆 HLL				
type Program ReadWrite Read Write default	options EXclude NOMARK	Temporary DISable DISableHIT	aut – act Trac	lementation to v ton teEnable v advanced				
Ok	Add	Delete		B::Break.Set address / ex TASK.CONFI	press			X
				<ul> <li>type</li> <li>Program</li> <li>ReadWrite</li> <li>Read</li> <li>Write</li> <li>default</li> </ul>	•	options EXclude NOMARK DATA	Temporary DISable DISableHIT	implementation
				O default	]	Add	Delete	✓ advanced Cancel



Break.Set TASK.CONFIG(magic[0]) /Write /TraceEnable
Break.Set TASK.CONFIG(magic\_isr2[0]) /Write /TraceEnable
Break.Set TASK.CONFIG(magic[1]) /Write /TraceEnable
Break.Set TASK.CONFIG(magic\_isr2[1]) /Write /TraceEnable

2. Start and stop the program execution to fill the trace buffer.

## 3. Display the result.

Tra	ce Perf Cov MPC5XX	<	٧	
Ju Ju	Configuration CTS Settings NEXUS Settings			
	List			Default
	Timing	۲		All
2	Chart	۲	₽	Tracking with Source
Ģ	Save trace data		6	List Context Tracking System
9	Load reference data			Task Switches and Services
	Reset			Default and Tasks

🖉 Setup 🛛 🖪	Goto	Find	Chart	Profile 🛛 🔛 N	1IPS	More	Less		
record	run ac	ldress	cycle	data		symbol		ti.back	
1		$R2 = NO_{ISR}$							
0000044868		D:40000A36		00	005	\\samplel\0	lobal\OSISRId_+0x2	108.140us	
0000044866		SR2 = SYSTEM D:40000A34		00	001	\\sample1\0	lobal\OSISRId_	1.904ms	
		$R2 = NO_{ISR}$							
0000044864		D:40000A34		00	05	\\sample1\0	lobal\OSISRId_	29.515us	
		R2 = SECOND						1 242	
0000044862		D:40000A36 R2 = NO_ISR		00	02	\\samplel\@	lobal\OSISRId_+0x2	1.342ms	
0000044860		D:40000A36		00	05	\\sample1\0	lobal\OSISRId_+0x2	110.070us	
0000044859		<pre>SR2 = OS_TIMI D:40000A34</pre>		00	003	\\sample1\0	lobal\OSISRId_	612.740us	
	IS	$R2 = NO_{ISR}$				··· · ·	-		
0000044857	0	D:40000A34	wr-word	00	05	\\sample1\0	lobal\OSISRId_	43.300us	

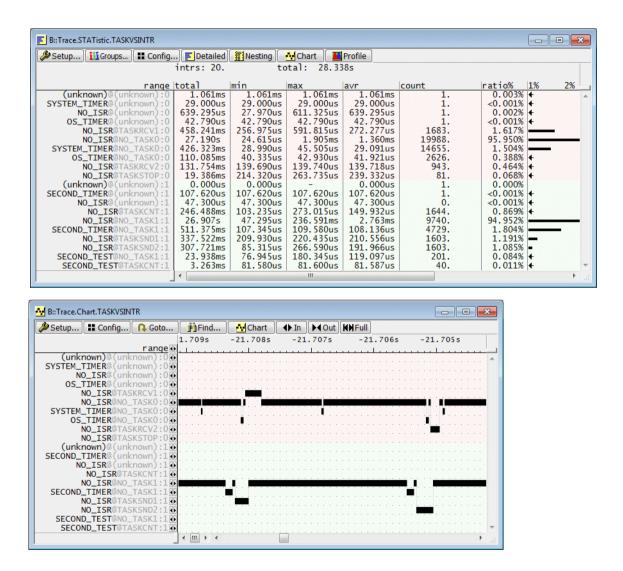
The following commands allow to perform a statistical analysis of the OSEK interrupt service routines related to the active tasks.

Trace.STATistic.TASKVSINTR [/SplitCORE]

Trace.Chart.TASKVSINTR [/SplitCORE]

Task-related statistic on interrupt service routines, result per core

Time-chart for task related interrupt service routines, result per core



intr information that was generated before the first task information is assigned to the @(unknown) task.

General setup:

Break.Set TASK.CONFIG(magic[0]) /Write /TraceData
Break.Set TASK.CONFIG(magic[1]) /Write /TraceData
...
; advise TRACE32 to regard the time between interrupt entry
; and exit as function
Trace.STATistic.InterruptIsFunction ON

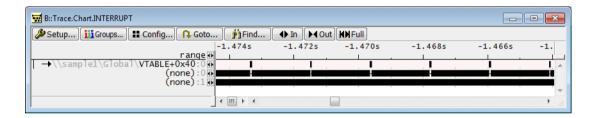
### **Statistic Analysis of Interrupts**

Trace.Chart.INTERRUPT [/SplitCORE]

Interrupt time chart (default), results split up per core

Trace.Chart.INTERRUPT /CORE <n>

Interrupt time chart for specified core

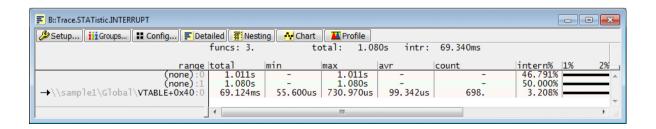


Trace.STATistic.INTERRUPT [/SplitCORE]

Interrupt statistic (default), results split up per core

Trace.STATistic.INTERRUPT /CORE <n>

Interrupt statistic for specified core



### Trace.Chart.TASKORINTERRUPT [/SplitCORE]

Trace.Chart.TASKORINTERRUPT /CORE <n>

Time chart for interrupts and tasks (default), results split up per core

Time chart for interrupts and tasks for specified core

🎾 Setup 🚺 Groups	E Config	🔒 Goto	Find	◆ In ► Out	KNFull		
	-1.8	75s -	1.874s	-1.873s	-1.872s	-1.871s	-1.870
r ange 👀							
(unknown):0 💀							
NO_TASK0:00							
TASKRCV1:0							
TASKRCV2:0							· · · · · i i
TASKSTOP:0							
→VTABLE+0x40:0							· · · · · · <b>·</b>
(unknown):1		<del>.</del>		· · · · · · · · · · · · · · · · · ·		· · · <del>· </del> · · · · · · · ·	

Trace.STATistic.TASKORINTERRUPT [/SplitCORE]

Statistic for interrupts and tasks (default), results split up per core

Trace.STATistic.TASKORINTERRUPT /CORE <n>

Statistic for interrupts and tasks for specified core

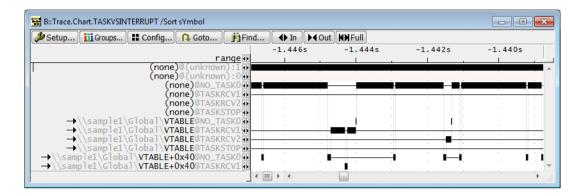
Setup 🎦 Groups	tasks: 7.		Nesting Nesting 1.97	Chart Pro '8s	ofile		
range	total	min	max	avr	count	ratio% 1	%
(unknown):0	5.544ms	5.544ms	5.544ms	5.544ms	0.	- 4	F .
NO_TASK0:0	1.854s	203.605us	1.795ms	1.475ms	1257.	93.759%	
TASKRCV1:0	30.321ms	234.958us	434.675us	250.584us	121.	1.533%	
TASKRCV2:0	9.016ms	128.725us	128.900us	128.797us	70.	0.455%	+
TASKSTOP:0	1.131ms	169.810us	207.215us	188.511us	6.	0.057%	F
$\rightarrow$ VTABLE+0x40:0	77.415ms	55.181us	307.750us	61.343us	1262.	3.914%	
(unknown):1	1.978s		1.978s	1.978s	0.		Longer

Trace.Chart.TASKVSINTERRUPT [/SplitCORE]

Trace.Chart.TASKVSINTERRUPT /CORE <n>

Interrupt time chart, task-related (default), results split up per core

Interrupt time chart task-related, for specified core



Trace.STATistic.TASKVSINTERRUPT [/SplitCORE]

Interrupt statistic, task-related (default), results split up per core

Trace.STATistic.TASKVSINTERRUPT /CORE <n>

Interrupt statistic, task-related, for specified core

B::Trace.STATistic.TASKVSINTERR	UPT /Sort sYmbo	bl					
Setup 🚻 Groups 🎛 Conf	fig 🗾 Detaile	ed [ Nesting	Chart	🚺 Profile			
	funcs: 12.	to	tal: 1.78	37s			
range	total	min	max	avr	count	intern%	1%
(none)@(unknown):0	0.000us	-	-	0.000us	0.(1/0)	-	+
(none)@(unknown):1	0.000us	-	-	0.000us	0.(1/0)	-	+
(none)@NO_TASK0	0.000us	-	-	0.000us	0. (1/0)	0.000%	
(none)@TASKRCV1	0.000us	-	-	0.000us	0.(1/0)	0.000%	
(none)@TASKRCV2	0.000us	-	-	0.000us	0. (1/0)	0.000%	
(none)@TASKSTOP	0.000us	-	-	0.000us	0. (1/0)	0.000%	
1\Global\ <b>VTABLE</b> @NO_TASK0	60.841ms	234.720us	775.965us	318.539us	191.	1.702%	
1\Global\ <b>VTABLE</b> @TASKRCV1	0.000us	-	-	0.000us	0.	0.000%	
1\Global\ <b>VTABLE</b> @TASKRCV2	0.000us	-	-	0.000us	0.	0.000%	
1\Global\ <b>VTABLE</b> @TASKSTOP	0.000us	-	-	0.000us	0.	0.000%	
bal\VTABLE+0x40@NO_TASK0	53.271ms	55.590us	55.625us	55.607us	958.	1.490%	
bal\ <b>VTABLE+0x40</b> @TASKRCV1	432.895us	71.750us	72.260us	72.149us	6.	0.012%	+
	l						~
	•						ь. <b>4</b>
-							

The TRACE32 Instruction Set Simulator can be used for a belated OS-aware trace evaluation. To set up the TRACE32 Instruction Set Simulator for belated OS-aware trace evaluation proceed as follows:

1. Save the trace information for the belated evaluation to a file.

```
Trace.SAVE belated__orti.ad
```

2. Set up the TRACE32 Instruction Set Simulator for a belated OS-aware trace evaluation (here OSEK on a MPC5553):

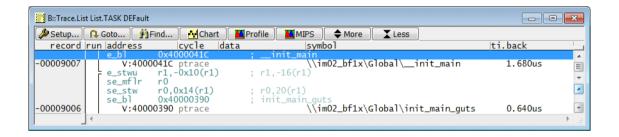
```
SYStem.CPU MPC5553
                                            ; select the target CPU
SYStem.Up
                                            ; establish the
                                            ; communication between
                                            ; TRACE32 and the TRACE32
                                            ; Instruction Set
                                            ; Simulator
Trace.LOAD belated orti.ad
                                            : load the trace file
Data.Load.ELF my_app.out /NoCODE /GHS
                                            ; load the symbol and
                                            ; debug information
                                            ; load the ORTI file
TASK.ORTI my orti.ort
Trace.List List.TASK DEFault
                                            ; display the trace
                                            ; listing
```

All commands for the function run-time analysis introduced in this chapter use the **contents of the trace buffer** as base for their analysis.

If you use Branch History Tracing it is recommended to enable Program Trace Correlation Messages for bl *<func>* and e\_bl *<func>* instructions (saves return address in link register, then jumps to *<func>*) (IEEE-ISTO 5001-2008 and subsequent standards only).

As a result function entries are timestamped in the trace.

NEXUS.HTM ON	
NEXUS.PTCM BL_HTM ON	; generate Program Trace ; Correlation message when a ; "Branch and Link" instruction ; executes



# Software under Analysis (no OS or OS)

For the use of the function run-time analysis it is helpful to differentiate between two types of application software:

- 1. Software without operating system (abbreviation: **no OS**)
- 2. Software with an operating system (abbreviation: **OS**)

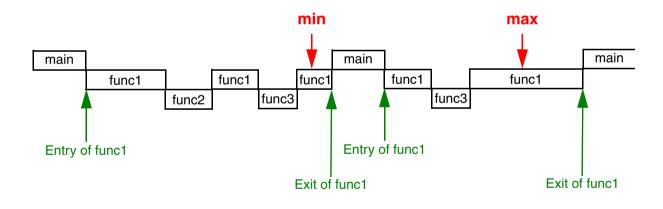
# Flat vs. Nesting Analysis

TRACE32 provides two methods to analyze function run-times:

- Flat analysis
- Nesting analysis

The flat function run-time analysis bases on the symbolic instruction addresses of the trace entries. The time spent by an instruction is assigned to the corresponding function/symbol region.

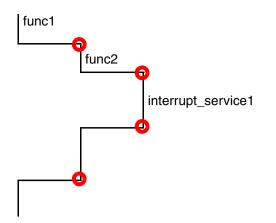
B::Trace.List	/Track									×
	run addre	255	Chart cycle	Profile dat a	Symbol	And the More X	Less	ti.b	ack	
156 -0188026394 -0188026391	l ldr	(*intptr) r1,[u 0:08007FF 0:080000 r1,#( r1,[u	0] 8 rd-long 2 ptrace 0x1	9 000002	D \\demo_i \\demo_	r4\Global\ r4\demo\fu	_end+0x632 nc1+0x2	28 <0 0	0.005us 0.185us	• •
B::Trace.Char	t.sYmbol /Trac	k								8
\\demo_r4	i Groups addre 4\demo\fun 4\demo\func r4\demo\ma \demo\func	3050 ss () c2 () c1 () r) () in () 2a ()	Q Goto	j Find	-6.730504			. 73050400 I	0s	▲ ● ●
B::Trace.STAT										8
🌽 Setup 🚺	Groups		Goto s: 29.			730s	Profile	Solution 2018 States State		
\\demo_i	addr (oth _r4\demo\ <b>m</b> r4\demo\ <b>fu</b> r4\demo\ <mark>fu</mark>	ain 17 nc2 7	1 mi .000us .953ms .483us .690us	0.000us 0.000us 0.185us 0.185us	max 2.955u 6.095u 1.475u	is 7.48	3ms 3us	0. 1. 1. 7.	ratio% 0.000% 0.266% <0.001% <0.001%	€ -



min	shortest time continuously in the address range of the function/symbol region
max	longest time continuously in the address range of the function/symbol region

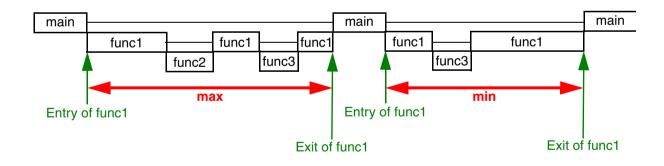
The function nesting analysis analyses only high-level language functions.

B::Trace.List /Track	
595 func2(); b1 0x10E4 ; func2	
void func2() 160 {	*
<pre>stmdb r13!,{r3,r14}</pre>	·▼ El 4
🖼 B::Trace.Chart.Func /Track	- • •
	)ms 2.
🖉 Setup ) 🔢 Goups ) 😫 Config 🔃 Goto 📻 List all ) 🧱 Nesting 🔛 Chart ) 📀 Init	
	1% 2%
(root)         12.238s         -         12.238s         12.238s         -         0.018%           \\armla\a_li_aif\main         12.236s         -         12.236s         12.236s         1.0/1)         0.837%           \\armla\a_li_aif\func1         33.695us         33.695us         33.695us         1.         <0.001%	+ + + +
< III	H. (▲2000000000000000000000000000000000000



In order to display a nesting function run-time analysis TRACE32 analyzes the structure of the program execution by processing the trace information. The focus is put on the transition between functions (see picture above). The following events are of interest:

- 1. Function entries
- 2. Function exits
- 3. Entries to interrupt service routines
- 4. Exits of interrupt service routines
- 5. Entries to TRAP handlers (not implemented yet)
- 6. Exits of TRAP handlers (not implemented yet)



min	shortest time within the function including all subfunctions and traps
max	longest time within the function including all subfunctions and traps

The nesting analysis provides more details on the structure and the timing of the program run, but it is much more sensitive then the flat analysis. Missing or tricky function exits for example result in a worthless nesting analysis.

This chapter applies for single-core TRACE32 instances.

## **Flat Analysis**

It is recommended to reduce the trace information generated by NEXUS to the required minimum.

- To avoid an overload of the NEXUS port.
- To make best use of the available trace memory.
- To get a more accurate timestamp.

## **Optimum NEXUS Configuration (No OS)**

Flat function run-time analysis does **not** require any **data information** if no OS is used. That's why it is recommended to switch Data Trace Messaging off.

NEXUS.DTM OFF

Your function time chart **can** include task information if you advise NEXUS to export the instruction flow and task switches. For details refer to the chapter "**OS-Aware Tracing (ORTI File)**", page 193.

Trace.Chart.sYmbol /TASK "TASK3"

B::Trace.Chart.sYmbol /TASK "TASK3"			
🎾 Setup 🚺 👖 Gonfig 🕞 🗛 Goto 🚺 🖓 Find	i In I Out I Full	]	
80m s	-12.460ms	-12.440ms	-12.420ms
address 💀			
(other)@(other)			
\\imO2_bf1x\os\OSSetPIDO			
\\im02_bf1x\ossch\OSTaskInternalDispatch			
\\imO2_bf1x\imO2\PreTaskHook			
<pre>\\im02_bf1x\ostsk\OS_GetTaskID</pre>			
\\imO2_bf1x\imO2\FuncTASK3			
<pre>\\im02_bf1x\osisr\0S_Suspend0SInterrupts</pre>			
\\1m02_bt1x\1m02\InvokeSciInt			
\\imO2_bf1x\Global\ <b>VTABLE</b>			
4 III + 4			•

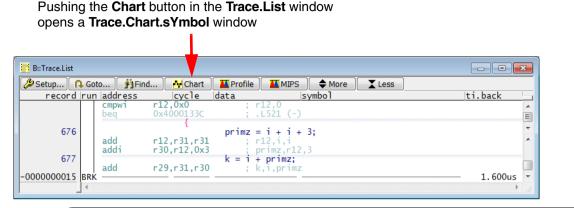
Optimum Configuration 1 (if OSEK generated OTMs):

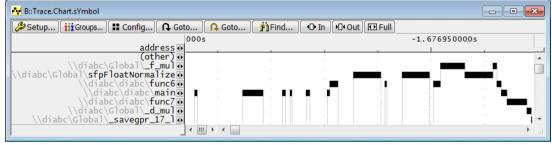
NEXUS.OTM ON

**Optimum Configuration 2** (if OSEK does not support OTMs, NEXUS class 3 only):

Break.Set TASK.CONFIG(magic) /Write /TraceData

TRACE32 PowerView provides a time chart which shows when the program counter was in which function/symbol range.





Trace.Chart.sYmbol

Trace.Chart.sYmbol [/MergeTASK]

Display function time chart (no OS)

Display function time chart (OS but task information is not of interest)

Setup	iii Groups	Config	Goto	Goto	Find	(In In	•D•Out	🖸 Full			
	FIFOFU		m	s -930	.400ms	-930	.200ms	-930.00	Oms	-929.80	Oms
			ddress 🚯		· · · ·			· · · · · · · · · · · · · · · · · · ·			
		abc\diabc				ļ					
		iabc\diabc							- U		
		abc\diabc\									
		abc\diabc\									
		abc\diabc\t									1 1 1
diaba) (		:\Global\_@ DoubleNori								1.1.	
\ulabelo		SDOUDTENON Clobal								1011	
		:\Global\_									
		c\Global\								·	1.11.1
		c\Global\								1 (1	1111
		abc\diabc\i									
		iabc\diabc									
		abc\diabc									
		abc\diabc									
		\Global\_		1							
		\Global\_		1						· · ·	
			OPPED) 🚯								
		(ETC)	OFULL) 🐻								

**(STOPPED):** If the trace recording contains time periods in which the program execution was stopped, these time periods are assigned to (STOPPED).

(FIFOFULL): If the trace recording contains time periods in which FIFO overflow was indicated, these time periods are assigned to (FIFOFULL).

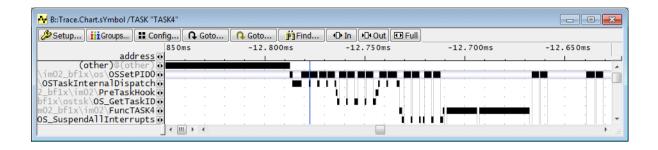
B::Trace.Chart.sYmbol /SplitTASK	Goto	<b>In →□•</b> Out	10 Full		
	Oms	-12.900ms	-12.880ms	-12.860ms	-12.
\\imO2_bf1x\os\OSSetPI					f.
im02_bf1x\ossch\ <b>OSTaskInternalDispat</b> \\im02_bf1x\im02\ <b>PreTaskHo</b>				■ ■ .	
\\imO2_bflx\ostsk\OS_GetTask \\imO2_bflx\imO2\FuncTAS	ID@TASK0		<u>,</u> , , , , , , , ,		
\\imO2_bf1x\ostsk\ <b>O5_ActivateTa</b>	sk@task0		· · ·	. ju u 📻	
	_ < <u> </u>				۲.

B::Trace.Chart.sYmbol /SplitTASK	🕞 Goto 🎁 Find		Full	
Setup 11 Gloups ) an Coning ) 14 Goto	address 💀	-22.624ms	-22.622ms	-22.620ms
(other) \\imO2_bf1x\Global\_start	@(unknown)		• • • •	
\\imO2_bf1x\Global\init_mair \\imO2_bf1x\Global\init_main_guts	@(unknown)		· ·	· · ·
\\im02_bf1x\Global\memset \\im02_bf1x\Global\init	@(unknown)			
, ,		· _		· · · ·

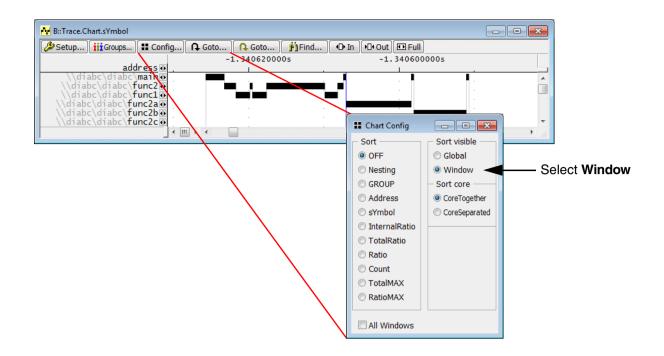
@ <task_name></task_name>	Task name information					
@(unknown)	<ul> <li>Function was running before the OS was started</li> <li>Function was recorded before first task switch information was recorded</li> </ul>					
(UNKNOWN)@	Message decoding not possible.					
(other)@(unknown)	No trace information available.					

Trace.Chart.sYmbol /TASK <task\_name>

Display function time chart for specified task (OS only)



(other)@(other)	All other trace information	
(*****)*		



If **Window** is selected in the **Chart Config** window, the functions that are active at the selected point of time are visualized in the scope of the **Trace.Chart.sYmbol** window. This is helpful especially if you scroll horizontally.

For a detailed description of all options provided by the **Chart Config** window refer to the command description of **Trace.STATistic.Sort**.

If you want to get the time chart only for a few functions, you can use the /Address option to list them.

Trace.Chart.sYmbol /Address <func1>||<func2>||...

🙀 B::Trace.Chart.sYmbol /Address fun	c10  fi	unc2c	func	B  fu	nc13													(	-		×
🖉 Setup 👔 Groups 📲 Config	][[	∩ Go	to	A	Got	D	É	Find.		•••	in )	•□• Ou	t) 🖸	🖻 Ful							
	2000	000	s				-1.	391	900	000	s				-1	. 39	180	000	0 s		
address 💀								1													
(other) 💀																					
\\diabc\diabc\func2c			11	.			1			11											
\\diabc\diabc\func8																					
\\diabc\diabc\func10																					1
\\diabc\diabc\func13																					Ŧ
	III	+ 4																		. +	

More features to structure your trace analysis are introduced in "Structure the Trace Evaluation", page 335.

Analog to the timing diagram also a numerical analysis is provided.

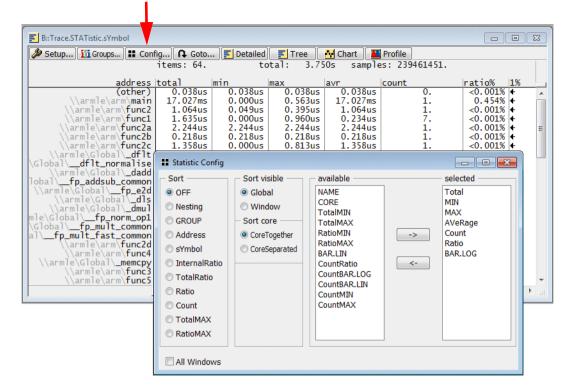
B::Trace.STATistic.sYmbol							×
🌽 Setup) 🏭 Groups) 🏭 Conf					Profile S Init		
	items: 29.	to	tal: 14.97	'4s samp	oles: 207931632.		
address	total	min	max	avr	count	ratio% 1%	
(other)	0.000us	0.000us	-	0.000us	0.	0.000%	
\\demo_r4\demo\ <b>main</b>	36.096ms	0.000us	2.765us	36.096ms	1.	0.241%	
\\demo_r4\demo\func2	7.658us	0.185us	6.090us	7.658us	1.	<0.001% +	=
\\demo_r4\demo\func1	3.387us	0.180us	1.480us	0.484us	7.	<0.001% +	-
\\demo_r4\demo\func2a	6.275us	6.275us	6.275us	6.275us	1.	<0.001% +	
\\demo_r4\demo\func2b	4.800us	4.800us	4.800us	4.800us	1.	<0.001% +	
\\demo_r4\demo\func2d	5.725us	5.725us	5.725us	5.725us	1.	<0.001% +	
_r4\Global\call_via_r0	0.093us	0.093us	0.093us	0.093us	1.	<0.001% +	
\\demo_r4\demo\func3	0.123us	0.123us	0.123us	0.123us	1.	<0.001% +	
\\demo_r4\demo\func5	0.998us	0.998us	0.998us	0.998us	1.	<0.001% +	
\\demo_r4\demo\func8	17.078us	17.078us	17.078us	17.078us	1.	<0.001% +	-
	•		III				H.

survey	
item	number of recorded functions/symbol regions
total	time period recorded by the trace
samples	total number of recorded changes of functions/symbol regions (instruction flow continuously in the address range of a function/symbol region)

Only the grey rows provide useful information about the run-time of a function/symbol range.

function details	
address	function/symbol region name
	<b>(other)</b> program sections that can not be assigned to a function/symbol region
total	time period in the function/symbol region during the recorded time period
min	shortest time continuously in the address range of the function/symbol region
max	longest time continuously in the address range of the function/symbol region
avr	average time continuously in the address range of the function/symbol region (calculated as total/count)
count	number of new entries (start address executed) into the address range of the function/symbol region
ratio	ratio of time in the function/symbol region with regards to the total time period recorded

Pushing the **Config** button provides the possibility to specify a different sorting criterion for the address column or a different column layout. By default the functions/symbol regions are sorted by their recording order.



Trace.STATistic.sYmbol	Flat function run-time analysis (no OS) - numerical display
Trace.STATistic.sYmbol [/MergeTASK]	Flat function run-time analysis (OS) - numerical display - no task information
Trace.STATistic.sYmbol /SplitTASK	Flat function run-time analysis (OS) - numerical display including task information
Trace.STATistic.sYmbol /TASK <task_name></task_name>	Flat function run-time analysis (OS) - numerical display for specified task

## Restrictions

- 1. The nesting analysis analyses only high-level language functions.
- 2. The nested function run-time analysis expects common ways to enter/exit functions.
- 3. The nesting analysis is sensitive with regards to FIFOFULLs.

## **Optimum NEXUS Configuration (No OS)**

The nesting function run-time analysis doesn't require any data information if no OS is used. That's why it is recommended to switch the export of data information off.

NEXUS.DTM OFF

TRACE32 PowerView builds up a separate call tree for each task.

Trace.STATistic.TREE /TASK "Cyclic"

	funcs: 19. total: 5	.892s			
rar	ge tree	total min	max avr	count	intern% 1%
_demo_sieve_40MH2/clobal\05_PAStarTaskh0, 60XP_0s_demo_sieve_40MH2/clemo/05_TASK_Cycl 60XP_0s_demo_sieve_40MH2/demo/05_TASK_Cycl .sieve_40MH2/kern-leavekerne 05_Leavekern _sieve_40MH2/kern-leavekerne 05_Leavekern _sieve_40MH2/kern-leavekerne 05_Leavekern _va_XPC560XP_0s_demo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40MH2/testfun (PA_XPC560XP_sdemo_sieve_40M2/testfun (PA_XPC560XP_sdemo_sieve_40M2/testfun (	<pre>D □ (root)</pre>	5.892s         -           2.340us         -           2.340us         -           5.892s         47.672mc           1.275ms         10.2570s           707.086us         6.137us           3.736ms         6.109us           214.183us         1.732us           5.845s         46.996us           208.287ms         1.398us           208.287ms         1.398us           220.518ms         0.841us           222.518ms         0.450us           72.651ms         0.313us           1.937s         1.188us           1.937s         1.652us           1.937s         1.620ms           1.937s         1.840us           1.937s         1.652us           1.940ms         8.478us	$\begin{array}{llllllllllllllllllllllllllllllllllll$	124, (0/1) 124, 124, 123, 123408, (0/1) 123408, 123408	0.000% <0.001% <0.001% <0.001% <0.008% 0.013% 0.003% <0.003% -0.004% -0.004% -0.004% -0.004% -0.004% -0.004% -0.005% -0.004% -0.005% -0.005% -0.004% -0.005% -0.005% -0.005% -0.005% -0.004% -0.005

In order to hook a function entry/exit into the correct call tree, TRACE32 PowerView needs to know which task was running when the entry/exit occurred.

The standard way to get information on the current task is to advise the NEXUS to export the instruction flow and task switches. For details refer to the chapter "OS-Aware Tracing (ORTI File)", page 193.

Optimum Configuration 1 (if OSEK generated OTMs):

```
NEXUS.OTM ON
; default setting since 2015-01
Trace.STATistic.InterruptIsFunction ON
```

Optimum Configuration 2 (if OSEK does not support OTMs, NEXUS class 3 only):

```
Break.Set TASK.CONFIG(magic) /Write /TraceData
; default setting since 2015-01
Trace.STATistic.InterruptIsFunction ON
```

In order to prepare the results for the nesting analysis TRACE32 post-processes the instruction flow to find:

#### Function entries

The execution of the first instruction of an HLL function is regarded as function entry.

Additional identifications for function entries are implemented depending on the processor architecture and the used compiler.

Trace.Chart.Func

; function func10 as

; example

Trace.List /Track

🚰 B::Trace.Chart.Func /Track	
🌽 Setup 🚺 Groups 🗱 Config 📭 Goto 🎁 Find 🚸 In 🕞 Out 🕅 Full	
00000s -3.27200000s	-3.27150
range 🔂	
\\diabc_ext\diabc\ <b>main</b> ∰ \\diabc_ext\diabc\ <b>func1</b> ₩	
\\diabc_ext\diabc\funcf@	· · · ·
\\diabc_ext\diabc\func70	
\\diabc_ext\diabc\func8	
\\diabc_ext\diabc\func900 \diabc_ext\diabc\func1000	
(root)	
	• • • • •
B:: Trace.List /Track	
🥬 Setup 📭 Goto 🎁 Find 🙌 Chart 🔳 Profile 📃 MIPS 🔷 More 🗶 Less	
record run address cycle data symbol	ti.back
bl 0x20000798 ; tunc10	A
-0000378119 P:20000798 ptrace \\diabc_ext\diabc\func10	2.000us
func10()	-
355 {	~
stwu r1,-0x58(r1) ; r1,-88(r1)	
mflr r0 stmw r14,0x10(r1) ; v16,16(r1)	
stw r0,0x5C(r1) ; r0,92(r1)	
register i, j;	-
	E. 4

#### • Function exits

A RETURN instruction within an HLL function is regarded as function exit.

Additional identifications for function exits are implemented depending on the processor architecture and the used compiler.

🚟 B::Trace.Chart.Func /Track				
Setup iii Groups I Config	🔒 Goto 🎁 Find	d 🔶 In 🕨	Out KN Full	
		-3.2710	00000s	-3.270500000s
range	и и			
\diabc_ext\diabc\func10				
\diabc_ext\diabc\func11				
\diabc_ext\diabc\func1300 \diabc_ext\diabc\func1400				_ · · · · · ·
\diabc_ext\diabc\func15				
\diabc_ext\diabc\func16				
\diabc_ext\diabc\func17				
				- 14
B::Trace.List /Track				
🌽 Setup 🔃 Goto 🎁 Find	🕂 Chart 📕 📕 Prof	file 🛛 🔛 MIPS	More Less	
record run address	cycle data		ymbol	ti.back
	,0x10(r1) ; 0x5C(r1) ;	v16,16(r1) r0,92(r1)		<b>A</b>
mtlr r0	,0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	10,52(11)		
addi r1	r1,0x58 ;	r1,r1,88		•
-0000377985 P:200012A8	ptrace	\	\diabc_ext\diab	c\main+0x170 11.500us
	unc11(5);	r3,5		No. of Control of Cont
		func11		-
				E. A

#### • Entries to interrupt service routines (asynchronous)

If an indirect branch to the Interrupt Vector Table occurs, an interrupt entry is detected. The interrupt function gets the name VTABLE+<offset> if no symbol is specified.

B::Trace.List				• ×
🖉 Setup 🔃 Goto 🎁 F	Find 🚺 🚺 Chart 🕴 🍱 Prot	file 🛛 🔛 MIPS 🔶 More	Less	
record run address	cycle data	symbol	ti.ba	ck 🔄
mtmsr	r3			
→ interru -0000012939 V:40	ipt 0000040 ptrace		obal\VTABLE+0x40 1.	250us
-0000012939 V:40	0x40004FEE	; OSInterruptDispatcher	ODAT VIABLE+0X40 I.	23005 -
	0004FEE ptrace	x\Global\OSIn	terruptDispatcher 0.	750us 🔺
e_stwu	r1,-0x50(r1)	; r1,-80(r1)		
	prw 0x0C(r1)	; 12(r1)		
	prw 0x38(r1) rrw 0x48(r1)	; 56(r1) ; 72(r1)		
e bl	0x40003D50	; OSInterruptDispatcher1		-
4				њ. <i>4</i>
B::Trace.Chart.Func /Track				
Setup iii Groups 🔡 Co	onfig] 📭 Goto 🛛 📭 Goto	o ) 👘 Find   ◀D• In   ▶D• Out	🖸 Full	
			.640ms -12.620ms -1	2.600m
		range 🕕	.640ms -12.620ms -13 I	2.600m
	bf1x\im02\InvokeSciIn	range 🛛 t@TASK4 🖸	.640ms -12.620ms -1 	2.600m
	OS_ResumeAllInterrupt	range 00 t@TASK4 00 s@TASK4 00	.640ms -12.620ms -13	2.600m
\\imO2_bf1x\osisr\( →\\imO2_bf1x\G]	OS_ResumeAllInterrupt (root)@(int obal\VTABLE+0x40@(int	range	640ms -12.620ms -1	2.600m
\\imO2_bf1x\osisr\ →\\imO2_bf1x\Gl \imO2_bf1x\osisr\ <b>OSInte</b>	OS_ResumeAllInterrupt (root)@(int obal\VTABLE+0x40@(int rruptDispatcher1@(int	range () t@TASK4 () errupt) () errupt) () errupt) ()	.640ms -12.620ms -1	2.600m
\\im02_bf1x\osisr\( →\\im02_bf1x\G] \im02_bf1x\osisr\ <b>05Inte</b> \\im02_bf1x\osi	OS_ResumeAllInterrupt (root)@(int obal\VTABLE+0x40@(int rruptDispatcher1@(int tsk\OSCheckStack@(int	range () t@TASK4 () s@TASK4 () errupt)	640ms -12.620ms -1	2.600m
\\imO2_bf1x\osisr\( →\\imO2_bf1x\Gl \imO2_bf1x\osisr\ <b>OSInte</b> \\imO2_bf1x\osi \\imO2_bf1x\osi	OS_ResumeAllInterrupt (root)@(int obal\VTABLE+0x40@(int rruptDispatcher1@(int tsk\OSCheckStack@(int f1X\os\OSSetPID0@(int	range () t@TASK4 () s@TASK4 () errupt) () errupt) () errupt) () errupt) () errupt) () errupt) ()	.640ms -12.620ms -1	2.600m
\\im02_bf1x\osisr\( →\\im02_bf1x\Gl \im02_bf1x\osisr\ <b>OSInte</b> r \\im02_bf1x\osi \\im02_bf1x\osi \\im02_bf1x\osi	OS_ResumeAllInterrupt (root)@(int obal\VTABLE+0x40@(int rruptDispatcher1@(int tsk\OSCheckStack@(int	range	640ms -12.620ms -1	2.600m

• Exits of interrupt service routines

RETURN FROM INTERRUPT is regarded as exit of the interrupt function.

- Entries to TRAP handlers (not implemented yet)
- Exits of TRAP handlers (not implemented yet)
- Task switches

Task switches are needed to build correct call trees if a target operating system is used.

🔑 Setup 🛛 💦	- Goto 🎁 Find 🙌 Chart 🔛 Profile 🔛 MIPS 🔷 M	lore 🗶 Less
record	run address cycle data symbol	ti.back
	se_addi r1,0x10 ; r1,16 se_blr task: Cyclic (00002A94)	- E
-0002382520 -0002382519	owner 00002A94	lispatch\05_Dispatch+0x100 0.530us
	/* Now leaving the kernel. Inter * runs. */	rupts remain disabled until the task
176	05_inKernel = 0; e_lis r6,0x40000000 ; r6,0S_kernStack se_li r7,0x0 ; r7,0 e_stb r7,0x848(r6) ; r7,2123(r6)	

## Numerical Nested Function Run-time Analysis for all Software

Trace.STATistic.Fu		Nested function run-time analysis numeric display
Perf Cov Window Help Perf Configuration Perf List Perf List Dynamic Perf Off	Transvo	1
Function Runtime     Distribution	Prepare Show Numerical	
Duration A to B Distance trace records	Show as <u>T</u> ree Show <u>D</u> etailed Tree	
<u>R</u> eset	Show as Timing Show Nesting	

B::Trace.STATistic.FUNC							x
🌽 Setup 🚻 Groups 🔡 Confi	ig 📭 Goto	. 🗾 Detailed	Nesting	- Chart			
	funcs: 33.	to	tal: 22.76	0s			
	_						
range					count	intern% 1%	_
(root)	22.760s	-	22.760s	22.760s	-	0.000%	-
\\diabc_ext\diabc\main	22.760s	-	22.760s	22.760s	1.(0/1)	0.009% 🗲	
\\diabc_ext\diabc\func2	1.062ms	132.495us	132.830us	132.786us	8.	0.003% 🗲	
\\diabc_ext\diabc\func1	846.290us	13.330us	16.005us	15.112us	56.	0.003% 🗲	
\\diabc_ext\diabc\func2a	563.975us	70.495us	70.500us	70.497us	8.	0.002% +	
\\diabc_ext\diabc\func2b	557.305us	69.660us	69.665us	69.663us	8.	0.002% +	=
\\diabc_ext\diabc\func2c	33.832ms	4.114ms	4.303ms	4.229ms	8.	0.148% +	
\\diabc_ext\diabc\func2d	654.630us	81.825us	81.830us	81.829us	8.	0.002% +	
\\diabc_ext\diabc\func4	241.320us	30.160us	30.170us	30.165us	8.	0.001% +	
\\diabc_ext\diabc\func3	67.995us	8.495us	8.505us	8.499us	8.	<0.001% +	
	170.830us	21.330us	21.495us	21.354us	8.	<0.001% +	
\\diabc_ext\diabc\func6	2.483ms	310.315us	310.320us	310.319us	8.	0.010% +	
\\diabc_ext\diabc\func7	1.733ms	216.655us	216.660us	216.657us	8.	0.007% +	
\\diabc_ext\diabc\func8	933.295us	116.660us	116.665us	116.662us	8.	0.004% +	
\\diabc_ext\diabc\func9	1.077ms	134.660us	134.660us	134.660us	8.	0.002% 🗲	
\\diabc_ext\diabc\ <b>func10</b>	8.732ms	1.091ms	1.092ms	1.092ms	8.	0.038% 🗲	
\\diabc_ext\diabc\ <b>func11</b>	195.985us	24.495us	24.500us	24.498us	8.	<0.001% +	
\\diabc_ext\diabc\func13	883.955us	25.995us	110.495us	68.163us	32.	0.003% 🗲	
\\diabc_ext\diabc\ <b>func14</b>	115.990us	14.495us	14.500us	14.499us	8.	<0.001% 🗲	-
	•		III				Þ:

## funcs: 103. total: 22.618ms intr: 2.574ms 10 workarounds

survey	
funcs: <number></number>	number of functions in the trace
total: <time></time>	total measurement time
intr: <time></time>	total time in interrupt service routines

survey (issue indication)	
stopped: <time></time>	The analyzed trace recording contains program stops. <i><time></time></i> indicates the total time the program execution was stopped.
<number> problems</number>	The nested analysis contains problems. Please contact support@lauterbach.com.
<number> workarounds</number>	The nested analysis contains issues, but TRACE32 found solutions for them. It is recommended to perform a sanity check on the proposed solutions.
stack overflow at <record></record>	The nested analysis exceeds the nesting level 200. It is highly likely that the function exit for an often called function is missing. The command <b>Trace.STATistic.TREE</b> can help you to identify the function. If you need further help please contact <b>support@lauterbach.com</b> .
stack underflow at <record></record>	The nested analysis exceeds the nesting level 200. It is highly likely that the function entry for an often executed function is missing. The command <b>Trace.STATistic.TREE</b> can help you to identify the function. If you need further help please contact <b>support@lauterbach.com</b> .

The main reasons for all the issues are code optimizations.

columns	
range (NAME)	function name, sorted by their recording order as default

HLL function

# \\diabc\_ext\diabc\func6

• (root)

# (root)

The function nesting is regarded as tree, (root) is the root of the function nesting.

Interrupt service routine

\\PA\_XPC560XP\_os\_demo\_sieve\_40MHz\Global\OS\_InterruptTable+0x78

• HLL trap handler (not implemented yet)

B::Trace.STATistic.FUNC							×
Setup III Groups II Con	fig 📭 Goto	. F Detailed	Nesting	Chart			
funcs: 33. total: 22.760s							
range total min max avr count intern% 1%							
(root)	22.760s	-	22.760s	22.760s	-	0.000%	
\\diabc_ext\diabc\main		-	22.760s	22.760s	1.(0/1)	0.009% 🗲	
\\diabc_ext\diabc\func2		132.495us	132.830us	132.786us	8.	0.003% 🗲	
\\diabc_ext\diabc\func1		13.330us	16.005us	15.112us	56.	0.003% +	
\\diabc_ext\diabc\func2a		70.495us	70.500us	70.497us	8.	0.002% +	
\\diabc_ext\diabc\func2b		69.660us	69.665us	69.663us	8.	0.002% +	=
\\diabc_ext\diabc\func2c		4.114ms	4.303ms	4.229ms	8.	0.148% +	-
\\diabc_ext\diabc\func2d		81.825us	81.830us	81.829us	8.	0.002% +	
\\diabc_ext\diabc\func4	241.320us	30.160us	30.170us	30.165us	8.	0.001% +	
\\diabc_ext\diabc\func3		8.495us	8.505us	8.499us	8.	<0.001% +	
\\diabc_ext\diabc\func5		21.330us	21.495us	21.354us	8.	<0.001% +	
\\diabc_ext\diabc\func6		310.315us	310.320us	310.319us	8.	0.010% +	
\\diabc_ext\diabc\func7		216.655us	216.660us	216.657us	8.	0.007% +	
\\diabc_ext\diabc\func8		116.660us	116.665us	116.662us	8.	0.004% +	
\\diabc_ext\diabc\func9		134.660us	134.660us	134.660us	8.	0.002% +	
\\diabc_ext\diabc\func10		1.091ms	1.092ms	1.092ms	8.	0.038% +	
\\diabc_ext\diabc\func11		24.495us	24.500us	24.498us	8.	<0.001% +	
\\diabc_ext\diabc\func13		25.995us	110.495us	68.163us	32.	0.003% +	
\\diabc_ext\diabc\func14	115.990us	14.495us	14.500us	14.499us	8.	<0.001%  ←	-
	•						

columns (cont.)	
total	total time within the function
min	shortest time between function entry and exit, time spent in interrupt service routines is excluded
	No <b>min</b> time is displayed if a function exit was never executed.
max	longest time between function entry and exit, time spent in interrupt service routines is excluded
avr	average time between function entry and exit, time spent in interrupt service routines is excluded

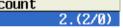
B::Trace.STATistic.FUNC							×
Setup iii Groups II Con	fig 🖪 Goto	. F Detailed	E Nesting	- Chart			
funcs: 33. total: 22.760s							
		min		avr	count	intern% 1	<u>%</u>
(root)	22.760s	-	22.760s	22.760s		0.000%	
\\diabc_ext\diabc\main		-	22.760s	22.760s	1. (0/1)	0.009% 🕇	
\\diabc_ext\diabc\func2		132.495us	132.830us	132.786us	8.	0.003%	
\\diabc_ext\diabc\func1		13.330us	16.005us	15.112us	56.	0.003%	
\\diabc_ext\diabc\func2a		70.495us	70.500us	70.497us	8.	0.002%	
\\diabc_ext\diabc\func2b		69.660us	69.665us	69.663us	8.	0.002%	
\\diabc_ext\diabc\func2c		4.114ms	4.303ms	4.229ms	8.	0.148%	
\\diabc_ext\diabc\func2d	654.630us	81.825us	81.830us	81.829us	8.	0.002%	
\\diabc_ext\diabc\func4	241.320us	30.160us	30.170us	30.165us	8.	0.001% +	
\\diabc_ext\diabc\func3		8.495us	8.505us	8.499us	8.	<0.001%	
\\diabc_ext\diabc\func5	170.830us	21.330us	21.495us	21.354us	8.	<0.001%	
\\diabc_ext\diabc\func6		310.315us	310.320us	310.319us	8.	0.010%	
\\diabc_ext\diabc\func7		216.655us	216.660us	216.657us	8.	0.007%	
\\diabc_ext\diabc\func8		116.660us	116.665us	116.662us	8.	0.004%	
\\diabc_ext\diabc\func9		134.660us	134.660us	134.660us	8.	0.002%	
\\diabc_ext\diabc\func10		1.091ms	1.092ms	1.092ms	8.	0.038% 🗲	
\\diabc_ext\diabc\func11		24.495us	24.500us	24.498us	8.	<0.001% 🗲	
\\diabc_ext\diabc\func13	883.955us	25.995us	110.495us	68.163us	32.	0.003% 🗲	
\\diabc_ext\diabc\func14	115.990us	14.495us	14.500us	14.499us	8.	<0.001%	-
	•		III				M. 422

columns (cont.)	
count	number of times within the function

If function entries or exits are missing, this is displayed in the following format:

<ti>times within the function >. (<number of missing function entries>/<number of missing function exits>).

count



#### Interpretation examples:

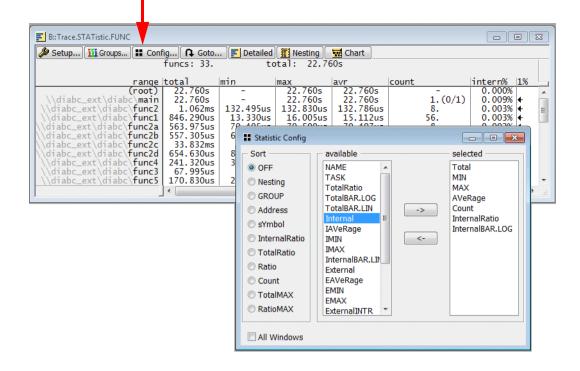
- 1. 2. (2/0): 2 times within the function, 2 function entries missing
- 2. 4. (0/3): 4 times within the function, 3 function exits missing
- 3. 11. (1/1): 11 times within the function, 1 function entry and 1 function exit is missing.



If the number of missing function entries or exits is higher the 1 the analysis performed by the command Trace.STATistic.Func might fail due to nesting problems. A detailed view to the trace contents is recommended.

columns (cont.)	
intern% (InternalRatio, InternalBAR.LOG)	ratio of time within the function without subfunctions, TRAP handlers, interrupts

#### Pushing the Config... button allows to display additional columns

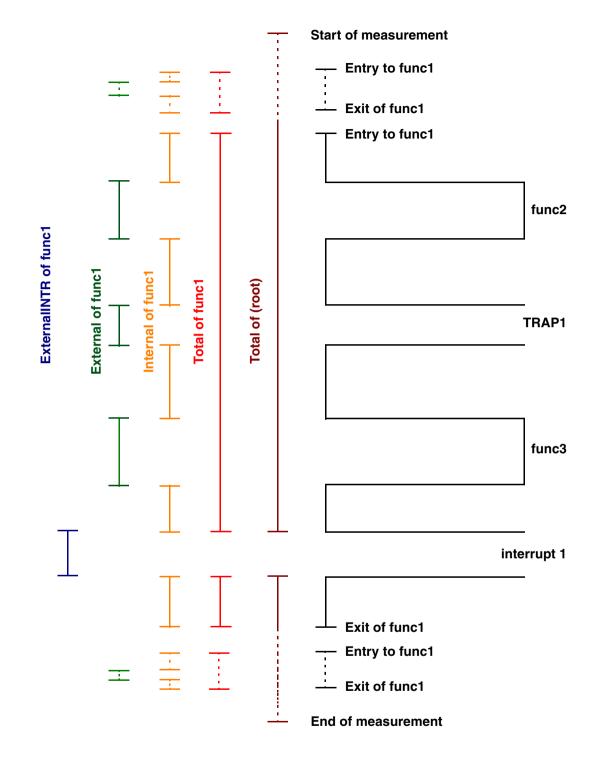


columns (cont.) - times only in function	
Internal	total time between function entry and exit without called sub-functions, TRAP handlers, interrupt service routines
IAVeRage	average time between function entry and exit without called sub- functions, TRAP handlers, interrupt service routines
IMIN	shortest time between function entry and exit without called sub- functions, TRAP handlers, interrupt service routines
ΙΜΑΧ	longest time spent in the function between function entry and exit without called sub-functions, TRAP handlers, interrupt service routines
InternalRatio	<internal function="" of="" time="">/<total measurement="" time=""> as a numeric value.</total></internal>
InternalBAR	<internal function="" of="" time="">/<total measurement="" time=""> graphically.</total></internal>

columns (cont.) - times in sub-functions and TRAP handlers	
External	total time spent within called sub-functions/TRAP handlers
EAVeRage	average time spent within called sub-functions/TRAP handlers
EMIN	shortest time spent within called sub-functions/TRAP handlers
ЕМАХ	longest time spent within called sub-functions/TRAP handlers

columns (cont.) - interrupt times	
ExternalINTR	total time the function was interrupted
ExternalINTRMAX	max. time one function pass was interrupted
INTRCount	number of interrupts that occurred during the function run-time

The following graphic give an overview how times are calculated:



B::Trace.STATistic.FUNC								×
🌽 Setup 👖 Groups 🔡 Config 🔃 Goto 🗾 Detailed 📳 Nes	ting 🔚 🚟 Cha	rt						
	funcs: 113	. to	tal: 6.95	1s intr:	192.665ms			
range	total	min		avr	count	intern% 1%	2%	
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func8@Cyclic	72.651ms	0.313us	1.127us	0.589us	123408.	1.045%		
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func9@Cyclic			2.113us	1.590us	123408.	2.822%		
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func10@Cyclic		15.293us	16.605us	15.692us	123408.	27.861%		_
\\PA_XPC560XP_os_demo_sieve_40MHz\test\sieve@Cyclic	2.396s	18.652us	20.149us	19.414us	123408. (0/1)	34.469%		_
z\kern-releaseresource\OS_KernReleaseResource@Cyclic	1.262ms	10.157us	10.314us	10.258us	123.	0.018% 🗲		
40MHz\kern-terminatetask\05_KernTerminateTask@Cyclic	1.049ms	8.478us	8.582us	8.531us	123.	0.015% +		
demo_sieve_40MHz\Global\OS_PAStartTask+0x30@Task_St1	2.400us	-	2.400us	2.400us	1.(1/0)	<0.001% +		
(root)@Task_St1	1.127ms	-	1.127ms	1.127ms		0.000%		
P_os_demo_sieve_40MHz\demo\OS_TASK_Task_St1@Task_St1	1.124ms	74.985us	75.265us	74.943us	15.(0/1)	<0.001% +		
\\PA_XPC560XP_os_demo_sieve_40MHz\test\test@Task_St1	711.729us	47.355us	47.570us	47.449us	15.	<0.001% +		-
	•							► at

HLL function

\\PA\_XPC560XP\_os\_demo\_sieve\_40MHz\test\func2@Cyclic

HLL function "func2" running in task "Cyclic"

• Root of call tree for task "Cyclic"



E: Trace.STATistic.FUNC						_   0 .
🌽 Setup 🛛 🏭 Groups 🕽 🎛 Config 🕻 Goto 📄 Detailed 🗱 Nesting 🖉 Chart						
	funcs: 113.	. to	otal: 6.9	51s intr:	192.665ms	
range		min	max		count	intern%
(root)@(unknown)	160.813us	-	160.813us	160.813us		0.000% 🔺
\\PA_XPC560XP_os_demo_sieve_40MHz\demo\main@(unknown)	160.813us	-	160.813us	160.813us	1.(0/1)	<0.001%
\\PA_XPC560XP_os_demo_sieve_40MHz\demo_arch\ArchInitHardware@(unknown)	0.000us	0.000us	-	0.000us	1.	0.000%
\\PA_XPC560XP_os_demo_sieve_40MHz\kern-startos\05_KernStart0s@(unknown)	146.413us	146.413us	146.413us	146.413us	1.	<0.001% -
	•	III				E M

Unknown task

\\PA\_XPC560XP\_os\_demo\_sieve\_40MHz\demo\main@(unknown)

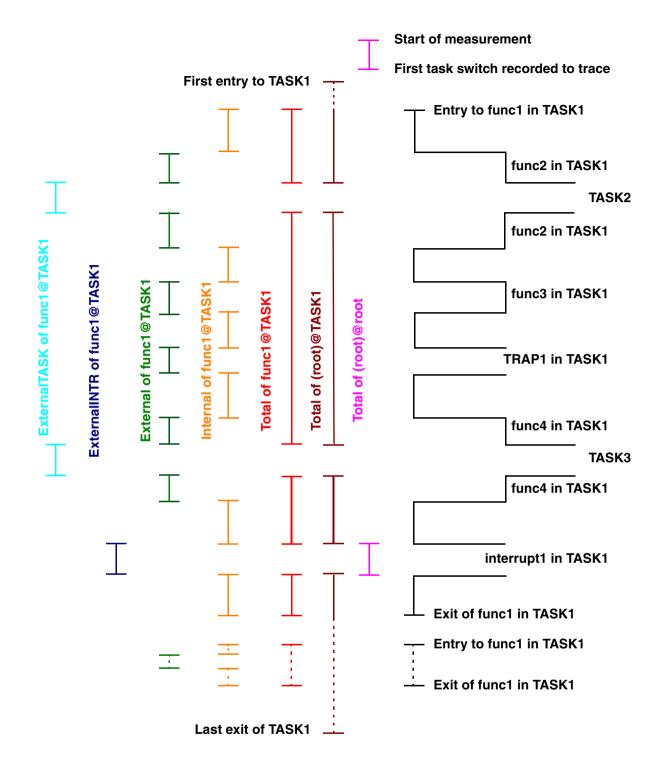
Before the first task switch is found in the trace, the task is unknown

Root of unknown task

(root)@(unknown)

B::Trace.STATistic.FUNC										x
🖉 Setup 🚺 Groups 🔡 Config	g 🕻 🕻 Goto	Detailed	🛐 Nesting 📔	- Chart						
	funcs: 113.	to	otal: 6.95	intr:	192.665ms					
range		taskcount			min			count	intern% 1%	
OS_PAStartTask+0x30@Loop	2.285us	1.	0.000us	0.000us	-	2.285us	2.285us	1. (1/0)	<0.001% +	
(root)@Loop	863.762ms	125.	6.058s	6.058s	-	863.762ms	863.762ms	-	0.000%	
Hz\demo\OS_TASK_Loop@Loop	863.760ms	124.	6.058s	6.058s	-	863.760ms	863.760ms	1. (0/1)	<0.001% +	
EndlessLoopInternal@Loop	863.759ms	124.	6.058s	6.058s	-	863.759ms	863.759ms	1. (0/1)	0.078% +	
ieve_40MHz\test\test@Loop	858.284ms	124.	6.058s	49.268ms	46.924us	48.063us	47.268us	18158. (0/1)	0.943% +	
eve_40MHz\test\func2@Loop	27.401ms	-	-	-	1.330us	1.904us	1.509us	18158.	0.394%	التبا
ve_40MHz\test\func2a@Loop	30.654ms	5.	245.764ms	49.246ms	1.544us	2.176us	1.688us	18158.	0.441% +	
ve_40MHz\test\func2b@Loop	17.380ms	-	-	-	0.731us	1.409us	0.957us	18158.	0.250% +	
ve_40MHz\test\func2d@Loop	32.744ms	4.	196.311ms	49.207ms	1.599us	2.410us	1.803us	18158.	0.471% +	
eve_40MHz\test\func3@Loop	7.442ms	-	-	-	0.405us	0.470us	0.410us	18158.	0.107% +	-
	•			III					)	E at

columns - task/thread related information						
TASKCount	number of tasks that interrupt the function					
ExternalTASK	total time in other tasks					
ExternalTASKMAX	max. time 1 function pass was interrupted by a task					



The standard NEXUS settings do often not allow to locate exactly the instructions that are already executed by a newly activated task. This is especially true is Branch History Messaging is used. This might disturb the task-aware function run-time measurement.

An instruction-accurate assignment of the task switches may improve the results.

#### IEEE-ISTO 5001-2008 and Subsequent Standards

The Ownership Trace Messages (task switches) can be exactly assigned to an instruction, if the following setting is done.

NEXUS.PTCM PID\_MSR ON

NEXUS.POTD ON

- ; enable Program Trace Correlation
- ; Messages for PID0/NPIDR accesses
- ; disable Periodic Ownership Trace
- ; Messages

B::Trace.Lis	t List.TASK DEFault	
🌽 Setup	📭 Goto 🛱 Find 🚺 Chart 🛛 🛄 Profile 🛛 🔜 MIPS 🛛 🖨 More 🛛 🗶 Less	
record	run address cycle data symbol	ti.back
-00013778	se_isync V:400005BE ptrace \\im02_bf1x\os\OSSetPID0+0x0	6 0.160us
	mtpid r3 ; value	
-00013777	task: TASKO (00000009) V:400005C2 owner 00000009 \\im02_bf1x\os\OSSetPID0+0x0	0A 0.300us
-00013775	se_isync V:400005C8 ptrace \\im02_bf1x\os\OSSetPID0+0x1	10 0.200us -

#### Alternative

; mark instruction that performs the task switch for the task-aware

; function run-time analysis

sYmbol.MARKER.Create TASKSWITCH osDispatcher+0x100

🛓 B::sYmbol.MAR	KER.List 🗖 🗖 🗾	×
address	info	
P:00023578	TASKSWITCH	*
		Ŧ
	4 F	

TRACE32 analyzes the structure of the program execution by processing the trace information in order to provide the nesting statistic. The objective is to construct a complete call tree. When a OS is used, it is more likely the TRACE32 has issues while construction the call tree. There are two types of issues:

#### PROBLEMS

A PROBLEM is a point it the trace recording that TRACE32 can not integrate into the current nesting. TRACE32 does not discard this point for the call tree, it integrates this point by assigning a meaningful interpretation.

TRACE32 marks functions that include a PROBLEM with ! in the count column.

### WORKAROUNDS

A WORKAROUND is a point it the trace recording that TRACE32 can not integrate into the current nesting. But TRACE32 integrates this point into the function nesting, by supplementing information based on previous scenarios in the nesting. TRACE32 marks functions that include a WORKAROUND with **?** in the count column.

It is recommended to drag the count column wider to see all details.

B::Trace.STATistic.FUNC /TASK	Task2"						
Setup iii Groups II Cor	nfig 🔒 Goto.	F Detailed	Nesting	Chart			
	funcs: 11.	to	otal: 6.41	.7s 1 prob	olems		
		min		avr	count	<⇒dintern% 1%	2
(root)@(other)		-	6.417s	6.417s	-	- +	
EE_rq2stk_exchange@Task2	0.620us	-	0.620us	0.620us	1. (1/0)	<0.001% +	
(root)@Task2		-	22.560us	22.560us	-	<0.001% +	
std_change_context@Task2	21.580us	-	21.580us	21.580us	1. (0/1)	<0.001% +	
std_run_task_code@Task2	16.140us	-	16.140us	16.140us	1.(0/1)	<0.001% +	
EE_oo_thread_stub@Task2	15,400us	-	15.400us	15.400us	1. (0/1)	<0.001% +	
_terminate_savestk@Task2		-	14.420us	14.420us	1.(0/1)	<0.001% +	
ppc\code\FuncTask2@Task2		-	11.840us	11.840us	1. (0/1)	<0.001% +	
E_oo_TerminateTask@Task2		-	10.360us	10.360us	1. (0/1,1!		
hal_terminate_task@Task2		3.100us	3.100us	3.100us	1. (0/1,1.	<0.001% +	
hread_end_instance@Task2		5.10003	3.820us	3.820us	1. (0/1)	<0.001%	
ni eau_enu_mscance@raskz	J. 02003	_	5.02003	J. 02003	1.(0/1)	<0.001%   <del>•</del>	-
	1						
1	•		III				h. 🗶
						a   n	

The following two TRACE32 windows are recommended if you want to inspect the issues:

Trace.ListNesting	
Trace.List List.TASK List.ADDRESS List.sYmbol DEFau	ilt /Track

**Example 1:** We inspect the problem with the function EE\_oo\_TerminateTask.

We start to search for the entry to the function EE\_oo\_TerminateTask in the task Task2 in the **Trace.ListNesting** window.

Trace Find								
Expert	Ocycle	Group	Changes	🔘 Up				
			Signal	Own				
address / exp	ression							
EE_oo_Termi	inateTask		•	🚹 🗆 HLL				
Cycle	Da	ta						
	•			-				
Find Next F	Find First F	ind Here Find	All Clear	Cancel				
B::Trace.ListN								
	lesting		TREE		Profile		<b>♦</b> More	Less
Setup	lesting				Profile	MIPS	◆ More	
Setup 6	lesting				Profile	MIPS	◆ More	¥ Less
© Setup	lesting				Profile	MIPS	◆ More	¥ Less
Setup	lesting	₽̂jFind ) [ ₽]			Profile	MIPS	◆ More )	¥ Less
Setup record ************************************	Goto	ĵjFind) <b>F</b> ∣			Profile			X Less ti.back
Setup	Goto	řjFind] <b>F</b> ∣ tart_ m_setup			Profile			Less ti.back
Setup)     record     re	Goto	FjFind) [₽] tart msetup			Profile			Less  ti.back
Setup         F           record         F </td <td>Goto</td> <td>fjFind) <b>F</b> tart m_setup _setup</td> <td>TREE . Al Cha</td> <td></td> <td>Profile</td> <td></td> <td></td> <td>Less  ti.back 0.740us 0.740us 5.660us</td>	Goto	fjFind) <b>F</b> tart m_setup _setup	TREE . Al Cha		Profile			Less  ti.back 0.740us 0.740us 5.660us
Setup)     record     re	Goto	<pre>#jFind ) F #jFind ) F tart m_setup setup setup mmyFrl+0x1B/</pre>	TREE . Al Cha		Profile			Less ti.back
record ******** 04172160 € 04172159 € 04172158 04171841 € 04171839 04171837 €	Goto	fjFind) Fj tart- m_setup setup setup mmyFnl+0x1B/ c_setup			Profile			Less  ti.back 0.740us 0.740us 5.660us 0.740us 0.740us
Setup) f record ********* 04172160 @ 04172159 @ 04172158 04171841 @ 04171837 @ 04171837 @	Goto	tart m_setupsetup setupsetup setup csetup+0x184 csetup+0x14			Profile			Less ti.back
Setup) (* record ******** 04172160 @ 04172159 @ 04172158 04171841 @ 04171837 @ 04171837 @	Goto	tart m_setupsetup setupsetup setup csetup+0x184 csetup+0x14			Profile			Less  ti.back 0.740us 0.740us 5.660us 0.740us 0.740us



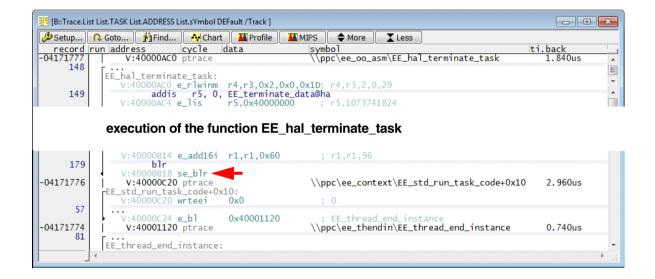
B::Trace.List	
record	ti.back
-04171782 -04171782 -04171780	B       Image: Contract Contrel Contract Contrel Contract Contract Cont
-04171780 (# -04171777 (# -04171776 -04171776 -04171774 (#	EE_hal_terminate_task+0x58         2.960us           EE_oo_TerminateTask (!)         1

In the screenshot above we can see that the exit from the function EE\_oo\_TerminateTask is marked as a problem. Why is that?

Let's examine the function EE\_oo\_TerminateTask by looking to the trace listing.

	Goto	Chart	📕 🔛 Profile	MIPS 🔶 More	Less		
record	run address		lata	symbol		ti.back	
106	V:40001000 EE_oo_Terminat V:400010EC V:400010F4 V:400010F4 V:400010F8	eTask+0x60 mfmsr wrteei e_lis	r8 0x0 r8,0x40000000	; np_flags ; 0 ; np_flags : r0.4			(
103 04171778 04171777	V:400010FA V:400010FE V:40001100 V:40001104	e_add16i se_bmaski stwx se_stb e_b1 rd-long	r8,r8,0x2364 r5,0x0 r5,r8,r6 r0,0x0(r4) 0x40000AC0	; np_flags ; r5,0 ; r5,np_fla ; r0,0(r4) ; EE_hal_t(	erminate_task 🥌	2.840us	
148 149	addis	e_rlwinm; r5, 0,	r4,r3,0x2,0x0, EE_terminate_d r5,0x40000000	ata@ha			•

In its execution the function EE\_oo\_TerminateTask calls the function EE\_hal\_terminate\_task.



Now one would expect the function EE\_hal\_terminate\_task returns with the se\_blr instruction to the calling function (which was EE\_oo\_TerminateTask). But if we look at the trace listing, we see that the program execution continued in the middle of the function EE\_std\_run\_task\_code.

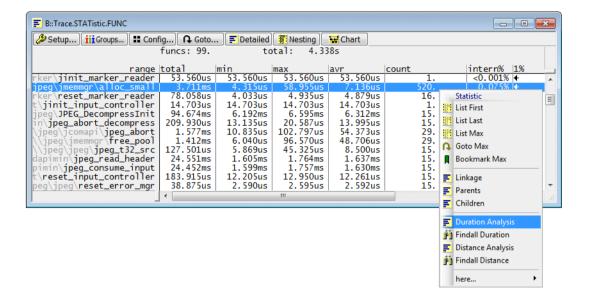
But since function EE\_std\_run\_task\_code+0x10 does not fit into the call tree and the function EE\_oo\_TerminateTask does not continue later in the trace recording, TRACE32 adds the function exit of EE\_oo\_TerminateTask to the call tree and marks it with !.

🌡 Setup 🛛 🗛 G	oto 🛉 Find 🗧 TREE 🛃 Chart 🔛 Chart 🔛 Profile	MIPS 🔶 More 🗶	Less
record			ti.back
	———— task: Task2 (00000001) —————		
04171786	EE_rg2stk_exchange+0x24	A ▼	
)4171784 🕀 🖃	-EE_std_change_context	\1▲ ▼	
)4171783 ⊞ 🗉	EE_std_run_task_code	\1 <b>_</b>	
)4171782 🕀 🖃	EE_oo_thread_stub	\1 <b>_</b>	
4171782 ⊞ ⊟	EE_hal_terminate_savestk	\ <u>1</u>	
4171780 ⊞ ⊟	FuncTask2	\146▲ ▼	
)4171780 ⊞ ⊟	= EE_oo_TerminateTask	\1	
4171777 <b>H</b>	_EE_hal_terminate_task	\120 <b>A V</b>	- 2.960us
4171776	EE_hal_terminate_task+0x58	·····	- 2.960us
4171776	EE_oo_TerminateTask (!)	\1	
04171774 <b>E</b>	EE_thread_end_instance	\î <b>* *</b>	
	task: NO_TASK (FFFFFFF)	<u></u>	

### Look and Feel (No OS)

Trace.STATistic.FuncDURation <function>

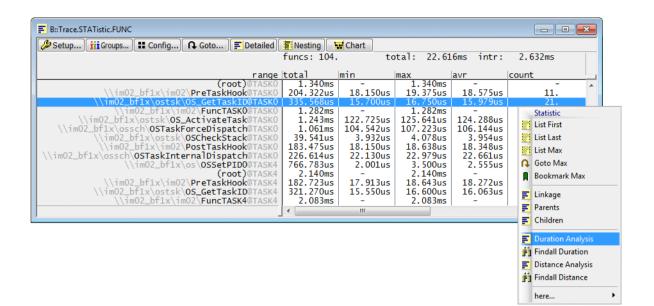
Detailed analysis of a single function, time between function entry and exit, time spent in interrupt service routines is excluded.



B	Setup	Chart 🗘 🏮	Zoom 🛓	Zoom	Full						
		samples: total:	520. 4.338s	avr: in:	7.136us 3.711ms	min: out:	4.315us 4.335s	max: ratio:	58.955us 0.085%		
		count	ratio	1%	2%	5%	10%	20%	50%	100	
٢.	0.000us	0.									
	5.000us	194.							•		
	10.000us	308.									
	15.000us	1.									
	20.000us 25.000us	0.									
	30.000us	1.									
	35.000us										
	40.000us	14									
	45.000us	0									
	50.000us	ŏ.									
	55.000us	ŏ									
	60.000us	1									
	65.000us	0.									
	70.000us	0.	0.000%								
	75.000us	0.	0.000%								
	80.000us	0.	0.000%								
•		0.	. 0.000%								

Detailed analysis of a single function, time between function entry and exit, time spent in interrupt service routines and other tasks is excluded.

Trace.STATistic.FuncDURation <function>[/TASK "<task\_name>"]

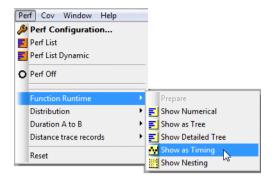


F	B::Trace.STAT.F	uncDURation P:	)x40000B68								x
	🔑 Setup 🛛 🎼			.oom	Full		15 550		10.750	10	_
		samples: total:	101. 22.616ms	avr: in:	16.031us 1.619ms	min: out:	15.550us 20.996ms	max: ratio	16.750us : 7.159%	10	w
-	<u>up to</u> 15.500us	count 0.	ratio 0.000%	1%	2%	5%	10%	20%	50%	100	
ſ	15.600us 15.700us	10. 0.	9.900%								
	15.800us 15.900us	10. 25.	9.900%								
	16.000us 16.100us	9. 6.	8.910% 5.940%				-				
	16.200us 16.300us	5.	4.950%			_					
	16.400us 16.500us	11. 13.	10.891%								
	16.600us 16.700us	1.	0.990%	<b>←</b>							
	16.800us 16.900us	3. 0.	2.970% 0.000%								
	17.000us 17.100us	0. 0.	0.000%								
>		0.	0.000%							•	

Please be aware, that details are shown for all function runs. If you are interested in a task-specific analysis, you have to use the **/TASK** "<task\_name>" option.

F B	::Trace.STAT.F	uncDURation F	0:0x40000B68 /1	ASK "TA	SKO"						×
B	Setup 🛛 📶	Chart 🏮	Zoom 🔁	Zoom	Full						
		samples: total:	21. 22.616ms	avr: in:	15.979us 335.568us	min: out:	15.700us 22.280ms	max: ratio	16.750us : 1.483%	10 \	N
		count	ratio	1%	2%	5%	10%	20%	50%	100	
<	15.700us 15.800us	0.	19.047%					-			*
	15.900us 16.000us	7.									
	16.100us	0.									
	16.200us 16.300us	0									
	16.400us 16.500us	1.0				-					
	16.600us	0.	0.000%								
	16.700us 16.800us	0.									
	16.900us 17.000us	0.									
	17.100us	0.	0.000%								
	17.200us 17.300us	0.									
>		0.									Ŧ
		•								•	H.

Trace.Chart.Func
------------------



## Look and Feel (No OS)

🎾 Setup 🚺 Groups 🔡 Config	g 📭 G	oto ]	Find	I 🕪 In	► Out	<b>KN</b> Full		
	480.000	)us 50	0.000us	520.	000us	540.000u	s 560.000	Dus
r ange 🕖						1		
(root) 🕢								
\\diabc_ext\diabc\main@-		_		<u> </u>				-
\diabc_ext\diabc\ <b>func2</b> 🔃								
\diabc_ext\diabc\func1								
diabc_ext\diabc\func2a								
diabc_ext\diabc\ <b>func2b</b>								
diabc_ext\diabc\func2c								
diabc_ext\diabc\ <b>func2d</b>								
\diabc_ext\diabc\ <b>func4</b>								-
\diabc_ext\diabc\ <b>func3</b>								
\diabc_ext\diabc\func5								
\diabc_ext\diabc\ <b>func6</b>								
\diabc_ext\diabc\func7								
\diabc_ext\diabc\func8								
\diabc_ext\diabc\func9								_
diabc_ext\diabc\func10								
diabc_ext\diabc\func11 🚯								

## Look and Feel (OS)

23970000s	-6.8239600
a a a a a <mark>a</mark> a	and the second second
<mark></mark>	A A

Per	f Cov Window Help		
لا	Perf Configuration		
F	Perf List		1
=	Perf List Dynamic		
0	Perf Off		
	Function Runtime	Þ	Prepare
	Distribution	•	Show Numerical
	Duration A to B	•	Show as Tree
	Distance trace records	•	Show Detailed Tree 5
	Reset		Show as Timing

## Look and Feel (No OS)

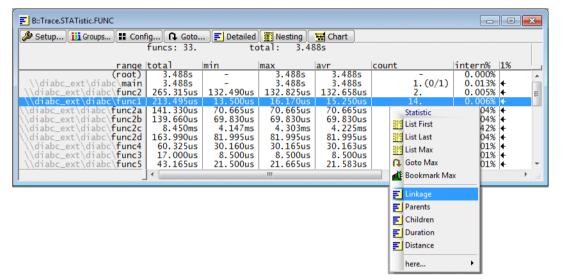
B::Trace.STATistic.TREE						×
🌽 Setup ) 👖 Groups 🚼 Config 📭 Goto 🛒 Detailed 🖉 Nesting	Chart					
funcs: 37. total: 3.	488s					
range tree		min			count	
(root) 🗉 (root)	3.488s	-	3.488s	3.488s		
\\diabc_ext\diabc\main 🖵 🖻 main	3.488s		3.488s	3.488s	1.(0/1)	
\\diabc_ext\diabc\func2 - func2	265.315us	132.490us	132.825us	132.658us	2.	
\\diabc_ext\diabc\func1 func1	86.500us	13.500us	16.170us	14.417us	6.	
\\diabc_ext\diabc\func2a - func2a	141.330us	70.665us	70.665us	70.665us	2.	
\\diabc_ext\diabc\func2b func2b	139.660us	69.830us	69.830us	69.830us	2.	
\\diabc_ext\diabc\func2c func2c	8.450ms	4.147ms	4.303ms	4.225ms	2.	E
\\diabc_ext\diabc\func2d func2d	163.990us	81.995us	81.995us	81.995us	2.	
\\diabc_ext\diabc\func4 \\diabc_ext\diabc\func3	60.325us 17.000us	30.160us 8.500us	30.165us 8.500us	30.163us 8.500us	2. 2. 2.	
\\diabc_ext\diabc\func5	43.165us	21.500us	21.665us	21.583us	2.	
\\diabc_ext\diabc\func6	620.640us	310.320us	310.320us	310.320us	2.	
\\diabc_ext\diabc\func7	433.645us	216.820us	216.825us	216.823us	5.	1
\\diabc_ext\diabc\func8	233. 320us	116.660us	116.660us	116.660us	5.	
$\langle diabc = func9$	269.650us	134.825us	134.825us	134.825us	2.	
\\diabc_ext\diabc\func1 - func1	126.995us	15.000us	16.170us	15.874us	2. 2. 2. 8.	
\\diabc_ext\diabc\func10 - func10	2.183ms	1.092ms	1.092ms	1.092ms	2	
\\diabc_ext\diabc\func11 func11	48.995us	24.495us	24.500us	24.498us	2.	
$\langle diabc_ext \langle diabc \rangle func13 = 0 func13$	220.990us	110.495us	110.495us	110.495us		
\\diabc_ext\diabc\func13	164.655us	82.325us	82.330us	82.328us	2. 2. 2. 2.	
\\diabc_ext\diabc\func13	108.330us	54.160us	54.170us	54.165us	2.	
\\diabc_ext\diabc\func13	52.330us	26.165us	26.165us	26.165us	2.	-
(	1					
						-11

B::Trace.STATistic.TREE		
👂 Setup 🚺 Groups 🚼 Config 📭 Goto ) 🗾 Detailed 📳 Nesting 🛛 🖼 Chart 🛛	funcs: 121. total: 6	.951s intr
	TUNCS: 121. LOLAT: 6	apre incr
	ange tree	total
	clic 🖂 (root)	5.892s
\\PA_XPC560XP_os_demo_sieve_40MHz\Global\ <b>0S_PAStartTask+0x30</b> @Cy		2.340us
\\PA_XPC560XP_os_demo_sieve_40MHz\demo\ <b>OS_TASK_Cyclic</b> @Cy	clic B 0S_TASK_Cyclic	5.892s
\\PA_XPC560XP_os_demo_sieve_40MHz\kern-getresource\OS_KernGetResource@Cy	clic - OS_KernGetResource	1.275ms
\\PA_XPC560XP_os_demo_sieve_40MHz\kern-getresourcefromtask\ <b>OS_GetResourceFromTask</b> @Cy	clic _ 05_GetResourceFromTask	
\\PA_XPC560XP_os_demo_sieve_40MHz\kern-leavekernel\OS_LeaveKernel@Cy	clic - OS_LeaveKernel	3.736ms
\\PA_XPC560XP_os_demo_sieve_40MHz\kern-leavekernel\ <b>OS_LeaveKernel</b> @Cy		214.183us 5.845s
\\PA_XPC560XP_os_demo_sieve_40MHz\test\ <b>test</b> @Cy \\PA_XPC560XP_os_demo_sieve_40MHz\test\ <b>func2</b> @Cv		195.175ms
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func2a@Cv		208.287ms
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func2b@Cy		118.091ms
\\PA_XPC560XP_os_demo_sieve_40MHz\test\ <b>func2d</b> @Cv		222.518ms
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func3@Cv		50.566m
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func8@Cv		72.651m
\\PA_XPC560XP_os_demo_sieve_40MHz\test\ <b>func9</b> @Cy		196.202m
\\PA_XPC560XP_os_demo_sieve_40MHz\test\func10@Cv		1.937s
\\PA_XPC560XP_os_demo_sieve_40MHz\test\sieve@Cy		2.396s
\\PA_XPC560XP_os_demo_sieve_40MHz\kern-releaseresource\05_KernReleaseResource@Cy		1.262ms
\\PA_XPC560XP_os_demo_sieve_40MHz\kern-terminatetask\ <b>OS_KernTerminateTask</b> @Cy		1.049ms
	_St1 🗉 (root)	1.127ms
<pre>\\PA_XPC560XP_os_demo_sieve_40MHz\Global\0S_PAStartTask+0x30@Task</pre>		2.400us
\\PA_XPC560XP_os_demo_sieve_40MHz\demo\ <b>0S_TASK_Task_St1</b> @Task	_St1 🖵 🗉 OS_TASK_Task_St1	1.124ms
	< III	F

It is also possible to get a task-specific tree.

Trace.STATistic.TREE /TASK "Cyclic"

### Look and Feel (No OS)



B::Trace.STAT.LINKage C:	0x200000	)4C						_	
🌽 Setup 🚺 Groups		ig 📭 Goto funcs : 2.		Nesting					
								le a Tor le	~
	range			max		count		total% 1	.%
\\diabc_ext\diabc\	func2	86.500us	13.500us	16.170us	14.417us		6.	40.516%	A
\\diabc_ext\diabc\	func9	126.995us	15.000us	16.170us	15.874us		8.	59.483%	
								1 1	-
		•		III					ti, 4

# Look and Feel (OS)

E: Trace.STATistic.FUNC					
range mo_sieve_40MHz\test\func2d@Cyclic emo_sieve_40MHz\test\func3@Cyclic emo_sieve_40MHz\test\func8@Cyclic emo_sieve_40MHz\test\func20@Cyclic	funcs: 113. total 222.518ms 50.566ms 72.651ms 196.202ms 1.937s	min	max		123408. 123408.
<pre>emo_sieve_40MHz\test\sieve@Cyclic rce\oS_KernReleaseResource@Cyclic etask\0S_KernTerminateTask@Cyclic obal\0S_PAStartTask+0x30@Task_St1 (root)@Task_St1 Hz\demo\OS_TASK_Task_St1@Task_St1 no_sieve_40MHz\test\test@Task_St1 _sieve_40MHz\test\func2@Task_St1 _sieve_40MHz\test\func2@Task_St1</pre>	1.262ms 1.049ms 2.400us	18.65205 10.1570s 8.4780s - 74.9850s 47.3550s 1.5970s 1.7150s 0.9520s	20.14903 10.314us 8.582us 2.400us 1.127ms 75.265us 47.570us 1.603us 1.774us 0.962us	19.414us 10.258us 8.531us 2.400us 1.127ms 74.943us 47.449us 1.599us 1.735us 0.957us	List Last List Max Goto Max Bookmark Max
			0.90203	0.93743	Parents Children Duration Distance here

E::Trace.STAT.LINKage C:0x2498						• 🗙
	led [ Nesting funcs: 1.		tal: 225.10	)3ms		
range \\PA_XPC560XP_os_demo_sieve_40MHz\test\ <b>test</b>			max 2.113us	avr 1.590us	count 141588.	~
		111				*

TRACE32 also provides an interface to third-party timing tools. For details refer to "Trace Export for Third-Party Timing Tools" (app\_timing\_tools.pdf).

This chapter applies for SMP TRACE32 instances.

# **Flat Analysis**

It is recommended to reduce the trace information generated by NEXUS to the required minimum.

- To avoid an overload of the NEXUS port.
- To make best use of the available trace memory.
- To get a more accurate timestamp.

# **Optimum NEXUS Configuration (No OS)**

Flat function run-time analysis does **not** require any **data information** if no OS is used. That's why it is recommended to switch the broadcasting of data information off.

NEXUS.DTM OFF

Your function time chart **can** include task information if you advise NEXUS to export the instruction flow and task switches. For details refer to the chapter **OS-Aware Tracing** of this training.

```
Trace.Chart.sYmbol /TASK "TASKRCV1"
```



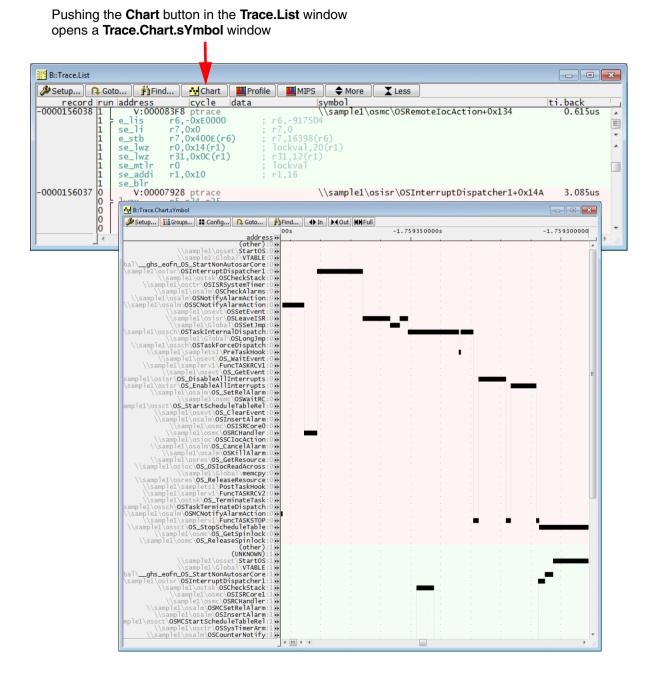
Optimum Configuration 1 (if OSEK generated OTMs):

NEXUS.OTM ON

Optimum Configuration 2 (if OSEK does not support OTMs, NEXUS class 3 only):

```
Break.Set TASK.CONFIG(magic[0]) /Write /TraceData
Break.Set TASK.CONFIG(magic[1]) /Write /TraceData
...
```

TRACE32 PowerView provides a timing diagram which shows when the program counter was in which function/symbol range.



# Trace.Chart.sYmbol [/SplitCore /Sort CoreTogether] Flat function

Flat function run-time analysis

- graphical display
- split the result per core

- sort results per core and then per recording order

B::Trace.Chart.sYmbol /Sort CoreSeparated										- 8	×
🖉 Setup 🚺 Groups 🔡 Config 📭 Goto	Find	<b>♦</b> In	►Out	KNFull							
		2500000	s			-2.912450	0000s				
address (other):04											
(other):1											ĥ
(UNKNOWN):1	E										
\\sample1\osset\StartOS:0											
\\sample1\osset\StartOS:1] \\sample1\Global\VTABLE:0]											
ghs_eofn_0S_StartNonAutosarCore:00									· · '=		
mple1\osisr\OSInterruptDispatcher1:00											
\\sample1\ostsk\OSCheckStack:0]											
\\sample1\osctr\OSISRSystemTimer:00 \\sample1\osalm\OSCheckAlarms:00											
\sample1\osalm\OSCheckAlarms:00 \sample1\osalm\OSNotifyAlarmAction:00											
ample1\osalm\OSSCNotifyAlarmAction:0											
\\sample1\osevt\OSSetEvent:0											
\\sample1\osisr\OSLeaveISR:08											
\\sample1\Global\OSSetJmp:00 mple1\ossch\OSTaskInternalDispatch:00											
<pre>mplel\ossch\OSTaskInternalDispatch:03</pre>											E
\sample1\ossch\OSTaskForceDispatch:08											
\\sample1\samplets1\PreTaskHook:0	E .										
\\sample1\osevt\OS_WaitEvent:0											
\\sample1\samplerv1\FuncTASKRCV1:00 \\sample1\osevt\OS_GetEvent:00											
<pre>\sample1\osevt\OS_GetEvent:0 ple1\osisr\OS_DisableAllInterrupts:00</pre>											
mple1\osisr\OS_EnableAllInterrupts:00											
\\sample1\osalm\OS_SetRelAlarm:0											
\\samp]e1\osmc\OSWaitRC:08											
\\sample1\Global\VTABLE:1											
ghs_eofn_OS_StartNonAutosarCore:13 mple1\osisr\OSInterruptDispatcher1:13											
\\sample1\ostsk\0SCheckStack:1											
\\sample1\osmc\OSISRCore1:1											
\\sample1\osmc\OSRCHandler:1											
\\sample1\osalm\OSMCSetRelAlarm:1	E .										
le1\ossct\05_StartScheduleTableRel:0											
e1\ossct\OSMCStartScheduleTableRel:10											
\\sample1\osctr\OSSysTimerArm:1											
\\sample1\osalm\OSCounterNotify:10											
<pre>mple1\ossct\OSProcessScheduleTable:13 \sample1\osslm\OSNotifyAlarmAction:13</pre>											
ample1\osalm\OSSCNotifyAlarmAction:1											
\\sample1\osisr\OSLeaveISR:1		-									
\\sample1\osevt\OS_ClearEvent:0	E										
\\sample1\Global\OSSetJmp:10											
mple1\ossch\OSTaskInternalDispatch:10											
\\sample1\samplets1\PreTaskHook:1] \\sample1\samplets1\FuncTASKSND1:1				. •							
mple1\osioc\OS_OSIocWriteAcrossRef:1											
\sample1\osalm\OSInsertAlarm:0											
\\sample1\Global\memcpy:1	B I										
\\sample1\osioc\OSIocAction:1								. =			
\\sample1\osmc\OSRemoteIocAction:1											
\\sampier\Osmc\OSwattkc:18	► • Ⅲ ► •				<u> </u>						
]											·

## Trace.Chart.sYmbol [/SplitCore] /Sort CoreSeparated

#### Flat function run-time analysis

- graphical display
- split the result per core
- sort the results per recording order

Setup 🚻 Groups 🔡 Config 📭 Goto	Find	<b>♦</b> In	►Out	KNFull					
	- 2	.91240	00000s		-	2.912350	000s		
address 💀		1				1			
(other)									
(UNKNOWN)									
\\sample1\osset\StartOS									
\\sample1\Global\VTABLE []									
ghs_eofn_OS_StartNonAutosarCore									
nple1\osisr\OSInterruptDispatcher1									
\\sample1\ostsk\05CheckStack									
\\sample1\osctr\OSISRSystemTimer									
\\sample1\osalm\OSCheckAlarms									
\sample1\osalm\OSNotifyAlarmAction@									
ample1\osalm\OSSCNotifyAlarmAction									
\\sample1\osevt\OSSetEvent								· · _	
\\sample1\osisr\OSLeaveISR									_
\\sample1\Global\OSSetJmp								 	
mple1\ossch\OSTaskInternalDispatch									
\\sample1\Global\OSLongJmp									
\sample1\ossch\OSTaskForceDispatch								_ · ·	
\\sample1\samplets1\PreTaskHook								I	
\\sample1\osevt\OS_WaitEvent									
\\samplel\osevt\OS_GetEvent@ ole1\osisr\OS_DisableAllInterrupts@									
mple1\osisr\OS_EnableAllInterrupts									
	< III > <								

### Trace.Chart.sYmbol /MergeCore

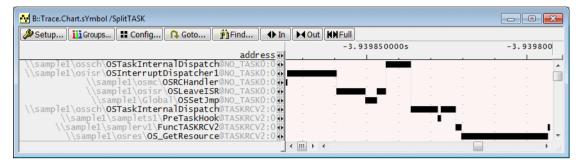
Flat function run-time analysis - graphical display

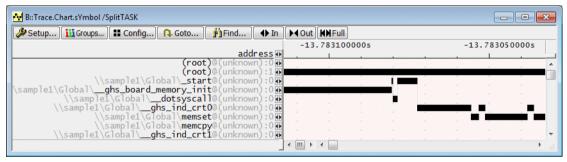
- merge the results of all cores

Default setting

Trace.Chart.sYmbol [/MergeTASK] [/SplitCore /Sort CoreTogether]

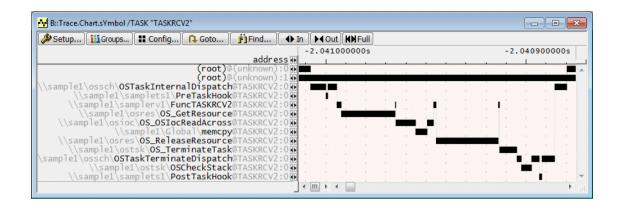
Display function time chart with task information Trace.Chart.sYmbol /SplitTASK [/SplitCore /Sort CoreTogether]



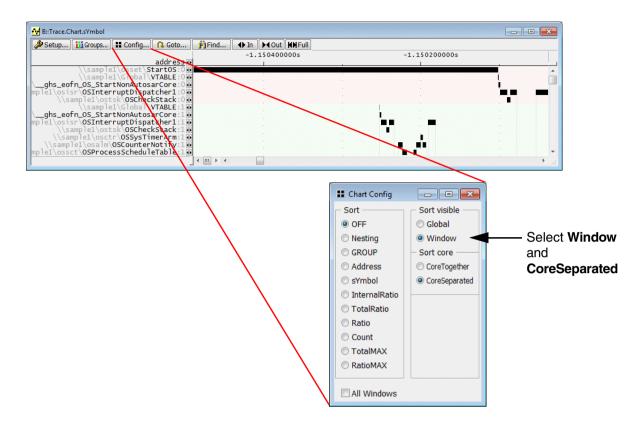


@ <task_name></task_name>	Task name information
@(unknown)	<ul> <li>Function was running before the OS was started</li> <li>Function was recorded before first task switch information was recorded</li> </ul>
(root)@(unknown)	No trace information available

Display function time chart for the specified task Trace.Chart.sYmbol /TASK <task\_name> [/SplitCore /Sort CoreTogether]



@ <task_name></task_name>	Functions running while the specified task was running
(root)@(unknown)	All other trace information



If **Window** and **CoreSeparated** is selected in the **Chart Config** window, the functions that are active at the selected point of time are visualized in the scope of the **Trace.Chart.sYmbol** window. This is helpful especially if you scroll horizontally.

For a detailed description of all **Sort** options provided by the **Chart Config** window refer to the command description of **Trace.STATistic.Sort**.

Analog to the timing diagram also a numerical analysis is provided.

🖉 Setup 🚺 Groups 🔠 Config 🔃 🗛 Goto 📰 Detailed 📑	Tree C	hart 🛛 📕 Prof	file					
	items: 90.		tal: 1.25	6s sample	s: 607808.			
address					count	ratio% 1%	2%	
\\sample1\osalm\ <b>OSKillAlarm</b> :0	384.695us	4.805us	4.810us	4.809us	80.	0.030% +		
<pre>\\sample1\osres\OS_GetResource:0</pre>	3.279ms	25.150us	27.015us	25.820us	127.	0.261%		
<pre>\\sample1\osioc\OS_OSIocReadAcross:0</pre>	1.619ms	2.710us	9.870us	12.745us	127.	0.128% +		
\\sample1\Global\memcpy:0	663.230us	4.930us	5.675us	5.222us	127.	0.052% +		
<pre>\\sample1\osres\OS_ReleaseResource:0</pre>	3.947ms	31.065us	31.085us	31.077us	127.	0.314% +		
<pre>\\sample1\samplets1\PostTaskHook:0</pre>	122.505us	0.860us	1.115us	0.935us	131.	0.009% +		
\\samp]e1\samp]erv1\FuncTASKRCV2:0	196.935us	0.490us	1.975us	4.190us	47.	0.015% +		
\\sample1\ostsk\ <b>OS_TerminateTask</b> :0	458.640us	8.875us	9.740us	8.993us	51. 51.	0.036% +		
\\sample1\ossch\OSTaskTerminateDispatch:0	641.470us	2.215us	6.415us	12.578us	51.	0.051% +		
\\sample1\osalm\OSMCNotifyAlarmAction:0	17.635us	4.065us	4.565us	4.409us	4.	0.001% +		
\\sample1\samplerv1\FuncTASKSTOP:0	18.380us	0.370us	1.850us	4.595us	4.	0.001% +		
\\sample1\ossct\OS_StopScheduleTable:0	121.585us	12.330us	18.375us	30.396us	4.	0.009% +		
(other):1	0.000us	0.000us		0.000us	0.	0.000%		
(UNKNOWN):1	0.620us	0.620us	0.620us	0.620us	1.(1/0)	<0.001% +		
<pre>\\sample1\osset\StartOS:1</pre>	1.204s	12.455us	226.588ms	1.204s	- (1.10)	95.840%		-
\\sample1\Global\VTABLE:1	92.830us	0.365us	0.740us	92.830us	1. (1/0)	0.007% +		
ple1\Global\ghs_eofn_OS_StartNonAutosarCore:1	1.019ms	1.355us	3.820us	4.109us	248.	0.081% +		
\\sample1\osisr\OSInterruptDispatcher1:1	6.563ms	2.220us	17.140us	26.464us	248.	0.522% +		
<pre>\\sample1\ostsk\OSCheckStack:1</pre>	1.883ms	2.835us	5.920us	3.858us	488.	0.149% +		
\\sample1\osmc\OSISRCore1:1	22.080us	1.725us	2.100us	1.840us	12.	0.001% +		
\\sample1\osmc\OSRCHandler:1	177.450us	4.560us	10.360us	14.788us	12.	0.014% +		
\\sample1\osalm\OSMCSetRelAlarm:1	68.805us	1.600us	15.660us	17.201us	4.	0.005% +		
\\sample1\osalm\OSInsertAlarm:1	90.660us	1.480us	9.500us	11.333us	8.	0.007%		
\\sample1\ossct\OSMCStartScheduleTableRel:1	74.730us	2.215us	16.405us	18.683us	4.	0.005% +		
\\sample1\osctr\OSSysTimerArm:1	562.090us	2.215us	3.330us	2.304us	244.	0.044%		
\\sample1\osalm\OSCounterNotify:1 \\sample1\ossct\OSProcessScheduleTable:1	3.877ms	4.560us	9.005us	16.153us	240.	0.308%		
<pre>\\Samprel\OSSCL\OSProcessSchedurerabre:1</pre>	2.251ms	3.080us	8.510us	9.379us	240.	0.179%		

survey	
item	number of recorded functions/symbol regions
total	time period recorded by the trace
samples	total number of recorded changes of functions/symbol regions (instruction flow continuously in the address range of a function/symbol region)

B::Trace.STatistic.sYmbol								X
🌽 Setup 👖 Groups 🔛 Config 🕞 Goto 📑 Detailed	Tree MC	Chart 🛛 🛄 Prot	file					
	items: 90.	to	tal: 1.25	6s sample	es: 607808.			
address				avr	count	ratio% 1%	2%	_
\\sample1\osalm\OSKillAlarm:0		4.805us	4.810us	4.809us	80.	0.030% +		
<pre>\\sample1\osres\OS_GetResource:0</pre>	3.279ms	25.150us	27.015us	25.820us	127.	0.261% +		
<pre>\\sample1\osioc\OS_OSIocReadAcross:0</pre>		2.710us	9.870us	12.745us	127.	0.128% +		
\\sample1\Global\memcpy:0	663.230us	4.930us	5.675us	5.222us	127.	0.052% +		
<pre>\\sample1\osres\OS_ReleaseResource:0</pre>		31.065us	31.085us	31.077us	127.	0.314% +		
\\sample1\samplets1\PostTaskHook:0	122.505us	0.860us	1.115us	0.935us	131.	0.009% +		
\\sample1\samplerv1\FuncTASKRCV2:0	196.935us	0.490us	1.975us	4.190us	47.	0.015% +		
\\sample1\ostsk\OS_TerminateTask:0	458.640us 641.470us	8.875us	9.740us	8.993us	51. 51.	0.036% +		
\\sample1\ossch\OSTaskTerminateDispatch:0	17.635us	2.215us	6.415us 4.565us	12.578us		0.051% ← 0.001% ←		
<pre>\\sample1\osalm\OSMCNotifyAlarmAction:0</pre>	17.655us 18.380us	4.065us 0.370us	1.850us	4.409us 4.595us	4.	0.001%		
\\sample1\ossct\OS_StopScheduleTable:0	121.585us	12.330us	18.375us	30.396us	4.	0.009%		
(other):1	0.000us	0.000us	10. 37 303	0.000us	0.	0.000%		
(UNKNOWN):1	0.620us	0.620us	0.620us	0.620us	1. (1/0)	<0.001% +		
\\sample1\osset\StartOS:1	1.204s	12.455us	226.588ms	1.204s	1.(1/0)	95.840%		=
\\sample1\Global\VTABLE:1	92.830us	0.365us	0.740us	92.830us	1. (1/0)	0.007% +		
ample1\Global\ghs_eofn_05_StartNonAutosarCore:1	1.019ms	1.355us	3.820us	4.109us	248.	0.081% +		
\sample1\osisr\OSInterruptDispatcher1:1	6.563ms	2.220us	17.140us	26.464us	248.	0.522% +		
\\sample1\ostsk\0SCheckStack:1	1.883ms	2.835us	5.920us	3.858us	488.	0.149%		
\\samplel\osmc\OSISRCore1:1	22.080us	1.725us	2.100us	1.840us	12.	0.001% +		
\\sample1\osmc\OSRCHandler:1	177.450us	4.560us	10.360us	14.788us	12.	0.014% +		
\\sample1\osalm\OSMCSetRelAlarm:1	68.805us	1.600us	15.660us	17.201us	4.	0.005% +		
\\sample1\osalm\OSInsertAlarm:1	90.660us	1.480us	9.500us	11.333us	8.	0.007% +		
\\sample1\ossct\OSMCStartScheduleTableRel:1	74.730us	2.215us	16.405us	18.683us	4.	0.005% +		
\\sample1\osctr\OSSysTimerArm:1	562.090us	2.215us	3.330us	2.304us	244.	0.044% +		
\\sample1\osalm\OSCounterNotify:1	3.877ms	4.560us	9.005us	16.153us	240.	0.308% +		
\\sample1\ossct\OSProcessScheduleTable:1	2.251ms	3.080us	8.510us	9.379us	240.	0.179% +		Ψ.
	•		III				Þ	

function details	
address	function/symbol region name
	(other) program sections that can not be assigned to a function/symbol region (UNKNOWN) program sections that can not be decoded
total	time period in the function/symbol region during the recorded time period
min	shortest time continuously in the address range of the function/symbol region
max	longest time continuously in the address range of the function/symbol region
avr	average time continuously in the address range of the function/symbol region (calculated as total/count)
count	number of new entries (start address executed) into the address range of the function/symbol region
ratio	ratio of time in the function/symbol region with regards to the total time period recorded

Trace.STATistic.sYmbol /MergeCORE	Flat function run-time analysis - numerical display - merge the results of all cores
Trace.STATistic.sYmbol /Sort CoreSeparated	Flat function run-time analysis - numerical display - split the result per core - sort the results per recording order
Trace.STATistic.sYmbol [/MergeTASK]	Flat function run-time analysis (OS) - numerical display - no task information
Trace.STATistic.sYmbol /SplitTASK	Flat function run-time analysis (OS) - numerical display including task information
Trace.STATistic.sYmbol /TASK <task_name></task_name>	Flat function run-time analysis (OS) - numerical display for specified task

# Restrictions

- 1. The nesting analysis analyses only high-level language functions.
- 2. The nested function run-time analysis expects common ways to enter/exit functions.
- 3. The nesting analysis is sensitive with regards to FIFOFULLs.

TRACE32 PowerView builds up a separate call tree for each task.

Trace.STATistic.TREE /TASK "TASKRCV1"

	Detailed 🗱 Nesting 🖼 Chart						
	funcs: 25. total: 22.0	530ms					
							Lt or land
range	tree ⊟ (root)	total 22.630ms	min	max 22.630ms	22,630ms	count	intern% 1% 0.059% ←
\\sample1\samplets1\PreTaskHook	PreTaskHook	0.865us	0.865us	0.865us	0.865us	-	0.003%
\\sample1\samplerv1\FuncTASKRCV1	FuncTASKRCV1	22.615ms	0.00505	22.615ms	22.615ms	1. (0/1)	3.036%
sample1\osisr\OS_DisableAllInterrupts	OS_DisableAllInterrupts	40.210us	8.015us	8.140us	8.042us	5.(0/1)	0.177% +
sample1\osisr\05_EnableAllInterrupts	— OS_EnableAllInterrupts	40.570us	8.015us	8.140us	8.114us	5.	0.179%
\\sample1\osalm\OS_SetRelAlarm	— S SetRelAlarm	3.902ms	37.360us	100.395us	41.074us	95.	13.205%
\\sample1\osmc\OSWaitRC	- OSWaitRC	341.095us	68.180us	68.325us	68,219us	5.	1.507%
\\sample1\osalm\OSInsertAlarm	- OSInsertAlarm	572.445us	6.285us	7.650us	6.360us	90.	2.529%
mple1\ossct\OS_StartScheduleTableRel	—⊟ OS_StartScheduleTableRel	816.825us	163.165us	163.880us	163.365us	5.	0.716% +
\\sample1\osmc\OSWaitRC	- OSWaitRC	654.795us	130.940us	130.985us	130.959us	5.	2.893%
\\sample1\osevt\05_ClearEvent	— 05_ClearEvent	1.159ms	12.700us	14.800us	12.882us	90.	5.123%
\\sample1\osevt\OS_WaitEvent	— 🗉 OS_WaitEvent	5.654ms	24.165us	65.620us	62.824us	90.(0/1)	7.003%
\\sample1\ossch\OSTaskForceDispatch	□ □ OSTaskForceDispatch	4.069ms	45.750us	48.345us	47.873us	85.(0/1)	7.865%
\\sample1\ostsk\OSCheckStack	- 0SCheckStack	455.520us	5.300us	5.430us	5.359us	85.	2.012%
\\sample1\samplets1\ <b>PostTaskHook</b>	— PostTaskHook	73.450us	0.860us	0.865us	0.864us	85.	0.324% 🗲
sample1\ossch\OSTaskInternalDispatch	— B OSTaskInternalDispat	1.760ms	19.235us	20.970us	20.709us	85.(0/1)	7.146%
\\sample1\samplets1\PreTaskHook	PreTaskHook	143.020us	1.230us	1.730us	1.703us	84.	0.632% 🗲
\\sample1\osevt\0S_GetEvent	— 0S_GetEvent	1.514ms	15.415us	18.250us	17.010us	89.	6.689%
\\sample1\osalm\OS_CancelA]arm	—⊞ OS_CancelAlarm	2.490ms	29.095us	31.570us	29.291us	85.	9.195%
\\sample1\osalm\OSKillAlarm	└── OSKillAlarm	408.775us	4.805us	4.815us	4.809us	85.	1.806%
\\samplel\osres\ <b>OS_GetResource</b>	— OS_GetResource	2.138ms	25.150us	25.165us	25.158us	85.	9.449%
\\sample1\osioc\OS_OSIocReadAcross	<ul> <li>— 0S_0SIocReadAcross</li> </ul>	1.488ms	17.505us	17.515us	17.511us	85.	6.577%
\\sample1\osres\OS_ReleaseResource	— 05_ReleaseResource	2.642ms	31.070us	31.085us	31.078us	85.	11.673%
<pre>\\sample1\osmc\OS_GetSpinlock \\sample1\osmc\OS_ReleaseSpinlock</pre>	OS_GetSpinlock	22.690us 20.225us	22.690us 20.225us	22.690us 20.225us	22.690us 20.225us	1.	0.100% + 0.089% +

In order to hook a function entry/exit into the correct call tree, TRACE32 PowerView needs to know which task was running when the entry/exit occurred.

The standard way to get information on the current task is to advise the NEXUS to export the instruction flow and task switches. For details refer to the chapter **OS-Aware Tracing** of this training.

Optimum Configuration 1 (if OSEK generated OTMs):

```
NEXUS.OTM ON
Trace.STATistic.InterruptIsFunction ON
```

Optimum Configuration 2 (if OSEK does not support OTMs, NEXUS class 3 only):

```
Break.Set TASK.CONFIG(magic[0]) /Write /TraceData
Break.Set TASK.CONFIG(magic[1]) /Write /TraceData
...
Trace.STATistic.InterruptIsFunction ON
```

# Numerical Nested Function Run-time Analysis for all Software

Trace.STATistic.Func		ted function meric disp		e analysis	i	
F Cov MPC5XXX ORTI_AUTOS						
Perf Configuration						
Perf List						
Perf List Dynamic						
Function Runtime						
Distribution 🕨 📰 Show Numerica	1					
Duration A to B						
Distance trace records F Show Detailed 1	ree					
The Device Show as Timing						
Task Runtime						
Task Services						
Task Function Runtime						
Task Status						
Reset						
B::Trace.STATistic.FUNC						
-	二 Datailad [詞]					
Setup 👖 Groups 🚦 Config 📭 Goto	EDetailed El		otal: 1.37	8s intr:	52.171ms	
					22.1/102	
	e total			avr	count	intern% 1%
ranc (root)@(unknown): (root)@(unknown):	e total 0 5.542ms		max 5.542ms 1.378s	avr 5.542ms 1.378s	count - -	intern% 1% 0.112% ← 50.000%
(root)@(unknown): (root)@(unknown): \\samp]el\samp]erv1\main@(unknown):	le total 0 5.542ms 1 1.378s 0 2.436ms	min - - -	5.542ms 1.378s 2.436ms	5.542ms 1.378s 2.436ms	 1. (0/1)	0.112% + 50.000% <0.001% +
(root)@(unknown): (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartCore@(unknown):	e total 0 5.542ms 1 1.378s 0 2.436ms 0 9.865us		5.542ms 1.378s 2.436ms 9.865us	5.542ms 1.378s 2.436ms 9.865us	1. (0/1) 1.	0.112% 50.000% <0.001% <0.001%
(root)@(unknown): (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartOre@(unknown): \\sample1\osset\StartOS@(unknown): L\osset\OSInitApplications@(unknown):	e total 0 5.542ms 1 1.378s 0 2.436ms 0 9.865us 0 2.414ms 0 28.980us	min - 9.865us 28.980us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us	1. (0/1) 1. 1. (0/1) 1.	0.112% + 50.000% + <0.001% + <0.001% + 0.007% + 0.001% +
(root)@(unknown) (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartCore@(unknown): \\sample1\osset\StartOS@(unknown): 1\osset\OSInitApplications@(unknown): ole1\osisr\OSInitalizeISR@(unknown):	le total 0 5.542ms 1 1.378s 0 2.436ms 0 9.865us 0 2.414ms 0 28.980us 0 2.014ms	min - 9.865us - 28.980us 2.014ms	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms	1. (0/1) 1. (0/1) 1. (0/1) 1.	0.112% + 50.000% + <0.001% + 0.007% + 0.001% + 0.073% +
<pre>(root)@(unknown): (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartOr@(unknown): \\sample1\osset\StartOS@(unknown): 1\osset\OSInitApplications@(unknown): 0e1\osisr\OSInitSystemTimer@(unknown): e1\osctr\OSInitSystemTimer@(unknown):</pre>	total 0 5.542ms 1 1.378s 0 2.436ms 0 9.865us 0 2.414ms 0 28.980us 0 2.014ms 0 3.575us	min  9.865us  28.980us 2.014ms 3.575us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us	1. (0/1) 1. 1. (0/1) 1.	0.112% + 50.000% <0.001% + 0.007% + 0.001% + 0.001% + 0.073% + <0.001% +
<pre>(root)@(unknown) (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartCore@(unknown): \\sample1\osset\StartOS@(unknown): 1\osset\OSInitAplications@(unknown): e1\osctr\OSInitSystemTimer@(unknown): uple1\osmc\OSWaitCoreSynch@(unknown): (sample1\ostsk\OSInitTasks@(unknown):</pre>	le total 5.542ms 1.378s 0.2.436ms 0.2.436ms 0.2.645us 0.2.414ms 0.2.624ms 0.2.014ms 0.2.35us 0.68.200us 68.200us	min - - 28.980us 2.014ms 3.575us 3.945us 68.200us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 5.118us 68.200us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1.	0.112% + 50.000% + <0.001% + 0.001% + 0.001% + 0.001% + <0.001% + <0.001% + 0.001% +
<pre>(root)@(unknown): (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartCore@(unknown): \\sample1\osset\StartOS@(unknown): ole1\osisr\OSInitApplications@(unknown): ole1\osisr\OSInitIalizeISR@(unknown): e1\osctr\OSInitSystemTimer@(unknown): sample1\ostsk\OSInitTasks@(unknown): ole1\osres\OSInitResources@(unknown): ole1\osres\OSInitResources@(unknown):</pre>	re total 0 5.542ms 1 1.378s 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us	min - - - 28.980us 2.014ms 3.575us 3.945us 68.200us 20.840us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 20.840us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 5.118us 68.200us 20.840us	- 1. (0/1) 1. (0/1) 1. 1. 2. 1. 1.	0.112% + 50.000% + 0.001% + 0.001% + 0.001% + 0.073% + 0.001% + 0.001% + 0.002% + 0.002% +
<pre>(root)@(unknown) (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartCore@(unknown): \\sample1\osset\StartOS@(unknown): 1\osset\OSInitAplications@(unknown): e1\osctr\OSInitSystemTimer@(unknown): uple1\osmc\OSWaitCoreSynch@(unknown): (sample1\ostsk\OSInitTasks@(unknown):</pre>	re total 5.542ms 1.378s 0.2.436ms 0.2.436ms 0.2.414ms 0.28.980us 0.2.014ms 0.3.575us 0.10.235us 0.68.200us 0.20.840us 0.10.605us	min - - 28.980us 2.014ms 3.575us 3.945us 68.200us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 5.118us 68.200us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1.	0.112% + 50.000% + 0.001% + 0.001% + 0.001% + 0.001% + 0.001% + 0.001% + 0.001% + 0.001% + 0.001% +
<pre>(root)@(unknown):</pre>	re total 0 5.542ms 1 1.378s 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 2.014ms 0 3.575us 0 68.200us 0 68.200us 0 10.605us 0 11.100us 0 3.580us	min - - - - 28.980us 20.14ms 3.575us 3.945us 68.200us 20.840us 10.605us 11.100us 3.580us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 20.840us 10.605us 11.100us 3.580us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 5.118us 68.200us 20.840us 10.605us 11.100us 3.580us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.001% 0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001%
<pre>(root)@(unknown):</pre>	re total 0 5.542ms 1 1.378s 0 9.865us 0 2.414ms 0 28.980us 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us 0 10.605us 0 11.100us 0 3.580us 0 2.590us	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 8. 980us 2. 014ms 5. 118us 68. 200us 20. 840us 10. 605us 11. 100us 3. 580us 2. 590us	- 1. (0/1) 1. (0/1) 1. 1. 2. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% + 50.000% + 0.001% +
<pre>(root)@(unknown) (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartOre@(unknown): \\sample1\osset\StartOS@(unknown): ple1\osset\OSInitalizeISR@(unknown): ple1\osset\OSInitSystemTimer@(unknown): sample1\ostsk\OSInitTasks@(unknown): oleft\osres\OSInitResources@(unknown): mple1\osset\OSInitResources@(unknown): sample1\ossalm\OSInitAlarms@(unknown): ssample1\ossalm\OSInitAlarms@(unknown): ssample1\ossmc\OSInitSpinlock@(unknown): ssample1\ossmc\OSInitSpinlock@(unknown): (\sample1\ossmc\OSInitSpinlock@(unknown): mple1\ossmc\OSInitSpinlock@(unknown): (\sample1\ossmc\OSInitTasks@(unknown): (\sample1\ossmc\OSInitSpinlock@(unknown): (\sample1\ossmc\OSInitSpinlock@(unknown): (\sample1\ossmc\OSInitSpinlock@(unknown): (\sample1\ossmc\OSInitIOC@(unknown): (\sampl</pre>	le total 0 5.542ms 1 1.378s 0 2.436ms 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 10.235us 0 68.200us 0 20.840us 0 10.605us 0 11.100us 0 2.590us 0 11.350us	min - - - - 28.980us 20.14ms 3.575us 3.945us 68.200us 20.840us 10.605us 11.100us 3.580us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 5.118us 68.200us 20.840us 10.605us 11.100us 3.580us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.001% 0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001%
<pre>(root)@(unknown):</pre>	le total 5.542ms 1.3785 0.2.436ms 0.2.436ms 0.2.414ms 0.2.6980us 0.2.014ms 0.2.014ms 0.10.255us 0.10.255us 0.10.605us 0.11.100us 0.3.580us 0.2.590us 0.11.350us 0.865us 0.3.085us 0.3.085us	min - - - 28.980us 2.014ms 3.575us 3.945us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 2.590us 11.350us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 68.200us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 8. 980us 2. 014ms 5. 118us 68. 200us 5. 118us 68. 200us 11. 100us 3. 580us 1. 100us 3. 580us 0. 605us 11. 350us 0. 865us 3. 085us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% + 50.000% + 0.001%
<pre>(root)@(unknown) (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartOre@(unknown): \\sample1\osset\StartOS@(unknown): ple1\osset\OSInitalizeISR@(unknown): ple1\osset\OSInitalizeISR@(unknown): sample1\ostsk\OSInitTasks@(unknown): olefl\osres\OSInitResources@(unknown): sample1\ostsk\OSInitTasks@(unknown): sample1\ossath\OSInitAlarms@(unknown): sample1\ossath\OSInitAlarms@(unknown): sample1\ossath\OSInitAlarms@(unknown): sample1\ossath\OSInitAlarms@(unknown): (\sample1\ossic\OSInitIoUcd@(unknown): (\sample1\ossic\OSInitIOUC@(unknown): (\sample1\ossic\OSInitIOUC@(unknown): (\osctr\OSStartSystemTimer@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\ossch\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(unknown): (\osctr\OSTaskForceDispatch@(Unknown):</pre>	le total 0 5.542ms 1 1.378s 0 2.436ms 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us 0 10.605us 0 11.100us 0 3.580us 0 2.590us 0 11.350us 0 3.085us 0 21.830us	min - - - 28.980us 2.014ms 3.575us 3.945us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 62.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us 21.830us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 5.118us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 2.590us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.007% 0.007% 0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001%
<pre>(root)@(unknown):</pre>	re total 0 5.542ms 1 1.378s 0 2.436ms 0 2.414ms 0 2.8980us 0 2.014ms 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 68.200us 0 20.840us 0 11.100us 0 3.580us 0 11.350us 0 11.350us 0 2.990us 0 11.350us 0 2.890us 0 2.990us 0 2.990us 0 2.890us 0 3.575us 0 3.	min - - - - - 28.980us 2.980us 2.014ms 3.575us 3.945us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 1.350us 0.865us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 68.200us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 8. 980us 2. 014ms 5. 118us 68. 200us 5. 118us 68. 200us 11. 100us 3. 580us 1. 100us 3. 580us 0. 605us 11. 350us 0. 865us 3. 085us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.0
<pre>(root)@(unknown) (root)@(unknown): \\sample1\samplerv1\main@(unknown): \\sample1\osmc\StartOre@(unknown): \\sample1\osset\StartOS@(unknown): Disset\OSInitaplications@(unknown): Diel\osstr\OSInitializeISR@(unknown): Diel\osstr\OSIniticOurtsreamer@(unknown): Sample1\ostsk\OSInitTasks@(unknown): Diel\osres\OSInitResources@(unknown): Diel\osres\OSInitResources@(unknown): Diel\osres\OSInitAlarms@(unknown): Diel\osstr\OSInitAlarms@(unknown): Diel\ossctr\OSInitAlarms@(unknown): Diel\osmc\OSInitAlarms@(unknown): Diel\osmc\OSInitSpinlock@(unknown): Dissct\OSInitScheduleTables@(unknown): Dissmple1\ossic\OSInitIOC_Init@(unknown): Dissmple1\osstr\OSInitSpinlock@(unknown): Dissmple1\osstr\OSInitSpinlock@(unknown): Dissch\OSTaskForceDispatch@(unknown): Sample1\ostsk\OSCheckStack@(unknown): Sch\OSTaskInternalDispatch@(unknown): Croot)@TASKRC\</pre>	le total 0 5.542ms 1 1.378s 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 28.980us 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us 0 11.050us 0 11.150us 0 3.580us 0 3.685us 0 3.085us 0 3.675us 0 2.590us 0 11.350us 0 3.685us 0 3.685us 0 3.685us 0 3.675us 0 2.630ms 0 2.500ms 0 2.5	min - - - 28.980us 2.014ms 3.575us 3.945us 68.20us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us - 3.575us -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us 3.085us 21.830us 3.575us 6.170us 22.630ms	5.542ms 1.378s 2.436ms 9.865us 2.414ms 3.575us 5.118us 68.200us 20.840us 20.840us 11.100us 3.580us 0.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us 2.590us 2.590us 2.590us 2.590us 2.590us 2.590us 2.590us 2.590us 2.630us 2.630ms	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.007% 0.007% 0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001%
<pre>(root)@(unknown):</pre>	re total 0 5.542ms 1 1.378s 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 28.980us 0 2.014ms 0 3.575us 0 68.200us 0 68.200us 0 10.235us 0 68.200us 0 11.100us 0 3.580us 0 11.350us 0 11.350us 0 21.830us 0 3.685us 0 3.675us 0 3.675us 1 43.885us 1 43.885us	min - - - 28.980us 2.014ms 3.575us 3.945us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 3.085us 3.085us 3.575us 6.170us 22.630ms 1.730us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 4. 200us 5. 118us 68. 200us 11. 100us 2. 590us 11. 350us 3. 885us 21. 830us 3. 575us 6. 170us 22. 630ms 1. 693us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% 40.001%
<pre>(root)@(unknown):</pre>	le total 5.542ms 1.3785 0.2.436ms 0.2.436ms 0.2.436ms 0.2.414ms 0.28.980us 0.2.014ms 0.2014ms 0.2014ms 0.255us 0.10.605us 0.11.100us 0.2.590us 0.11.350us 0.2.890us 0.11.350us 0.2.990us 0.11.350us 0.2.990us 0.2.990us 0.2.990us 0.2.990us 0.2.990us 0.2.990us 0.2.990us 0.2.990us 0.2.850us 0.2.850us 0.2.630ms 1.43.885us 1.43.885us 1.22.615ms	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 68.200us 68.200us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us 21.830us 3.575us 6.170us 22.630ms 1.730us 22.615ms	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 8. 980us 2. 014ms 5. 118us 68. 200us 5. 118us 68. 200us 11. 100us 3. 580us 11. 350us 0. 605us 11. 350us 0. 865us 3. 085us 21. 830us 3. 575us 6. 170us 22. 630ms 22. 630ms 22. 6315ms	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% + 50.000% + 0.001% + 0.002%
<pre>(root)@(unknown):</pre>	re total 0 5.542ms 1 1.378s 0 2.436ms 0 2.414ms 0 2.8980us 0 2.014ms 0 2.8980us 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us 0 11.100us 0 3.580us 0 11.350us 0 11.350us 0 11.350us 0 2.630ms 143.885us 143.885us 143.885us 140.210us 140.570us	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 3.085us 3.085us 3.085us 3.75us 6.170us 22.615ms 8.140us 8.140us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 4. 200us 5. 118us 68.200us 11. 100us 2. 600us 11. 100us 2. 590us 11. 350us 6. 170us 2. 630ms 6. 170us 2. 630ms 8. 042us 8. 114us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.0
<pre>(root)@(unknown):</pre>	le total 0 5.542ms 1 1.378s 0 2.436ms 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 28.980us 0 10.235us 0 10.235us 0 10.605us 0 10.605us 0 11.100us 0 3.580us 0 10.605us 0 11.350us 0 2.590us 0 11.350us 0 2.630ms 143.885us 0 21.830us 0 3.575us 0 3.085us 0 3.655us 0 1.120us 0 2.630ms 143.885us 144.885us 143.885us 143.885us 143.885us 143.885us 143.885us 14	min - 9.865us - 28.980us 2.014ms 3.575us 3.945us 68.200us 20.840us 20.840us 10.605us 11.100us 3.580us 3.580us 3.085us - 3.575us - 0.865us - 8.015us 8.015us 37.360us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 68.200us 68.200us 10.605us 11.100us 3.580us 2.590us 11.350us 3.085us 21.830us 3.575us 6.170us 22.630ms 1.730us 22.615ms 8.140us 100.395us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 8.980us 2.014ms 5.118us 68.200us 20.840us 11.100us 3.580us 11.350us 0.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us 21.830us 3.085us 21.830us 3.085us 21.630ms 22.630ms 22.630ms 22.630ms 22.630ms 21.693us 22.615ms 8.042us 8.114us 4.174us 4.174us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.0
<pre>(root)@(unknown):</pre>	te total 5.542ms 1 1.378s 0 2.436ms 0 2.436ms 0 2.414ms 0 2.8.980us 0 2.014ms 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us 0 11.050us 0 3.580us 0 11.350us 0 3.685us 0 3.085us 0 3.085us 0 3.085us 0 3.085us 0 3.655us 0 4.22.630ms 143.885us 140.570us 140.570us 1995.890us	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 3.085us 3.085us 3.085us 3.75us 6.170us 22.615ms 8.140us 8.140us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 4. 200us 5. 118us 68. 200us 11. 100us 2. 600us 11. 100us 2. 590us 11. 350us 6. 170us 2. 630ms 6. 170us 2. 630ms 8. 042us 8. 114us	- 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.0
<pre>(root)@(unknown):</pre>	le total 0 5.542ms 1 1.378s 0 2.436ms 0 2.436ms 0 2.414ms 0 2.8980us 0 2.014ms 0 2.014ms 0 2.035us 0 10.235us 0 10.235us 0 10.655us 0 10.655us 0 10.655us 0 11.100us 0 3.580us 0 2.590us 0 11.350us 0 2.590us 0 11.350us 0 2.630ms 1 43.885us 0 21.830us 0 3.75us 1 43.885us 0 2.615ms 1 40.570us 1 40.570us	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 20.840us 20.840us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 21.830us 3.085us 21.830us 22.630ms 1.730us 22.635ms 8.140us 8.140us 130.985us 130.985us 14.800us 14.800us 14.800us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 3.575us 5.118us 68.200us 20.840us 3.580us 1.100us 3.580us 1.350us 0.605us 11.100us 3.580us 3.085us 2.990us 11.350us 0.865us 3.085us 3.042us 8.114us 3.085us 3.1074us 3.365us 1.2882us 3.2882us	$\begin{array}{c} & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & &$	0.112% 50.000% 40.001% 40.0005% 40.0005% 40.00
<pre>(root)@(unknown):</pre>	te total 5.542ms 1 1.378s 0 2.436ms 0 2.436ms 0 2.414ms 0 2.8.980us 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us 0 10.605us 0 11.100us 0 3.580us 0 11.1350us 0 0.865us 0 3.085us 0 3.085us 0 3.085us 0 3.085us 0 3.085us 0 3.085us 0 40.210us 143.885us 1 22.630ms 1 43.885us 1 22.630ms 1 43.885us 1 22.630ms 1 40.570us 1 40.570us 1 3.902ms 1 45.800us 1 1.159ms 1 5.72.445us	min - - - 28.980us 2.014ms 3.575us 3.945us 68.200us 20.840us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us - - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 1.20000 1.20000 1.20000 1.20000 1.20000 1.2	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 28.980us 2.014ms 6.290us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 10.865us 3.085us 21.830us 21.830us 21.830us 21.630ms 1.730us 22.615ms 8.140us 130.985us 130.985us 130.985us 130.985us 140us 10.395us 140us 10.395us 140us 10.395us 140us 10.395us 140us 163.880us 14.800us 7.650us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 9865us 2. 014ms 3. 575us 5. 118us 68. 200us 10. 605us 11. 100us 20. 840us 10. 605us 11. 100us 2. 590us 11. 350us 2. 590us 11. 350us 2. 590us 11. 350us 2. 630ms 2. 630ms 2. 630ms 2. 630ms 2. 615ms 2. 630ms 2. 6176us 2. 630ms 2. 642us 8. 114us 4. 1074us 8. 114us 4. 1074us 8. 114us 4. 1074us 1. 638us 1. 882us 1. 882us 1. 882us 1. 636us 3. 650us 3. 650us 3. 650us 3. 6170us 3. 6170us 3. 6170us 3. 6170us 5. 6170us 5. 6170us 6. 1070us 6. 1070us	- - 1. (0/1) 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.0001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.001% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.002% <0.
<pre>(root)@(unknown):</pre>	total           0         5.542ms           1         1.378s           0         2.436ms           0         2.436ms           0         2.436ms           0         2.645us           0         2.645us           0         2.645us           0         2.640us           0         10.605us           0         10.605us           0         11.100us           0         3.580us           0         11.350us           0         1.350us           0         1.350us           0         2.630ms           1         2.630ms           1         2.2.630ms           1         2.2.615ms           1         40.570us           3.902ms         995.890us           1.159ms         1.159ms           1         5.654ms	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 20.840us 20.840us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 21.830us 3.085us 21.830us 22.630ms 1.730us 22.635ms 8.140us 8.140us 130.985us 130.985us 14.800us 14.800us 14.800us	5.542ms 1.378s 2.436ms 9.865us 2.414ms 3.575us 5.118us 68.200us 20.840us 3.580us 1.100us 3.580us 1.350us 0.605us 11.100us 3.580us 3.085us 2.990us 11.350us 0.865us 3.085us 3.042us 8.114us 3.085us 3.1074us 3.365us 1.2882us 3.2882us	- - 1. (0/1) 1. (0/1) 1. 1. (0/1) 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	0.112% 50.000% 40.001% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\% 40.005\%
<pre>(root)@(unknown):</pre>	le total 0 5.542ms 1 1.378s 0 2.436ms 0 2.436ms 0 2.414ms 0 28.980us 0 2.014ms 0 3.575us 0 10.235us 0 68.200us 0 20.840us 0 10.605us 0 11.1350us 0 3.580us 0 11.350us 0 3.685us 0 3.085us 0 3.085us 0 3.085us 0 3.085us 0 40.210us 143.885us 1 43.885us 1 22.630ms 1 43.885us 1 22.630ms 1 43.885us 1 40.270us 1 40.570us 1 40.570us 1 40.570us 1 572.445us 5 .654ms 0 0.00us 0 52.171ms	min - - - - 28.980us 2.014ms 3.575us 3.945us 68.200us 20.840us 20.840us 11.100us 3.580us 2.590us 11.100us 3.580us 2.590us 0.865us 3.085us - - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us 3.575us - 0.865us - 0.865us - 0.865us - - 0.865us - - 0.865us - - 0.865us - - 0.865us - - 0.865us - - - 0.865us - - - 0.865us - - - 0.865us - - - - 0.865us - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 20.840us 10.605us 11.100us 2.590us 11.350us 2.590us 11.350us 2.590us 11.350us 2.630ms 1.730us 22.615ms 8.140us 10.395us 163.880us 14.800us 8.140us 10.395us 163.880us 14.800us 165.080us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 014ms 2. 014ms 3. 575us 5. 118us 68. 200us 10. 605us 11. 100us 2. 590us 11. 350us 3. 580us 2. 590us 11. 350us 3. 685us 3. 0.865us 3. 0.865us 3. 0.85us 2. 630ms 2. 630ms 2. 630ms 2. 6170us 2. 630ms 2. 630ms 2. 6170us 2. 630ms 4. 1074us 2. 630ms 2. 615ms 8. 042us 8. 114us 4. 1074us 2. 630ms 2. 636us 1. 636us 2. 636us 6. 365us 1. 882us 6. 365us 1. 882us 6. 365us 2. 636us 2. 636us 6. 365us 1. 636us 6. 365us 2. 824us 0. 000us 5. 855us 5. 8555 5. 8555 5. 11805 5. 11	- - - - - - - - - - - - - -	0.112% 50.000% -0.001% -0.0001% -0.000% -0.
<pre>(root)@(unknown):</pre>	total           0         5.542ms           1         1.378s           0         2.436ms           0         2.436ms           0         2.436ms           0         2.645us           0         2.645us           0         2.640us           0         2.620us           0         10.605us           0         11.100us           0         3.80us           0         10.605us           0         11.100us           0         3.80us           0         11.350us           0         0.86sus           0         11.350us           0         0.86sus           1         13.88us           21.830us         6.170us           1         22.630ms           1         40.85sus           1         2.630ms           1         40.570us           3.902ms         195.880us           1         1.159ms           5.654ms         0.000us           0         3.422ms	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us 21.830us 22.630ms 22.630ms 8.140us 1.730us 22.615ms 8.140us 1.730us 22.615ms 8.140us 1.730us 22.615ms 8.140us 1.650us 1.650us 1.5080us 6.170us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 414ms 2. 014ms 3. 575us 5. 118us 68. 200us 5. 118us 68. 200us 10. 605us 11. 100us 3. 580us 2. 605us 11. 350us 6. 855us 3. 085us 21. 630ms 22. 630ms 22. 630ms 22. 630ms 22. 635us 21. 639us 21. 639us 21. 639us 22. 615ms 8. 014us 1. 4074us 99. 589us 21. 6365us 21. 882us 6. 365us 22. 882us 6. 365us 22. 882us 6. 365us 22. 882us 6. 365us 22. 882us 59. 285us 3. 888us	- - - - - - - - - - - - - -	0.112% 50.000% -0.001% -0.005% -0.002% -0.005% -0.002% -0.005% -0.0
<pre>(root)@(unknown):</pre>	le         total           0         5.542ms           1         1.378s           0         2.436ms           0         2.436ms           0         2.436ms           0         2.436ms           0         2.414ms           0         2.620ms           0         2.620ms           1         1.378s           0         2.436ms           0         2.414ms           0         2.800ms           1         1.025us           0         10.235us           0         10.605us           0         1.100us           0         2.590us           1         1.350us           0         2.630ms           1         1.43.885us           2.2.630ms         1.43.885us           2.2.630ms         1.43.885us           3.022ms         995.890us           1.159ms         572.445us           5.654ms         3.422ms           0         3.690ms	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 2.014ms 3.575us 6.290us 68.200us 20.840us 20.840us 20.840us 10.605us 11.100us 3.580us 21.830us 3.085us 21.830us 22.630ms 1.730us 22.630ms 1.730us 22.635ms 8.140us 1.730us 22.635ms 8.140us 1.730us 22.635ms 8.140us 1.730us 22.635ms 8.140us 1.730us 22.635ms 8.140us 1.730us 22.635ms 8.140us 1.730us 22.635ms 8.140us 1.755us 1.7755us 1.7755us 1.7755us 1.7755us 1.77555us 1.775555 1.7755555 1.775555 1.775555 1.775555 1.775555 1.775555 1.775555 1.775555 1.7755555 1.7755555 1.7755555 1.775555555 1.7755555555555555555555555555555555555	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 3. 575us 5. 118us 68. 200us 20. 840us 3. 580us 3. 580us 1. 100us 3. 580us 1. 350us 0. 605us 11. 350us 0. 865us 3. 085us 3. 085us 3. 085us 3. 085us 2. 630ms 1. 693us 2. 630ms 1. 693us 2. 630ms 1. 693us 2. 635us 1. 693us 2. 635us 1. 693us 2. 635us 1. 693us 2. 635us 1. 635us 1. 635us 1. 635us 1. 635us 1. 635us 1. 825us 3. 828us 3. 8	- - - - - - - - - - - - - -	0.112% 50.000% 40.001% 40.000%
<pre>(root)@(unknown):</pre>	total           0         5.542ms           1         1.378s           0         2.436ms           0         2.436ms           0         2.436ms           0         2.645us           0         2.645us           0         2.645us           0         2.640us           0         10.605us           0         10.605us           0         11.100us           0         2.590us           0         1.350us           0         2.630ms           1         1.350us           0         2.630ms           1         4.385us           1         4.385us           1         4.570us           3.902ms         3.902ms           1         3.902ms           1         3.902ms           1         1.564ms           0         0.000us           0         3.407ms           0         3.407ms	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 68.200us 20.840us 10.605us 11.100us 3.580us 2.590us 11.350us 0.865us 3.085us 21.830us 22.630ms 22.630ms 8.140us 10.395us 13.0985us 14.800us 7.650us 14.800us 7.650us 14.800us 7.650us 15.080us 6.170us 28.490us 28.490us 26.635us 10.730us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 414ms 2. 014ms 3. 575us 5. 118us 68. 200us 5. 118us 68. 200us 10. 605us 11. 100us 3. 580us 2. 605us 11. 350us 6. 605us 11. 350us 6. 865us 3. 685us 21. 630ms 22. 630ms 22. 615ms 8. 014us 10. 605us 11. 430us 22. 630ms 21. 639us 21. 639us 22. 615ms 8. 014us 9. 589us 11. 450us 6. 365us 11. 4074us 99. 589us 6. 365us 12. 882us 6. 365us 5. 1288us 6. 365us 5. 284us 0. 000us 5. 285us 3. 888us 26. 544us 24. 508us 9. 586us 25. 654us 26. 544us 26. 548us 26. 5	- - - - - - - - - - - - - -	0.112% 50.000% 40.001% 40.000%
<pre>(root)@(unknown):</pre>	total           0         5.542ms           1         1.378s           0         2.436ms           0         2.436ms           0         2.436ms           0         2.436ms           0         2.436ms           0         2.436ms           0         2.414ms           0         2.860us           0         10.235us           0         10.235us           0         10.605us           0         1.100us           0         2.590us           0         1.1350us           0         2.630ms           1         1.43.885us           2         2.615ms           1         40.570us           3         995.890us           1         1.159ms           5.654ms         5.654ms           1         1.59ms           5.654ms         0.600us           0         0.000us           0         3.402ms           0         3.407ms           0         3.407ms           0         496.865us	min - - - - - - - - - - - - -	5.542ms 1.378s 2.436ms 9.865us 2.414ms 28.980us 2.014ms 3.575us 6.290us 68.200us 20.840us 10.605us 11.100us 2.590us 11.350us 2.590us 11.350us 2.590us 11.350us 2.630ms 1.730us 22.615ms 8.140us 10.395us 163.880us 14.800us 163.880us 14.800us 165.620us 105.080us 6.170us 28.490us 28.490us 26.635us	5. 542ms 1. 378s 2. 436ms 9. 865us 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 414ms 2. 401ms 3. 575us 4. 200us 11. 100us 2. 590us 11. 350us 3. 580us 2. 590us 11. 350us 3. 685us 3. 685us 2. 630ms 2. 630ms 2. 630ms 2. 6170us 2. 630ms 2. 630ms 2. 630ms 2. 6402us 8. 114us 41. 074us 8. 642us 6. 365us 1. 882us 6. 365us 1. 882us 6. 365us 2. 884us 0. 000us 2. 542us 3. 888us 2. 654us 2. 654us 2. 630ms 2. 654us 2.	- - - - - - - - - - - - - -	0.112% 50.000% -0.001% + -0.001% + 0.001% + 0.001% + 0.001% + -0.001% + -0.000%

- Task-specific function run-time analysis, core information is discarded.
- Functions that can not be assigned to a task are assigned to the (@unknown) task, per core display.
- Interrupt service routines are assigned to (@interrupt) task, per core display.

survey	
func	number of functions in the trace
total	total measurement time
intr	total time in interrupt service routines

B::Trace.STATistic.FUNC							x
	Vesting 🛛 🚟 Cł	nart					
	funcs: 83.	to	tal: 1.37	'8s intr:	52.171ms	-	
range	total	min	max	avr	count	intern% 1%	
(root)@TASKRCV1	22.630ms	-	22.630ms	22.630ms	-	<0.001% +	-
\\sample1\samplets1\PreTaskHook@TASKRCV1	143.885us	0.865us	1.730us	1.693us	85.	0.005% +	
\\sample1\samplerv1\FuncTASKRCV1@TASKRCV1	22.615ms	-	22.615ms	22.615ms	1.(0/1)	0.024% +	
\\sample1\osisr\OS_DisableAllInterrupts@TASKRCV1	40.210us	8.015us	8.140us	8.042us	5.	0.001% +	
\\sample1\osisr\OS_EnableAllInterrupts@TASKRCV1	40.570us	8.015us	8.140us	8.114us	5.	0.001% +	
\\sample1\osalm\OS_SetRelAlarm@TASKRCV1	3.902ms	37.360us	100.395us	41.074us	95.	0.108% +	
\\sample1\osmc\OSWaitRC@TASKRCV1	995.890us	68.180us	130.985us	99.589us	10.	0.036% +	
\\sample1\ossct\OS_StartScheduleTableRel@TASKRCV1	816.825us	163.165us	163.880us	163.365us	5.	0.005% +	
\\sample1\osevt\05_ClearEvent@TASKRCV1		12.700us	14.800us	12.882us	90.	0.042% +	
\\sample1\osalm\ <b>OSInsertAlarm</b> @TASKRCV1	572.445us	6.285us	7.650us	6.360us	90.	0.020% +	
\\sample1\osevt\OS_WaitEvent@TASKRCV1	5.654ms	24.165us	65.620us	62.824us	90.(0/1)	0.057% +	•
	•		III				F .

columns	
range (NAME)	function name, sorted by their recording order as default

# • HLL function, task specific

# \\sample1\osmc\OSWaitRC@TASKRCV1

HLL function "OSWaitRC" running in task "TASKRCV1"

## Root of task-specific call tree

# (root)@TASKRCV1

(root) of call tree for task TASKRCV1

B::Trace.STATistic.FUNC							x
🌽 Setup 🚺 Groups ) ☵ Config 📭 Goto ) 🗾 Detailed 📳 Nesting				-			
	funcs: 84.	-	otal: 1.37	'8s intr:	56.229ms stac		-
range	total	min	max	avr	count	intern% 1%	
(root)@(interrupt):0	0.000us	-	-	0.000us	-	0.000%	
→\\sample1\Global\VTABLE+0x40@(interrupt):0	56.229ms	55.595us	109.885us	63.897us	880. (0/1)	0.147%	
\\sample1\osisr\OSInterruptDispatcher1@(interrupt):0	52.163ms	51.035us	105.070us	59.276us	880. (0/1)	0.964%	
\\sample1\ostsk\0SCheckStack@(interrupt):0		3.695us	6.165us	3.887us	880.	0.124%	
\\sample1\osmc\OSISRCoreO@(interrupt):0	3.689ms	24.530us	28,490us	26.540us	139.	0.010% +	
\\sample1\osmc\OSRCHandler@(interrupt):0		22.435us	26.640us	24.505us	139.	0.075% +	
\\sample1\osioc\OSSCIocAction@(interrupt):0	1.294ms	7.890us	10.730us	9.585us	135.	0.029% +	
<pre>\\sample1\osevt\OSSetEvent@(interrupt):0</pre>	496.780us		5.800us	5.582us	89.	0.018% +	-
	•		m			,	►d

Indirect branch into interrupt vector table

→\\sample1\Global\VTABLE+0x40@(interrupt):0

Interrupt service function

\\sample1\osevt\OSSetEvent@(interrupt):0

• Root of @(interrupt)

(root)@(interrupt):0

B::Trace.STATistic.FUNC											×
🌽 Setup 👖 Groups 📰 C	Config 🚺 Got	o 💽 Detaile	ed 💽 Nesting	Chart							
	funcs: 132.	to	tal: 927.75	8ms intr:	53.269ms	stopped:	0.370us				
range	total	min	max	avr	count	inter	n% ∣1%	2%	5%	10%	
	197.879ms	-		197.879ms	-	10.0					
(root)@(unknown):1		-	204.222ms		-	11.0					
(root)@(interrupt):0	0.000us	-	-	0.000us	-	0.0	00%				-
	•										A 1

TRACE32 assigns all trace information generated before the first task switch to the @(unknown) task.

B::Trace.STATistic.FUNC							×
🌽 Setup 👖 Groups 📲 Config 🔃 Goto 📰 Detailed  👔 N							
	funcs: 83.	to	otal: 1.37	8s intr:	52.171ms		
range		min			count	intern% 1%	
(root)@TASKRCV1	22.630ms	-	22.630ms	22.630ms	-	<0.001% +	
\\sample1\samplets1\PreTaskHook@TASKRCV1	143.885us	0.865us	1.730us	1.693us	85.	0.005% +	
\\sample1\samplerv1\FuncTASKRCV1@TASKRCV1	22.615ms	-	22.615ms	22.615ms	1. (0/1)	0.024% +	
\\sample1\osisr\OS_DisableAllInterrupts@TASKRCV1	40.210us	8.015us	8.140us	8.042us	5.	0.001% +	
\\sample1\osisr\OS_EnableAllInterrupts@TASKRCV1	40.570us	8.015us	8.140us	8.114us	5.	0.001% +	
\\sample1\osalm\ <b>OS_SetRelAlarm</b> @TASKRCV1	3.902ms	37.360us	100.395us	41.074us	95.	0.108% +	
\\sample1\osmc\ <b>OSWaitRC</b> @TASKRCV1	995.890us	68.180us	130.985us	99.589us	10.	0.036% +	
\\sample1\ossct\OS_StartScheduleTableRel@TASKRCV1	816.825us	163.165us	163.880us	163.365us	5.	0.005% +	
\\sample1\osevt\ <b>OS_ClearEvent</b> @TASKRCV1	1.159ms	12.700us	14.800us	12.882us	90.	0.042% +	
\\sample1\osalm\ <b>OSInsertAlarm</b> @TASKRCV1	572.445us		7.650us	6.360us	90.	0.020% +	
\\sample1\osevt\OS_WaitEvent@TASKRCV1	5.654ms		65.620us	62.824us	90. (0/1)	0.057% +	-
	•		m				►

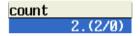
columns (cont.)	
total	total time within the function
min	shortest time between function entry and exit, time spent in interrupt service routines is excluded No <b>min</b> time is displayed if a function exit was never executed.
max	longest time between function entry and exit, time spent in interrupt service routines is excluded
avr	average time between function entry and exit, time spent in interrupt service routines is excluded

E:: Trace.STATistic.FUNC							×
🌽 Setup 🚺 Groups 🔠 Config 📭 Goto 🗾 Detailed			_				
	funcs: 83.	to	tal: 1.37	8s intr:	52.171ms		
range	total	min	max	avr	count	intern% 1%	
(root)@TASKRCV1	22.630ms	-	22.630ms	22.630ms	-	<0.001% +	
\\sample1\samplets1\ <b>PreTaskHook</b> @TASKRCV1	143.885us	0.865us	1.730us	1.693us	85.	0.005% +	
\\sample1\samplerv1\ <b>FuncTASKRCV1</b> @TASKRCV1	22.615ms	-	22.615ms	22.615ms	1. (0/1)	0.024%	
\\sample1\osisr\OS_DisableAllInterrupts@TASKRCV1	40.210us	8.015us	8.140us	8.042us	5.	0.001% +	
\\sample1\osisr\OS_EnableAllInterrupts@TASKRCV1	40.570us	8.015us	8.140us	8.114us	5.	0.001% +	
\\sample1\osalm\ <b>OS_SetRelAlarm</b> @TASKRCV1	3.902ms	37.360us	100.395us	41.074us	95.	0.108% +	
\\sample1\osmc\ <b>OSWaitRC</b> @TASKRCV1	995.890us	68.180us	130.985us	99.589us	10.	0.036% +	
\\sample1\ossct\OS_StartScheduleTableRel@TASKRCV1	816.825us	163.165us	163.880us	163.365us	5.	0.005% +	
\\samp]e1\osevt\ <b>OS_C]earEvent</b> @TASKRCV1	1.159ms	12.700us	14.800us	12.882us	90.	0.042%	
\\samp]e1\osa]m\ <b>OSInsertA]arm</b> @TASKRCV1	572.445us	6.285us	7.650us	6.360us	90.	0.020% +	
\\sample1\osevt\OS_WaitEvent@TASKRCV1	5.654ms	24.165us	65.620us	62.824us	90.(0/1)	0.057% +	-
	•		m				▶

columns (cont.)	
count	number of times within the function

If function entries or exits are missing, this is displayed in the following format:

<times within the function >. (<number of missing function entries>/<number of missing function exits>).



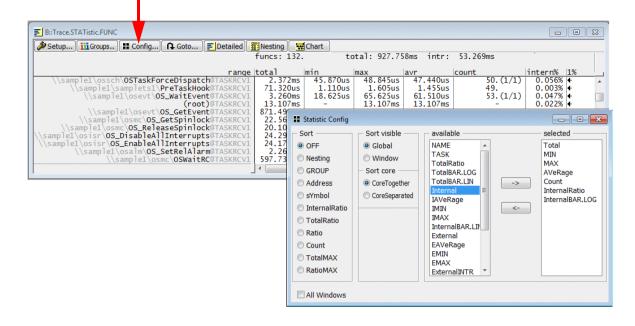
### Interpretation examples:

- 1. 2. (2/0): 2 times within the function, 2 function entries missing
- 2. 4. (0/3): 4 times within the function, 3 function exits missing
- 3. 11. (1/1): 11 times within the function, 1 function entry and 1 function exit is missing.

If the number of missing function entries or exits is higher the 1 the performed by the command <b>Trace.STATistic.Func</b> might fail due to problems. A detailed view to the trace contents is recommended.	
---	--

columns (cont.)	
intern% (InternalRatio, InternalBAR.LOG)	ratio of time within the function without subfunctions and interrupts

## Pushing the Config... button allows to display additional columns

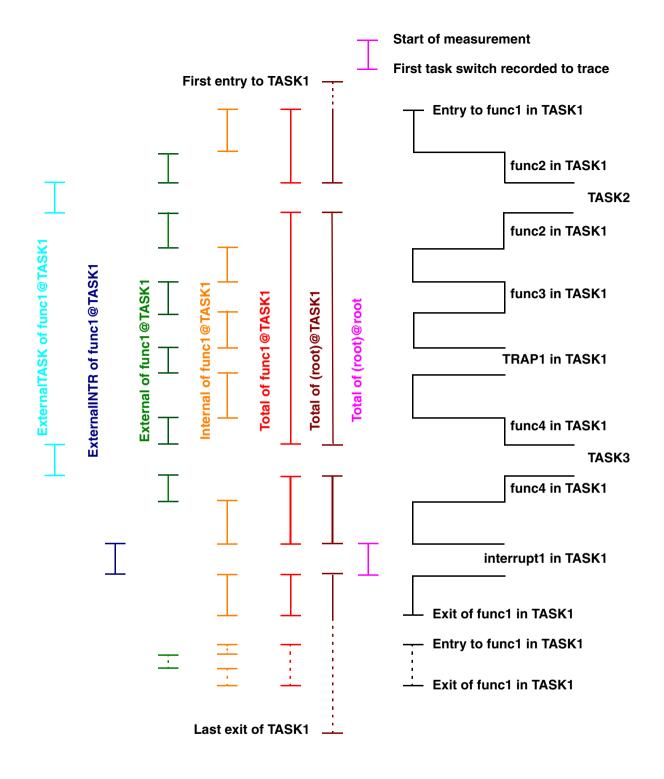


columns (cont.) - tim	es only in function
Internal	total time between function entry and exit without called sub-functions, TRAP handlers, interrupt service routines
IAVeRage	average time between function entry and exit without called sub- functions, TRAP handlers, interrupt service routines
IMIN	shortest time between function entry and exit without called sub- functions, TRAP handlers, interrupt service routines
ΙΜΑΧ	longest time spent in the function between function entry and exit without called sub-functions, TRAP handlers, interrupt service routines
InternalRatio	<internal function="" of="" time="">/<total measurement="" time=""> as a numeric value.</total></internal>
InternalBAR	<internal function="" of="" time="">/<total measurement="" time=""> graphically.</total></internal>

columns (cont.) - times in sub-functions and TRAP handlers			
External	total time spent within called sub-functions/TRAP handlers		
EAVeRage	average time spent within called sub-functions/TRAP handlers		
EMIN	shortest time spent within called sub-functions/TRAP handlers		
ЕМАХ	longest time spent within called sub-functions/TRAP handlers		

columns (cont.) - interrupt times	
ExternalINTR	total time the function was interrupted
ExternalINTRMAX	max. time one function pass was interrupted
INTRCount	number of interrupts that occurred during the function run-time

columns - task/thread related information	
TASKCount	number of tasks that interrupt the function
ExternalTASK	total time in other tasks
ExternalTASKMAX	max. time 1 function pass was interrupted by a task



The standard NEXUS settings do often not allow to locate exactly the instructions that are already executed by a newly activated task. This is especially true is Branch History Messaging is used. This might disturb the task-aware function run-time measurement.

An instruction-accurate assignment of the task switches may improve the results.

#### IEEE-ISTO 5001-2008 and Subsequent Standards

The Ownership Trace Messages (task switches) can be exactly assigned to an instruction, if the following setting is done.

NEXUS.PTCM PID\_MSR ON

NEXUS.POTD ON

- ; enable Program Trace Correlation
- ; Messages for PID0/NPIDR accesses
- ; disable Periodic Ownership Trace
- ; Messages

B::Trace.Lis	t List.TASK DEFault	
🌽 Setup	📭 Goto 🛱 Find 🚺 Chart 🛛 🛄 Profile 🛛 🔜 MIPS 🛛 🖨 More 🛛 🗶 Less	
record	run address cycle data symbol	ti.back
-00013778	se_isync V:400005BE ptrace \\im02_bf1x\os\OSSetPID0+0x0	6 0.160us
	mtpid r3 ; value	*
-00013777	task: TASKO (00000009) V:400005C2 owner 00000009 \\im02_bf1x\os\OSSetPID0+0x0	0A 0.300us
-00013775	se_isync V:400005C8 ptrace \\im02_bf1x\os\OSSetPID0+0x1	10 0.200us -

#### Alternative

; mark instruction that performs the task switch for the task-aware

; function run-time analysis

sYmbol.MARKER.Create TASKSWITCH osDispatcher+0x100

🛓 B::sYmbol.MAR	KER.List 🗖 🗖 🗾	×
address	info	
P:00023578	TASKSWITCH	*
		Ŧ
	4 F	

Detailed analysis of a single function, time between function entry and exit, time spent in interrupt service routines and other tasks is excluded.

Trace.STATistic.FuncDURation <function> [/TASK "<task\_name>"] [/FilterCORE "<core\_number>"]

Setup	iii Groups	Config	Goto	E Detailed	Nesting	- Chart				
betapin		j( a comginj			funcs: 141.		otal: 796.21	4ms intr:	44.876ms	
				range		min			count	
		ScheduleTa			2.070ms	16.386us	23.431us	17.109us	121.	
		fyAlarmAct			934.457us	7.568us	10.783us	7.723us	121.	
		FyAlarmAct			569.101us	4.608us	6.567us	4.703us	121.	
\\sa		oSLeave			3.791ms 579.194us	29.357us 12.991us	39.785us 14.713us	31.402us 13.163us	121.	
occeb)		l\osevt\OS <u></u> ernalDispat			2.460ms	12.991us 18.467us	24.594us	20.440us	121.	<u>6.</u>
		l\PreTaskH			1.023ms	6.705us	11.886us	8.597us	121.	8
sampier		L\osalm\OS			304.401us	6.808us	8.187us	6.918us	44.	
X		samplets1			9.740us	-	9.740us	-	1.	E
`	(bump rez (	samp record (		(TASKE	5.443ms	-	5.443ms	-		
11	sample1\s	amplets1\F			5.429ms	-	5.429ms	-	41.	
		5_OSIocWri			3.476ms	84.187us		84.770us	41.	
	\\samp	le1\osioc\	OSIocActi	ON@TASKE	2.680ms	65.052us	69.119us	65.360us	Statistic	
1/	sample1\o:	smc\OSRemo	teIocActi	on@TASKE	2.514ms	61.049us	64.711us	61.327us	List First	
		sample1\osi			1.980ms	48.099us	50.712us	48.300us	100.0	
		nc\OSISRCo			1.757ms	24.535us	28.610us	26.627us	List Last	
		c∖0SRCHand`			1.623ms	22.517us	26.454us	24.595us	📰 List Max	
		DSSCIocAct			670.376us	8.704us	11.662us	10.475us	Goto Max	
_ / /	samplel\o:	stsk\0S_Tei	rminateTa:	K@TASKE	5.335ms	-	5.335ms	-		
samplel	\ossch\05	TaskTermin	ateDispat	Ch@TASKE	5.326ms	- F27	5.326ms	-	Bookmark Ma	x
1.1		e1\ostsk\0			268.854us	6.527us	6.755us	6.557us		
11	sampier\s	amplets1\P	OSTIASKHO	DRETASKE	32.498us	0.761us	0.823us	0.793us	📰 Linkage	
					•	III			Parents	
									E Children	
									E Children	
									Duration Anal	vsis
									Findall Duratio	
									Distance Anal	
									Findall Distan	-

	3								x3B34	uncDURation P:0	B::Trace.STAT.F	F
samples:         64.         avr:         84.105us         min:         82.847us         max:         91.416us         511           total:         796.214ms         in:         5.383ms         out:         790.831ms         ratio:         0.676%           up to         count         ratio         1%         2%         5%         10%         20%         50%         100           < 82.000us							Full	Zoom	oom 🛛 📮 Z	Chart 🗘 🗘 Z	Setup	
up to         count         ratio         1%         2%         5%         10%         20%         50%         100           <		511					84.105us	avr:	64.	samples:		
<         82.000us         0.         0.000%           83.000us         23.         35.937%			: 0.676%	ratio	790.831ms	out:	5.383ms	in:	96.214ms	total: 7		
<         82.000us         0.         0.000%           83.000us         23.         35.937%		100	50%	20%	10%	5%	2%	1%	ratio	count	up to	
84.000us         0.         0.000%           85.000us         38.         59.375%           86.000us         0.         0.000%								1	0.000%	0.	82.000us	<
85.000us 38. 59.375% 86.000us 0. 0.000%			-									
86.000us 0. 0.000%												
87.000us 0. 0.000%												
88.000us 0. 0.000%												
89.000us 0. 0.000%												
90.000us 0. 0.000%												
91.000us 0. 0.000%												
92.000us 3. 4.687%						-						
93.000us 0. 0.000%												
94.000us 0. 0.000%												
95.000us 0. 0.000%												
96.000us 0. 0.000%												
97.000us 0. 0.000%												
98.000us 0. 0.000%											98.000us	
▶ 0. 0.000%	*								0.000%	0.		
		Þ								•		

Please be aware, that details are shown for all function runs. If you are interested in a task-specific analysis you have to use the **/TASK** "*<task\_name>*" option.

F B	::Trace.STAT.F	uncDURatio	on P:0	x3B34 /TASK	"TASKE"							×
B	Setup 🛛 📶	Chart	‡Z(	oom 🗋 🛓 🛛	Zoom	Full						
		samples:		41.	avr:	84.770us	min:	84.187us	max:	91.416us	40	w
		total:		96.214ms	in:	3.476ms	out:	792.738ms	ratio	: 0.436%		
		count			1%	2%	5%	10%	20%	50%	100	
<	84.000us		0.	0.000%								
	84.500us 85.000us	-	38. 0.	92.682% 0.000%							-	
	85.500us		ö.	0.000%								
	86.000us		ŏ.	0.000%								
	86.500us		0.	0.000%								
	87.000us		0.	0.000%								
	87.500us		0.	0.000%								
	88.000us 88.500us		0. 0.	0.000%								
	89.000us		ö.	0.000%								
	89.500us		ŏ.	0.000%								
	90.000us		0.	0.000%								
	90.500us		0.	0.000%								
	91.000us 91.500us		0. 3.	0.000%								
	91.500us 92.000us		5. 0.	7.317%								
>	52.00003		ŏ.	0.000%								-
												in the second second

The @(interrrupt) and the @(unknown) task are split up per core.

funcs: 141. total: 796.214ms intr: 44.876ms range total min max avr count (root)@(interrupt):0 →\samplel\Global\VTABLE+0x40@(interrupt):0 (samplel\ostsk\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (samplel\SocheckStack@(interrupt):0 (samplel\SocheckStack@(interrupt):0 (samplel\SocheckStack@(interrupt):0 (samplel\Soch\SocheckStack@(interrupt):0 (sa	🛿 Setup) 🚺 Groups) 💶 Config) 📭 Goto ) 🗾 Detailed) [ 😨 Nesti					
(root)@(interrupt):0       31.078ms       55.610us       109.905us       62.609us       496.         (sample1\Golal\TABLE+0x40@(interrupt):0       29.307ms       52.075us       106.145us       59.035us       496.         (sample1\osts/OSCHeckStack@(interrupt):0       29.307ms       52.075us       106.145us       59.035us       496.         (sample1\osts/OSCHeckStack@(interrupt):0       1.472ms       2.880us       6.709us       2.968us       496.         (sample1\osts/OSCHeckStack@(interrupt):0       7.90ms       18.000us       33.800us       18.117us       Statistic         (sample1\ostal\OSCHeckAlarmaction@(interrupt):0       7.90ms       18.000us       32.83us       12.350us       12.352us       12.352us         (sample1\ostal\OSCHotifyAlarmAction@(interrupt):0       42.333us       7.810us       8.883us       8.467us       111						
\\samplel\oscr\OSISRSystemTimer&(interrupt):0       7.790ms       18.000us       33.800us       18.117us       Statistic         \\samplel\osalm\OSCheckAlarms&(interrupt):0       2.938ms       6.719us       22.473us       6.834us         \\samplel\osalm\OSCNotifyAlarmAction@(interrupt):0       37.056us       12.350us       12.352us       12.352us       12.352us       12.452us         \\samplel\osalm\OSCNotifyAlarmAction@(interrupt):0       42.333us       7.810us       8.883us       8.467us       6.573us       13.14us       111111111111111111111111111111111111	(root)@(interrupt):0 →\\sample1\Global\VTABLE+0x40@(interrupt):0 \sample1\osisr\OSInterruptDispatcher1@(interrupt):0	- 31.078ms 29.307ms	- 55.610us 52.075us	- 109.905us 106.145us	- 62.609us 59.035us	496. 496.
	<pre>\\samplel.osctr\OSISRSystemTimer@(interrupt):0 \\samplel.osalm\OSCheckAlarms@(interrupt):0 \\samplel.osalm\OSCNotifyAlarmAction@(interrupt):0 \\samplel.osalm\OSCNotifyAlarmAction@(interrupt):0 \\samplel.ossetVoSSetEvent@(interrupt):0 \\samplel.ossetVoSLeaveISR@(interrupt):0 \samplel.osskInternalDispatch@(interrupt):0</pre>	7.790ms 2.938ms 37.056us 42.333us 289.209us 2.486ms 1.457ms 1.984ms	18.000us 6.719us 12.350us 7.810us 2.793us 34.382us 21.039us 45.916us	33.800us 22.473us 12.355us 8.883us 6.954us 38.104us 22.445us	18.117us 6.834us 12.352us 8.467us 6.573us 37.942us 22.314us	Statistic List First List Last Goto Max Bookmark Max Linkage Parents

Image: Chart         Q Zoom         Q Zoom         Q Full           samples:         809.         avr:         3.599us         min:         2.670us         max:         6.755us           total:         796.214ms         in:         2.911ms         out:         793.303ms         ratio:         0.365%           up to         count         ratio         1%         2%         5%         10%         20%         50%         100	
samples: 809. avr: 3.599us min: 2.670us max: 6.755us total: 796.214ms in: 2.911ms out: 793.303ms ratio: 0.365%	
total: 796.214ms in: 2.911ms out: 793.303ms ratio: 0.365%	
$10^{\circ}$ $10^{\circ}$ $10^{\circ}$ $10^{\circ}$ $10^{\circ}$ $10^{\circ}$ $10^{\circ}$	
up to count ratio 1% 2% 5% 10% 20% 50% 100	
< 2.500us 0. 0.000%	
3.000us 548. 67.737%	
3.500us 63. 7.787%	
4.000us 8. 0.988% <del>4</del>	
4.500us 0. 0.000%	
5.000us 55. 6.798%	
5.500us 0. 0.000%	
6.000us 0. 0.000%	
6.500us 67. 8.281%	
7.000us 68. 8.405%	
7.500us 0. 0.000%	
8.000us 0. 0.000%	
8.500us 0. 0.000%	
9.000us 0. 0.000%	
9.500us 0. 0.000%	
10.000us 0. 0.000%	
10.500us 0. 0.000%	
0. 0.000%	-
	▶

Please be aware, that details are shown for all function runs. If you are interested in a core-specific analysis you have to use the **/FilterCORE** *<core\_number>* option.

	23
up to         count         ratio         1%         2%         5%         10%         20%         50%         100           <         2.500us         0.         0.000%         3.000us         430.         62.957%         5%         10%         20%         50%         100           <         3.500us         63.         9.224%	
2.500us         0.         0.000%           3.000us         430.         62.957%           4.000us         63.         9.224%           4.000us         0.         0.000%           4.500us         0.         0.000%           5.000us         0.         0.000%           5.000us         0.         0.000%           5.500us         0.         0.000%	
3.000us         430.         62.957%           3.500us         63.         9.224%           4.000us         0.         0.000%           5.500us         0.         0.000%           5.000us         55.         8.052%           5.500us         0.         0.000%	
3.500us         63.         9.224%           4.000us         0.         0.000%           4.500us         0.         0.000%           5.000us         55.         8.052%           5.500us         0.         0.000%	~
4.000us         0.         0.000%           4.500us         0.         0.000%           5.000us         55.         8.052%           5.500us         0.         0.000%	
4.500us         0.         0.000%           5.000us         55.         8.052%           5.500us         0.         0.000%	
5.000us 55. 8.052% 5.500us 0. 0.000%	
5.500us 0. 0.000%	
6 000 0 0 0 000%	
6.500us 67. 9.809%	
7.000us 68. 9.956%	
7.500us 0. 0.000% 8.000us 0. 0.000%	
8.500us 0. 0.000%	
9.000us 0. 0.000%	
9.500us 0. 0.000%	
10.000us 0. 0.000%	
10.500us 0. 0.000%	
> 0. 0.000%	-
	▶

# Nested function run-time analysis - graphical display

erf Cov MPC5XXX ORTI	AUTOS
Perf Configuration	
Perf <u>L</u> ist	
Perf List Dynamic	
Eunction Runtime	Prepare
Distribution	Show <u>N</u> umerical
Duration A to B	Show as <u>Tree</u>
Distance trace records	Show <u>D</u> etailed Tree
Task Runtime	Show as T <u>i</u> ming
Task Services	Show Nesting
Task ISR2s	+
Task Function Runtime	•
Task <u>S</u> tatus	•
<u>R</u> eset	

Setup 🚻 Groups 📲 Config 🔼 Goto	Find	<b>♦</b> In		It KNF	ull										
			-232	. 95 Oms	5		- 2	32.90	Oms		-23	32.850	ms		
	range	10		1				1							
(root)	@TASKSND1	•													 
\\sample1\samplets1\PreTaskHook	@TASKSND1	41													
\\sample1\samplets1\FuncTASKSND1						-					-		-		 
<pre>umple1\osioc\OS_OSIocWriteAcrossRef</pre>	@TASKSND1	41										-			
\\sample1\osioc\OSIocAction	@TASKSND1	<b>O</b>										_			
\\sample1\osmc\OSRemoteIocAction	@TASKSND1	<b>()</b>						<u> </u>				- <b>İ</b>			
\\sample1\osmc\OSWaitRC	@TASKSND1	41													
\\sample1\ostsk\OS_TerminateTask	@TASKSND1	÷												<u> </u>	 
p]e1\ossch\OSTaskTerminateDispatch															 <b>_</b>
\\samp]e1\ostsk\0SCheckStack															- ·
\\sample1\samplets1\PostTaskHook		<b>D</b> D													
mple1\ossch\OSTaskInternalDispatch				1.0	· .			1.1		 					

Trace.STATistic.TREE

Perf Cov MPC5XXX ORTI_AU	ITOS
🔑 Perf Configuration	
Perf List	
Perf List Dynamic	
Function Runtime	Prepare
Distribution	Show Numerical
Duration A to B	Show as Tree
Distance trace records	Show Detailed Tree
Task Runtime	💦 🙀 Show as Timing
Task Services	Show Nesting
Task ISR2s	•
Task Function Runtime	•
Task Status	•
Reset	

Setup 👖 Groups 📰 Config 📭 Goto 🗐	Detailed 🗱 Nesting 🛛 🖼 Chart						
		630ms					
	tunes: 25. cocur. 22.	050115					
range	tree		min			count	intern% 1%
	🖻 (root)	22.630ms	-	22.630ms	22.630ms	-	0.059% +
\\sample1\samplets1\PreTaskHook		0.865us	0.865us	0.865us	0.865us	1.	0.003% +
\\sample1\samplerv1\FuncTASKRCV1	- FuncTASKRCV1	22.615ms	-	22.615ms	22.615ms	1. (0/1)	3.036%
sample1\osisr\OS_DisableAllInterrupts		40.210us	8.015us	8.140us	8.042us	5.	0.177% +
<pre>\sample1\osisr\OS_EnableAllInterrupts</pre>	— OS_EnableAllInterrupts	40.570us	8.015us	8.140us	8.114us	5.	0.179% +
\\sample1\osalm\OS_SetRelAlarm		3.902ms	37.360us	100.395us	41.074us	95.	13.205%
\\sample1\osmc\OSWaitRC		341.095us	68.180us	68.325us	68.219us	5.	1.507%
\\sample1\osalm\OSInsertAlarm	OSInsertAlarm	572.445us	6.285us	7.650us	6.360us	90.	2.529%
ample1\ossct\OS_StartScheduleTableRel	—⊟ OS_StartScheduleTableRel	816.825us	163.165us	163.880us	163.365us	5.	0.716% +
\\sample1\osmc\OSWaitRC		654.795us	130.940us	130.985us	130.959us	5.	2.893%
\\sample1\osevt\ <b>OS_ClearEvent</b>	<ul> <li>— 0S_ClearEvent</li> </ul>	1.159ms	12.700us	14.800us	12.882us	90.	5.123%
\\sample1\osevt\OS_WaitEvent	—	5.654ms	24.165us	65.620us	62.824us	90. (0/1)	7.003%
\\sample1\ossch\OSTaskForceDispatch	□ □ OSTaskForceDispatch	4.069ms	45.750us	48.345us	47.873us	85. (0/1)	7.865%
\\sample1\ostsk\0SCheckStack		455.520us	5.300us	5.430us	5.359us	85.	2.012%
\\sample1\samplets1\PostTaskHook	— PostTaskHook	73.450us	0.860us	0.865us	0.864us	85.	0.324% 🗲
sample1\ossch\OSTaskInternalDispatch			19.235us	20.970us	20.709us	85.(0/1)	7.146%
\\sample1\samplets1\PreTaskHook	PreTaskHook	143.020us	1.230us	1.730us	1.703us	84.	0.632% 🗲
\\sample1\osevt\05_GetEvent		1.514ms	15.415us	18.250us	17.010us	89.	6.689%
<pre>\\sample1\osalm\OS_CancelAlarm</pre>	— 🗉 OS_CancelAlarm	2.490ms	29.095us	31.570us	29.291us	85.	9.195%
\\sample1\osalm\OSKillAlarm	└── OSKillAlarm	408.775us	4.805us	4.815us	4.809us	85.	1.806%
\\sample1\osres\OS_GetResource	— OS_GetResource	2.138ms	25.150us	25.165us	25.158us	85.	9.449%
\\samp]e1\osioc\OS_OSIocReadAcross	— OS_OSIocReadAcross	1.488ms	17.505us	17.515us	17.511us	85.	6.577%
\\sample1\osres\OS_ReleaseResource	— OS_ReleaseResource	2.642ms	31.070us	31.085us	31.078us	85.	11.673%
\\sample1\osmc\05_GetSpinlock	- OS_GetSpinlock	22.690us	22.690us	22.690us	22.690us	1.	0.100% +
\\sample1\osmc\OS_ReleaseSpinlock	└── OS_ReleaseSpinlock	20.225us	20.225us	20.225us	20.225us	1.	0.089% +

It is also possible to get a task-specific tree.

Trace.STATistic.TREE /TASK "TASKRCV1"

# Trace.STATistic.LINKage <address>

# Nested function run-time analysis - linkage analysis

B::Trace.STATistic.FUNC							
Setup 👖 Groups 📲 Config 🖪 Goto 📑 Detailed 📲							
	funcs: 132.	to	tal: 6/4.4/	'6ms intr:	41.031ms		
range	total	min	max	avr	count	intern% 1	%
\\sample1\osmc\OS ReleaseSpinlock@TASKCNT	20.720us	20.720us	20.720us	20.720us	1.	0.001% +	
\\samplel\osalm\OSNotifyAlarmAction@TASKCNT \\samplel\osmc\OSRemoteNotifyAlmAction@TASKCNT	130.970us	65.485us	65.485us	65.485us	2. 2. 2.	<0.001% +	
\\samplel\osmc\OSRemoteNotityAImAction@IASKCNI \\samplel\osmc\OSWaitRC@TASKCNT	126.530us 109.260us	63.265us 54.625us	63.265us 54.635us	63.265us 54.630us	2.	0.001% + 0.008% +	
\\sample1\samplets1\PreTaskHook@TASKCNT	24.650us	0.615us	0.620us	0.616us	40.	0.001%	
\\sample1\samplets1\FuncTASKCNT@TASKCNT	2.844ms	-	2.844ms	1.438ms	40.	0.008% +	
\\sample1\osctr\05_IncrementCounter@TASKCNT	1.236ms	27.135us	99.645us	30.898us	40.	0.064%	
\\sample1\osalm\ <b>OSCheckAlarms</b> @TASKCNT	359.155us	5.425us	74.485us	8.979us	40. 40.	Statistic	
\sample1\ostsk\ <b>OS_TerminateTask</b> @TASKCNT \sample1\ossch\ <b>OSTaskTerminateDispatch</b> @TASKCNT	2.813ms 2.807ms	-	2.813ms 2.807ms	1.403ms 1.396ms	40.	List First	
(Joanprez (055ch (05raskreininnaeebrspacenerAsken)	4			1.55005		List Last	
-						List Max	
						Goto Max	
						Bookmark Max	
						DOOKINGIK IVIAX	_
					F	Linkage	
						Parents	
					E	Children	
					=	Duration	
					E	Distance	
							-

E::Trace.STAT.LINKage C:0x4A10							×
🌽 Setup 🚺 Groups 🔡 Config 📭 Goto	🗾 Detaile	d [ Nesting	Chart				
	funcs: 2.	to	tal: 2.68	37ms			
range				avr	count	total% 1%	
\\sample1\osctr\OSISRSystemTimer	2.327ms	6.285us	22.075us		365.	86.631%	-
\sample1\osctr\OS_IncrementCounter	359.155us	5.425us	74.485us	8.979us	40.	13.368%	_
							<b>T</b>
	•		III				•
-							111

TRACE32 also provides an interface to third-party timing tools. For details refer to "Trace Export for Third-Party Timing Tools" (app\_timing\_tools.pdf).

The command group GROUP allows to structure the software for the trace evaluation. This is especially useful if the software consists of a huge number of functions/modules.

# **GROUP** Creation

GROUP.Create

If the command **GROUP.Create** is entered without parameters, the **Group.Create** dialog is opened.

iii B::GROUP.Create				
name				GROUP name
addressrange(s)			- 1	GROUP members
options Enable	🔲 Hide	Merge	NONE	GROUP attributes
Ok	Set	Delete	Cancel	

The basic setup for a GROUP includes the following steps:

#### 1. Specify the GROUP name.

B::GROUP.Create			- • •
- name my_group			
- addressrange(s)			• 1
options	🕅 Hide	C Merge	NONE
Ok	Set	Delete	Cancel

#### 2. Specify the GROUP members.

GROUPS are address ranges, so you can use functions, modules, or programs to specify the group members.

iii B::GROUP.Create name my_group addressrange(s)				Open the symbol data base
options	🗌 Hide	🗖 Merge	NONE	to select the group members
Ok	Set	Delete	Cancel	

A new group member is selected by a double-click.

iii B::GROUP.Create		×	
name			
my_group			
L			
- addressrange(s)			
func10 func11	•	2	
	2 Browse Symbols		
options	* * *	<b>↑ "</b> ↓ Type:	Functions
Enable Hide	symbol	type	address
L	background	(int ())	P:40001368400013AF
	funcŌ func1	(int ()) (static void ())	P:400000304000004B P:4000004C4000007F
Ok Set	func10	(int ())	P:40000798400008F
	func11	(int ())	P:40000BF040000C87
	func13	(int ())	P:40000C8840000CEB
	func14	(int ())	P:40000CEC40000D17
	func15	(int ())	P:40000D1840000D47
	func16	(int ())	P:40000D4840000D73
	func17 func18	(int ()) (int ())	P:40000D7440000DB3 P:40000DB440000DEF
	func19	(int ())	P:40000DE4-40000EP
	func2	(void ())	P:400008040000113
	func20	(int ())	P:40000E2C40000E7F -

#### 3. Specify the GROUP color and close the dialog with Ok.

The GROUP color is used to mark the GROUP members in the trace analysis windows.

🙀 B::GROUP.Creat	e		
name			
my_group			
- addressrange(s func10 func11 fun	)	5 func17 func18 func19	9 func20 🔻 👔
options			
🗹 Enable	🔲 Hide	Merge	OLIVE -
Ok	Set	Delete	Cancel

4. Display the GROUP information.

View Var Break	lun CPU N	
Registers		
111 Dump		
List Source		
😽 Watch		
😹 Referenced Var		
🚱 Locals		
😽 Stackframe with	cals	
6 Stackframe		
🐲 Peripherals		
Symbols	•	
📅 Groups		
Bookmarks		
🧮 Trace List	👖 B::Group.List 📃 🗖 🗖	
📃 Message Area	🗱 Reset 🔘 Disable All 🔘 Enable All 🔘 Hide All 🔘 Show Al 😰 Store 😨 Load 🙀 Create 🙀 Create	
	group enable hide merge color	1
	Tunc10	*
	func11 ý OLIVE	
	func13 ý OLIVE func14 ý OLIVE	
	func15 ý OLIVE	
	func16 ý OLIVE func17 ý OLIVE	
	func18 ý OLIVE	
	func19 🗸 OLIVE	
	func20 V OLIVE	
	• • • •	1
		_

5. Push the Store... button, if you want to generate a scipt that allows you to re-set the specified groups at any time.

```
; script group_settings.cmm
B::
GROUP.RESET
GROUP.CREATE "my_group" \\diabc\diabc\func10 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func11 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func13 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func14 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func15 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func16 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func17 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func18 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func19 /OLIVE
GROUP.CREATE "my_group" \\diabc\diabc\func20 /OLIVE
```

ENDDO

The GROUP status determines the appearance of a GROUP in the trace display and analysis windows. The following three statuses are available:

- ENable
- ENable + Merge
- ENable + HIDE

# **GROUP Status ENable**

💥 Reset 🛛 🔿 Disable All 🖉 Enable	All 🖸 Hide All 🔘 Show All	Store Store	🙀 Create) 🚺 👔 Create Task.
group	enable hide	merge color	
🖻 "my_group"		OLIVE	
func10	V V	OLIVE	
func11	V V	OLIVE	
func13	V V	OLIVE	
func14	V V	OLIVE	
func15	V V	OLIVE	
func16	V V	OLIVE	
func17	V V	OLIVE	
func18	V V	OLIVE	
func19	V V	OLIVE	
func20	V V	OLIVE	
"other"	V V		

TRACE32 provide the following features if a GROUP has the status ENable:

1. GROUP members are marked in the Trace Listing by their group color.

B::Trace.Lis	t									×
🔑 Setup	🔒 Goto	. 📑 Find	Chart	🔼 Profile	MIPS	More	Less	]		
recor -000214685		address	cycle 77C ptrace	data		symbol	-hele	0.0.60	ti.back 0.495us	
-000214665	'   ¦	F:40000	//c purace			\\dTabc\d	iabc\func9	HUXOC	0.49505	Ê
		}	-							-
35	1	re	turn &stat1:							
35		subi	r3,r13,0x7Fi	C ;	r3,r13,32	2748				
		} lwz	r30,0x10(r1)	) ;	r30,16(r1	)				
		lwz lwz	r31,0x14(r1) r0,0x1C(r1)	) ;	reg1,20(r	1)				
		mtlr	r0,0x1C(r1)	i	r0,20(r1)					
		addi blr	r1,r1,0x18	;	r1,r1,24					
-000214685			1C8 ptrace			\\diabc\d	iabc\main+	0x16C	1.235us	
62	1 [		func1	oO:						
000014605			0x40000798		func10			•	0.370	
-000214685	۲ <u>ا</u>		798 ptrace p "my_group			\\diabc\d	1 abc \Tunci	.0	0.370us	_
	II r	func10()								
35	5	{								
			r1,-0x58(r1) r0	);	r1,-88(r1	.)				
			r0 r14,0x10(r1)	) ;	v16,16(r1	)				
		stw	r0,0x5C(r1)		r0,92(r1)					
			gister i, j gister v1, v		/4, v5, v6	5, v7, v8:				
			ğister v9, v				5, v16, v1	.7;		
1										

### 2. Special statistic commands are provide for the GROUPS.

Trace.STATistic.GROUPGroup-based run-time analysis.

Trace.Chart.GROUP

Group-based time chart.

B::Trace.STATistic.GROUP				- • •
		etailed 🗧 Tree 🛛 👭 Cha		
iter	ms: 2. tot	al: 2.072s sample	s: 15126968.	
address tota			count ratio% 1%	2%
	1.864s 0.365us 7.797ms 1.230us	1.336ms 132.950us 119.295us 14.821us	14020. (0/1) 89.970% 14020. 10.030%	A
group my_group 20	7.79/ms   1.250us	119.295us   14.021us	14020.   10.030%	-
				H. 4

💥 Reset 🛛 🔿 Disable All 🔘 Enable A	II 🔘 Hide All 🖉 Show All	🔁 Store 🛛 😰 Loa	ad ] 🙀 Create] 👔 Create Task.
group	enable hide	merge colo	r
⊟ "my_group"	V	OLIV	Έ
func10	V V	OLIV	Έ
func11	V V	OLIV	Έ
func13	V V	OLIV	Έ
func14	V V	OLIV	Έ
func15	V V	OLIV	Έ
func16	V V	OLIV	Έ
func17	V V	OLIV	Έ
func18	V V	OLIV	Έ
func19	V V	OLIV	Έ
func20	V V	OLIV	Έ
"other"	V V	V	
	· · ·	· · ·	

TRACE32 provide the following feature if a GROUP has the status ENable and Merge:

The GROUP represents its members in all trace analysis window. No details about the GROUP members are displayed.

Config	L C	oto	<b>N</b>	Got	<b>)</b>	<b>F</b> F	ind	•	•□• In	•	<b>10</b>	It E	D F	ull										
000s		-	2.0	2077	0000	s			-2.	020	760	000	s			- 1	2.0	207	500	009	5			
θİ				ι.						1								ı.						
•																						<u> </u>		
•				· •	i i	<u> </u>	È	É.		Ľ.,	_ <b>_</b>		È.						-				فع	
•				: [					· · · [		· 🗆													-
•				1							1													
											1				·  ·									- 1
											1				·  ·									- 1
H · ·															·  ·									- 1
16											1.1				·  ·									
								- L							· •									
		000s	000s -	000s -2.0	000s -2.02077	000s -2.020770000	000s -2.020770000s	000s -2.020770000s	000s -2.020770000s	0005 -2.0207700005 -2.	0005 -2.0207700005 -2.020 0 0 0 0 0 0 0 0 0 0 0 0 0	000s -2.020770000s -2.020760	0005 -2.0207700005 -2.020760000	000s -2.020770000s -2.020760000s	0005 -2.0207700005 -2.0207600005	000s -2.020770000s -2.020760000s	000s -2.020770000s -2.020760000s -	0005 -2.0207700005 -2.0207600005 -2.0	0005 -2.0207700005 -2.0207600005 -2.0207	0005 -2.0207700005 -2.0207600005 -2.0207500	0005 -2.0207700005 -2.0207600005 -2.0207500005	0005 -2.0207700005 -2.0207600005 -2.0207500005	000s -2.020770000s -2.020760000s -2.020750000s	0005 -2.0207700005 -2.0207500005

💥 Reset 🛛 🔿 Disable All 💿 Enable A	All 🔘 Hide All 🔘 Show All	🖁 Store 🛛 😰 Load 👔	🕯 Create) 🚺 🕅 Create Task
group	enable hide		
⊟ "my_group"	V	OLIVE	
func10	V V	OLIVE	
func11	V V	OLIVE	
func13	V V	OLIVE	
func14	V V	OLIVE	
func15	V	OLIVE	
func16	V V	OLIVE	
func17	V V	OLIVE	
func18	V V	OLIVE	
func19	V	OLIVE	
func20	V	OLIVE	
"other"	- V V		

TRACE32 provide the following feature if a GROUP has the status ENable and HIDE:

1. The GROUP represents its members in all trace analysis window. No details about the GROUP members are displayed.

Setup iii Groups 🔡 Con	fig	A	Goto.	][	A (	Goto	]	Í	Fir	nd		<b>∙⊡• I</b> r		•[]• (	Out	•	F	III								
	000s			- 2	. 020	077	000	0s				-2.	02	076	000	)0s	;		-	2.0	020	750	000	)s		
address 💀					. 1								1								1					
(other) 💀											_				_											
group "other" 🚯						1																				
\diabc\diabc\func11																		•							•	
\diabc\diabc\func13											1															
\diabc\diabc\func14 🚯 \diabc\diabc\func15 🚯																										
\diabc\diabc\func16											ŀ				H			•							•	
\diabc\diabc\func17											1				11											

2. The trace information recorded for the GROUP members is hidden in the Trace Listing.

B::Trace.List					
Setup 📭	Goto jij Find	🛛 🕂 Chart 🛛 🔼	Profile 🛛 🏧 MIPS	More Less	
record	run address		ata	symbol	ti.back
-0001406839		01220 ptrace	; x2,8(r1) ; x1,12(r1 ; r0,20(r1 ; r1,r1,16	}	
519	ğrα	oup "other" oup "my_group" - 20( x1, x2, x3 ) , x2, x3;	) /* Par	ameter: 3 Short */	1.850us 0.365us
	stwu mflr stw stw stw stw	r1,-0x18(r1) r0 r29,0x0C(r1) r30,0x10(r1) r31,0x14(r1) r0,0x1C(r1)	; r1,-24(r ; x3,12(r1 ; x2,16(r1 ; x1,20(r1 ; r0,28(r1	)	Ţ
	<b>I</b> €				►

The manual "Application Note for Trace-Based Code Coverage" (app\_code\_coverage.pdf) gives a detailed introduction to the trace-based code coverage.

Since the core information is discarded for trace-based code coverage, there is no difference between single-core and SMP TRACE32 instances with regard to this feature.

It is recommended to enable Branch History Messaging for Code Coverage. This advises the Nexus module to generate the trace messages in a compact way.

B::NEXUS		-		
nexus	- selection	- option	configuration —	CLIENT1
OFF	🔽 ВТМ	SmartTrace	- PortSize	- SELECT
ON	🗷 НТМ	POTD	MDO4 v	NONE -
	ОТМ	- STALL	- PortMode	MODE
RESet	MTM 🗌	OFF 👻	1/2 🔹	OFF 👻
🥬 Trace	DQM	suppression	DDR	
🔣 List	- DTM	SpenDQM		CLIENT2
	OFF 🔻	SpenWTM		- SELECT
	- PTCM	SpenPTM		NONE -
	PID_MSR	SpenDTM		MODE
	BL_HTM	SpenOTM		OFF 👻
	TLBNEW	- SupprTHReshold -		
	TLBINV	1/4 🔻		
1				

NEXUS.BTM ON

Incremental Code Coverage has to be used in the following situations:

- POWERTRACE/ETHERNET as universal debug and trace hardware
- POWER TRACE PX as universal trace hardware
- On-chip trace memory
- High-bandwidth trace interfaces