

# **Application Note for Trace.List**

Release 09.2023



TRACE32 Online Help		
TRACE32 Directory		
TRACE32 Index		
TRACE32 Documents		
Trace Application Notes		
Trace Analysis		
Application Note for Trace.List		1
Overview List items Record items Pre-defined Item Groups		<b>4</b> 4 5 5
Item reference: List items		6
List.Asm List.ADDRESS List.CODE List.Comment List.DIAG List.Dummy List.EXEC List.HII List.Label List.Label List.Mix List.NoDummy List.NoFetch List.SOURCE List.SOURCE List.SOURCEFILE List.SYmbol List.TASK List.TIme	Show disassembled mnemonics Show program address Show program opcode Show comments to disassembled mnemonics Disassembler-related diagnostic information Show dummy cycles Execution information about conditional branches Show source code Show associated label Show disassembled mnemonics and HLL code Suppress the display of dummy cycles Suppress the display of program fetches Display associated source file name Display associated source file name Show debug info for every linear program block Display OS Awareness information Display time information for HLL lines	6 7 7 8 9 9 9 10 10 10 11 11 11 12 12
		13
Address CPU CYcle Run SIGNALS sYmbol	Start address contained in trace record Set of items Run, Address, CYcle, Data and sYmbol Show bus cycle type of trace record Core and execution information Trace-port signals Debug symbol of start address in trace record	13 13 13 13 14 15 15

Item reference: CLOCKS items		16
CLOCKS.Back	Number of clocks relative time to previous record	16
CLOCKS.Fore	Number of clocks relative time to next record	16
CLOCKS.REF	Number of clocks relative to reference point	17
CLOCKS.Trigger	Number of clocks relative to trigger point	17
CLOCKS.Zero	Number of clocks relative to global zero point	17
Item reference: Data items		18
Data.any	Data value stored in trace record (auto width)	18
Data.B <x></x>	Data value stored in trace record (byte lanes)	18
Data.L <x></x>	Data value stored in trace record (long lanes)	18
Data.Oct	Data value stored in trace record (oct-word)	18
Data.Q <x></x>	Data value stored in trace record (oct-word)	19
Data.T <x></x>	Data value stored in trace record (tbyte)	19
Data.W <x></x>	Data value stored in trace record (word)	19
Data.sYmbol	Debug symbol associated to data value	19
Item reference: ENERGY items		20
ENERGY.Abs	Energy consumption since start of trace data	20
ENERGY.Back	Energy consumption since previous record	20
ENERGY.Fore	Energy consumption until next record	20
ENERGY.REF	Energy consumption since reference record	20
ENERGY.Trigger	Energy consumption since trigger point	20
ENERGY.Zero	Energy consumption since global zero point	20
Item reference: TIme items		22
TIme.AddressBack	Time relative to previous occurrence of address	22
TIme.AddressFore	Time relative to next address	23
TIme.Back	Time relative to previous record	23
TIme.Fore	Time relative to next record	23
TIme.FUNC	Time spent to execute a function	23
TIme.FUNCEX	Time spent outside the current function	23
TIme.FUNCIN	Time spent in code of function	24
TIme.MARK <x>BACK</x>	Time relative back to the last marker	24
TIme.MARK <x>FORW</x>	Time relative forward to the next marker	24
TIme.REF	Time relative to reference point	24
TIme.Trigger	Time relative to trigger point	24
TIme.Zero	Time relative to global reference	25

Version 09-Oct-2023

# Overview

This application note contains information about *Trace.List items*. In the context of this document, Trace.List items will usually be referred to as "items".

When a trace recording is processed, the resulting information about the trace messages and of the reconstructed program flow is displayed in various windows, e.g. in **Trace.List** or **Trace.FindAll** windows. Trace.List items are the pieces of information of a certain type, which represent certain kinds of information of the recorded trace or the information derived from the recorded trace.

There are two groups of items, one are List items and Record items.

#### List items

List items represent the program flow information that is reconstructed from the trace messages. These items are shown between two trace records. They are not assigned to a column, and their order and appearance is not configurable. All list items have the format List[.<subitem>][.subitem].

A trace listing with record items and list items:

B::Trace	e.List									×	
		nfig 🔒 G		Chart			More	Less			
ree	700	n address	cycle	data		symbol			ti.back		
		addi	r28,r28,0x1	; anz	ahl, anzahl	1,1				=	
	689	addi	<pre>for ( i = 0 ;     r31,r31,0x1     0v40001300</pre>		,1					^	A
-000000	1125	P:400	0 <b>013D0</b> ptrace			\\diabc_in	t\diabc\s	ieve+0x48	0.500u	s	
		bgt	0x40001424	; .15							
	691	lis addi lbzx cmpwi	r12,0x4000 r12,r12,0x50 r12,r12,r31	520 ; r	,16384 12,r12,22( ,r12,i	)48				~	A

#### A List Items

Record items directly refer to information directly related to a trace record. These items are arranged in columns and have a column title. The column width is usually configurable and the columns/items can be defined to appear multiple times in one line.

A trace listing containing several record items:

🔡 B::Trace	.List F	Run Address C	Ycle Data	a Time.REF s	Ymbol							x
🔑 Setup.	🎖	Config	🔒 Goto	🎒 Fir	nd 🔂	Chart	📕 Profile	📕 MI	PS 🗧	More	Less	
record	run	address		cycle	data	ti.	ref sy	/mbol				
-000010				fetch				∖diabp4`				~
-000009					0000000	В					P_END+0xF7C	
-000008		P:FFF	00610	fetch	8001001	2	\'	diabp4	\diabp4	\func	9+0x78	_
-000007					FFF010D	2	N'	diabp4	Global	\S	P_END+0xF84	~
-000006		P:FFF	00614	fetch	7C0803A	6	Ň	diabp4	\diabp4	\func	9+0x7C	~
-000005		P:FFF	00618	fetch	3821001	8	Ň	diabp4	\diabp4	\func	9+0x80	
-000004		P:FFF	0061C	fetch	4E80002	0	Ň	diabp4	\diabp4	\func	9+0x84	
-000003		P:FFF	010DC	fetch	4BFFF55	1	Ň	diabp4	\diabp4	\main	+0x118	
-000002		P:FFF	0062C	fetch	7FE0000	8	Ň	diabp4	\diabp4	func	10	
-000001	BRK					_	_			-		- v
	<											>

#### **Pre-defined Item Groups**

The following pre-defined item groups are available:

DEFault	Default trace display. The default trace display for the <b>Analyzer</b> can be configured with the command <b>SETUP.ALIST</b> .
ALL	Select all available channels
СРИ	Set of channels describing the CPU state (similar to the original setting of <b>DEFault</b> but no source code display).
LINE	Set of channels which contains all CPU control lines.

## List.Asm

Show disassembled mnemonics

Show disassembled mnemonics.

Shows the program address of the current instruction.

🕹 Setup 🔑	Config.	📭 Goto	iji Find	🔂 Chart 🛛 🚨 Profile	MIPS 🔷 More	🗶 Less
record						
	7	P:40000778	b	0x40000730		
	Г	P:40000730		r31,0x2		
	÷	P:40000734		0x4000077C		
_	Г	P:4000077C		r3,r13,0x7FEC		
A		P:40000780		r30,0x10(r1)		
		P:40000784 P:40000788		$r_{31,0x14(r1)}$	— В	
		P:40000788 P:4000078C		r0,0x1C(r1) r0		
		P:40000790		r1,r1,0x18		
		P:40000794		11,11,0,10		
	È	P:400012A4		0x40000798		
000000014	· ·	1.10001241	<b>M</b> 1	0x100007.50		

- A Program address column
- B Disassembled mnemonic at the address

#### List.CODE

#### Show program opcode

Shows the program opcode at the current address as hexadecimal values.

B::Trace.List List.CODE		
🔑 Setup 🔑 Config 📭 Goto	o 🛐 Find 孙 Chart 🔛 Profile 🔛 MIPS	♦ More Less
record		
1C64 2C05	adds r4,r4,#0x1 cmp r4,#0x5	^
DBF5	blt 0x20000AE	=
48F8	ldr r0,0x20000490	*
6800	ldr r0,[r0] ldr r1,[r13]	~
FB040001	mla r0,r4,r1,r0	
49F5	ldr r1,0x20000490	
6008 1C64	str r0,[r1] adds r4,r4,#0x1	
2005	cmp r4,#0x5	
DBF5	blt 0x200000AE	
48F2 6800	ldr r0,0x2000048C ldr r0,[r0]	
49F2	ldr r1.0x20000490	
-000000022		¥
<		> .::

#### List.Comment

#### Show comments to disassembled mnemonics

The mnemonic comments give information about which HLL resources are addressed in an instruction, e.g. the symbolic name of a branch target address, or the name of the variable(s) currently stored in the used register(s).

🎾 Setup 🔑 (	Config 📭	Goto 👘 Find	🕂 Chart 📕 Profile 📕 MIPS 🖨	More Less
record				
	str	r0,[r1]		~
	adds	r4,r4,#0x1		=
	cmp blt	r4,#0x5 0x20000AE	; regvar,#5	
	- ldr	r0,0x20000490		~
	ldr	r0,[r0]		1
	1dr	r1,[r13]		
	mla	r0,r4,r1,r0	; r0,regvar,r1,r0	
	1dr	r1,0x20000490		
	str	r0,[r1]		
	adds	r4,r4,#0x1		
	cmp blt	r4,#0x5 0x200000AE	; regvar,#5	
	ldr			
	ldr	r0,[r0]		
	1dr	r1,0x20000490		
000000022				~

## List.DIAG

## Disassembler-related diagnostic information

For internal use.

#### List.Dummy

## Show dummy cycles

Shows records in the List window that do not contain trace information, so-called dummy cycles. For internal use.

Adds a column to the Trace.List window, that show if a conditional instruction, e.g. a conditional branch, was executed.

	List List.EXEC							• <b>×</b>
🔑 Setup	🔑 Config 🕻	🕽 Goto	💾 Find	Chart	📕 Profile	👗 MIPS	More	Less
reco	rd							
		adds	r4,r4					~
		cmp blt	r4,#0: 0x200	000AE				≡
	I r	ldr		20000490				$\sim$
		ldr ldr mla ldr	r1,0x	13] ,r1,r0 20000490				^
	note	str adds cmp exec blt ldr	r0,[r: r4,r4 r4,#0: 0x2000 r0,0x2	,#0x1 x5				
-00000000	22	1dr 1dr	r0,[r0 r1,0x	0] 20000490				*
	<							>

#### List.HII

#### Show source code

Show the source code lines associated to the executed address-

B::Trace.List	List.HII	
Setup 2	🕈 Config 📭 Goto 🏥 Find 🚹 Chart 🛛 🧮 Profile 🖉 MIPS 🔷 More 🗌	Less
179 180	<pre>for ( regvar = 0; regvar &lt; 5 ; regvar++ )     mstatic1 += regvar*autovar;</pre>	
179	for ( regvar = 0; regvar < 5 ; regvar++ )	×
179	for ( regvar = 0; regvar < 5 ; regvar++ )	
182 -000000022	<pre>fstatic += mstatic1;</pre>	~
	<	>:

## List.Label

## Show associated label

Shows the label from the debug information that is associated to the listed address.

🎾 Setup	E Config 🔒 Goto	💾 Find	🔂 Chart 🛛 🔛 Profile	📕 MIPS 🛛 🖨 M	lore 🛛 🗶 Less
record					
		ori	r5,r5,0x9999		
		liș	r6,-0x6667		
		ori	r6,r6,0x999A		
		bl	0x40001828		
	add:	mr	r11,r1 r1,-0x40(r1)		
		stwu mflr	r1,-0x40(r1)		
		b1	0×40003018		
	savegpr_21_1:	stw	r210x2C(r11)		
	_savegpr_22_1:	stw	r220x28(r11)		
	_savegpr_23_1:	stw	r23,-0x24(r11)		
	_savegpr_24_1:	stw	r24,-0x20(r11)		
	savegpr_25_1:	stw	r25,-0x1C(r11)		
	<				>

Show disassembled mnemonics. Overrides List.ASM

📰 B::Trace.List Li	st.MIX		
Setup 🔑 ( record	Config 📭	Goto   🏂 Find   🕂 Chart   🎞 Profile   🔣 MIPS   🜩 More	Less
179	adds	<pre>for ( regvar = 0; regvar &lt; 5 ; regvar++ ) r4,r4,#0x1</pre>	^ =
179	cmp blt	<pre>for ( regvar = 0; regvar &lt; 5 ; regvar++ ) r4,#0x5 0x200000AE</pre>	~
180	ldr ldr ldr mla ldr str	<pre>mstatic1 += regvar*autovar; r0,[r0] r1,[r1] r0,r4,r1,r0 r1,0x2000490 r0,[r1]</pre>	
179	adds	<pre>for ( regvar = 0; regvar &lt; 5 ; regvar++ ) r4,r4,#0x1</pre>	
179	cmp blt	<pre>for ( regvar = 0; regvar &lt; 5 ; regvar++ ) r4,#0x5 0x200000AE</pre>	
-000000022	1dr 1dr 1dr	fstatic += mstatic1; r0,0x2000048C r0,[r0] r1,0x20000490	*
	(		>

## List.NoDummy

## Suppress the display of dummy cycles

This item is only available for backwards compatibility. Do not use. Suppressing dummy cycles is enabled by default. Hides records in the List window that do not contain trace information, so-called dummy cycles. For internal use.

## List.NoFetch

## Suppress the display of program fetches

This item is useful for bus traces and the analyzer module of the built-in instruction set simulator. Those traces have one trace record for each instruction fetch, while modern trace protocols only generate trace data for conditionally executed code.

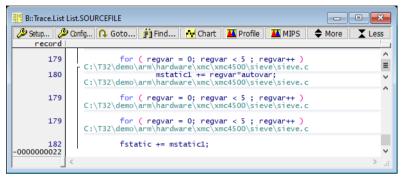
B::Trace.	List DEFault	B::Trace.	List DEFault List.NoF	etch		83
🔑 Setup 🚦	🕻 Corfig 📭 Goto 🏥 Find 🚹 Chart 🌉 Profile 🍱 MIPS 🗢 More 🗶 Less	🔑 Setup 🗄	Config 📭 Goto 🏥	Find 🚺 Chart 🔼 Prot	file 📕 MIPS 🗢 More 🗶	Less
	run address cycle data symbol			cycle da		
-000014	cmpwi r31,0x12 ; r31,18 P:FFF01200 fetch 41810050			r28,r28,0x1		
-000013	<pre>     bgt 0xFFF01250 ; 0xFFF01250 (-)     P:FFF01204 fetch 3D80FFF1     r lis r12,-0x0F ; r12,-15     </pre>		L b	r31,r31,0x1 0xFFF011FC r31,0x12		×
-000012	P:FFF01208 fetch 398Cc468 subi r12,r12,0x3B98 ; r12,r12,15256		<pre></pre>	0xFFF01250 r120x0F	; 0xFFF01250 (-)	)
-000011	P:FFF0120C fetch 7D8CF8AE lbzx r12,r12,r31		subi 1bzx		3 ; r12,r12,1525	56
-000010 -000009	D:FFF0C477 rd-byte 00 P:FFF01210 fetch 2C0C0000	-000010	cmpwi	r12,0x0	00 ; r12,0	
-000008	cmpwi r12,0x0 ; r12,0 P:FFF01214 fetch 41820034 beg 0xFFF01248 ; 0xFFF01248 (-)		+ beq Laddi	0xFFF01248 r31,r31,0x1 0xFFF011FC	; 0xFFF01248 (-) ; r31,r31,1	)
-000007	P:FFF01248 fetch 3BFF0001 r addi r31,r31,0x1 ; r31,r31,1		<pre>↓ cmpwi ↓ bgt</pre>	r 31,0x12 0xFFF01250	; r31,18 ; 0xFFF01250 (-)	~
			<		,	<b>&gt;</b>

Displays source file name for each HLL block shown in the trace listing.

## List.SOURCEFILE

#### Display associated source file path and name

Displays source file path and name for each HLL block shown in the trace listing.



#### List.sYmbol

## Show debug info for every linear program block

If this item is used, the Trace.List window will show debug symbol and offset before every linear program execution block.

🔢 B::Trace.Lis	st List.sYmbol							
🔑 Setup	E Config	🔒 Goto	🛉 Find	- Chart	🔼 Profile	K MIPS	More	Less
record								
	addi • blr	r1,r1,	0x8					^
	-main+0x1	14.						=
	bl	0xFFF0	0598					*
	-func9:	0,1110	0000					~
	stwu	r1,-0x	18(r1)					
	mflr	r0						
	stw	r30,0x	10(r1)					
	stw	r31,0x						
	stw	r0,0x1						
	lwz		x7FEC(r13)					
	stw	r12,0x						
	1i .	r31,0x						
	cmpwi	r31,0x						
	÷_bge	0xFFF0	0604					
	func9+0x		7559(-12)					
	lwz	r12,-0	x7FE8(r13)					~
	stw	F12,0X	00(11)					
								E. <

Displays information from the OS or Hypervisor Awareness. Usually information about task switches.

[B::Trace.List List.TASK DEFault /Track]	- • ×
🥔 Setup 🔑 Config 📭 Goto 👘 Find 🙌 Chart 🛛 🚨 Profile 🖉 MIPS 🗢	More 🗶 Less
record run address cycle data symbol ti.back	
TASK = 'AMX ', magic = 0001F494 0000058   D:0001E7A8 wr-long 0001F494\cj_kpdata+0x14 76.600 TASK = 'DEMO', magic = 00023994	)us ^
0000057   D:0001E7A8 wr-long 00023994\cj_kpdata+0x14 95.800 TASK = 'BKGR', magic = 000188A0	us 🗸
0000056   D:0001E7A8 wr-long 000188A0\cj_kpdata+0x14 22.700 TASK = 'BKGR', magic = 000188A0	lus ^
0000055   D:0001E7A8 wr-long 000188A0\cj_kpdata+0x14 112.200 TASK = 'BKGR', magic = 000188A0	lus
0000054   D:0001E7A8 wr-long 000188A0\cj_kpdata+0x14 114.300 TASK = 'BKGR', magic = 000188A0	lus
0000053   D:0001E7Á8 wr-long 000188A0\cj_kpdata+0x14 112.200 TASK = 'BKGR', magic = 000188A0	lus 🗸
	> .:

#### List.Tlme

## Display time information for HLL lines

Displays time information in assembler or HLL lines.

B::Trace.List									~
		Goto 👘 Find	Chart	🔼 Profile	👗 MIPS	🗢 More	Less		
record	run addres			data	sy	mbol		ti.back	
	fmov fadd fmov fcmpe b.mi	d0,#0x70 d8,d8,d0 d0,#0x14 d8,d0 0xFFFC0C2C	; d0,#5	. 0000000 . 0000000				0.139us 0.139us 0.139us 0.139us 0.139us 0.139us	
0000003711		MX:FFFC0C2C				_aarch64_	v8\sieve\func2c+0		
213	ldr fmul adrp add ldr fadd adrp add str	vdouble d0, [sp,#0x18] d1,d8,d0 x0,0xFFFC2000 x0,x0,#0x510 d0,[x0] d0,d1,d0 x0,0xFFFC2000 x0,x0,#0x510 d0,[x0]		),#1296				0.695 us 0.139us 0.162us 0.162us 0.162us 0.162us 0.162us 0.162us 0.162us 0.162us	5
212	fmov fadd fmov fcmpe b.mi	<pre>for ( regvar =</pre>	; d0,#1 ; d0,#5	var < 5.0 ; 1.0000000	-	-	v8\sieve\func2c+0;	1.458us 0.162us 0.162us 0.162us 0.162us 0.162us 2.268us	
000003700	#endif	/* NO FLOAT */	prrace			_aarch64_	volsieve (Tunc2c+0)	x00 2.268US	
	#end H	/ NO_FLOAT /							`

#### Address

#### Start address contained in trace record

The address retrieved from the trace record, which will be used for program flow and data reconstruction from this point on. The data from this column can be accesses with the function **Trace.RECORD.ADDRESS()**.

CPU

Set of items Run, Address, CYcle, Data and sYmbol

This item is a set and will be expanded to items Run, Address, CYcle, Data and sYmbol.

ſ	B::Trace.Li	st CPU /CC	DRE 0						[		×
	🔑 Setup	🔑 Config	🔒 Goto	🛐 Find	Chart	📕 Profile	👗 MIPS	More	Less		
	record	d  run  add	dress		cycle	data	syn	lod			
	-0000000543	3	M	X:FFFC14EC	ptrace					nc10+0x318	~
	-0000000533	3	M	X:FFFC14E4	ptrace					nc10+0x310	
	-0000000528			X:FFFC14E4						nc10+0x310	
	-0000000512			X:FFFC14E4						nc10+0x310	~
	-0000000508			X:FFFC14F4						nc10+0x320	
	-0000000488		M	X:FFFC2094	ptrace					main+0x344	$\sim$
	-0000000467			X:FFFC1500						eve\func11	
	-0000000462	2		X:FFFC1518						unc11+0x18	
	-0000000438			X:FFFC15B0						unc11+0xB0	
	-0000000430			X:FFFC209C						main+0x34C	
	-000000360			X:FFFC15E0						eve\func13	
	-000000350			X:FFFC1638						unc13+0x58	
	-0000000292	2		X:FFFC15E0						eve\func13	
	-0000000270			X:FFFC1638						unc13+0x58	
L	-0000000179		M	X:FFFC15E0	ptrace					eve\func13	
	-0000000170	6 BRK —	M	X:FFFC1630			m_	_aarch64_v	8\sieve\f	unc13+0x50	~
L		<								>	
L	1										

CYcle

#### Show bus cycle type of trace record

Shows the bus cycle type that is stored in the trace record.

B::Trace.l	List Address Cycle [	Data sYmbol					[	- • ×
🔑 Setup	🏭 Config 🕻	Goto	🛉 Find	Chart	🔼 Profile	MIPS	More	Less
record			data	symbol				
-000150	D:FFF0C434			\\diabp4	\diabp4\	vbfield+0x00	2	^
-000149	P:FFF00570		8001000C	\\diabp4	\diabp4\	func8+0x1F4		≡
-000148	D:FFF0FF84		FFF010D8	\\diabp4	\Global\.	SP_END+0	xF84	
-000147	P:FFF00580		7C0803A6	\\diabp4	\diabp4\	func8+0x1F8		*
-000146	P:FFF00584		38210008	\\diabp4	\diabp4\	func8+0x1FC		^
-000145	P:FFF00588		4E800020	\\diabp4	\diabp4\	func8+0x200		
-000144	P:FFF010D8		4BFFF4C1	\\diabp4	\diabp4\	main+0x114		
-000143	P:FFF00598		9421FFE8	\\diabp4	\diabp4\	func9	_	
-000142	D:FFF0FF68		FFF0FF80	\\diabp4	\Global\.	SP_END+0	xF68	
-000141	P:FFF00590		7C0802A6					
-000140	P:FFF005A0		93C10010	\\diabp4	\diabp4\	func9+0x8		
-000139	D:FFF0FF78		FFF0C338	\\diabp4	\Global\.	SP_END+0	xF78	
-000138	P:FFF005A4		93E10014	\\diabp4	\diabp4\	func9+0x0C	_	
-000137	D:FFF0FF70		0000000B	\\diabp4	\Global\.	SP_END+0	xF7C	
-000136	P:FFF005A8		9001001C	\\diabp4	\diabp4\	func9+0x10		
-000135	D:FFF0FF84	wr-long	FFF010DC	\\diabp4	\Global\.	SP_END+0	xF84	*
	<							>:

Some common cycle types are listed below.

ptrace	The trace record contains information about the program flow.
fetch	The trace record contains information about instruction fetches.
read rd- <width></width>	The trace record contains information about a data read access.
write wr- <word></word>	The trace record contains information about a data write access.
owner	The trace record contains information about the trace ownership. This cycle type typically used to signal task switches.

#### Run

## Core and execution information

- GO: the first instruction that was executed by the CPU after starting program execution with Go. BRK Indicates that the program execution was stopped.
- T : Indicates a trigger event.
- f : Foreground program
- b : Background program
- ft : Trigger event occurred in the foreground program
- bt: Trigger event occurred in the background program
- 0,1,2,3 ... in SMP systems, the run column indicates the number of the core that executed the given code; additionally, the background color of the records changes to high-light the relevant core (light red, light green, ...).

Shows the state of the trace port signals, as recorded by the trace module. The names of the individual signals depend on the trace port protocol. In some cases, the recorded trace port signals may not directly correlate to physical trace port signals. Intended for low-level diagnostic purposes.

B::Trace.L	ist SIGN/	ALS																					x
🔑 Setup	🔑 Confi	g 🖪	Goto	🧯	Find.	. A	Chart	. 🔼	Profile	🔼 МІ	IPS	:	<b>\$</b> M	ore		X	Less						
		swo1	swo2	swo3	swo4	swo5	swo6	swo7	swostop	clk	d0	d1	d2	d3	d4	d5	d6	d7	tp0	tp1	tp2	tp3	
-000000001																							~
-000000001										_													-
-000000001										clk													≡
-000000001			swo2	swo3	swo4	swo5	swo6	swo7		c1k			d2	d3	d4	d5	d6			tp1	tp2	tp3	$\sim$
-000000001								_			d0								tp0				~
-000000001								swo7										d7					~
-000000001																							
-000000001																							
-000000000																							
-000000000																							
-000000000					swo4										d4								
-000000000		swo1	ewo2	ewo3		swo5	swo6	swo7			do	d1	d2	d3		ds	d6	d7	± n0	tn1	tp2	±n3	
-000000000											do	di	d2	da		ds	de	d7	tn0	tn1	tp2	tn3	
-000000000											do	di	d2	d3	d4	ds	dő	d7	th	tn1	tp2	tn3	
-000000000																	dő					tp3	
-000000000								swo7		c1k													
-000000000																						tp3	$\mathbf{v}$
	<																				-		
1	ш.°																						> .d

#### sYmbol

## Debug symbol of start address in trace record

This item shows the debug symbol and offset, which is associated to the address of the Address item.

B:: Trace.List Address sYmbol	×
🥬 Setup 🔢 Config 📭 Goto 👘 Find 🔥 Chart 🛛 🗮 Profile 🖉 MIPS 🗢 More 🗶 Less	s
record address symbol	
-000050 P:FFF00078 \\diabp4\diabp4\func1+0x20	~
-000049 D:FFF0FF64 \\diabp4\Global\SP_END+0xF64	≡
-000048 P:FFF0007C \\diabp4\diabp4\func1+0x24	
-000047 D:FFF0FF6C \\diabp4\Global\SP_END+0xF6C	~
-000046 P:FFF00080 \\diabp4\diabp4\func1+0x28	^
-000045 P:FFF00084 \\diabp4\diabp4\func1+0x2C	
-000044 P:FFF00088 \\diabp4\diabp4\func1+0x30	
-000043 P:FFF005EC \\diabp4\diabp4\func9+0x54	
-000042 P:FFF005F0 \\diabp4\diabp4\func9+0x58	
-000041 P:FFF00058 \\diabp4\diabp4\func1	
-000040 D:FFF0FF58 \\diabp4\Global\SP_END+0xF58	
-000039 P:FFF0005C \\diabp4\diabp4\func1+0x4	- <b>1</b> 12
-000038 P:FFF00060 \\diabp4\diabp4\func1+0x8	
-000037 D:FFF0FF64 \\diabp4\Global\SP_END+0xF64	~
	>

The function of the CLOCKS item group is similar to the function of the TIme item group, with the difference that the measurement unit is core clock ticks instead of seconds.

CLOCKS and TIME are closely related. In the case of a tool-generated timestamp, the base data is TIME and CLOCKS will be derived from it. In the case of processor-generated timestamps (e.g. cycle-accurate trace), the base data is in CLOCKS and TIME is derived from it.

The conversion from CLOCKS to TIME depends on the setting of the command Trace.CLOCK.

	🎾 Config 🔃 Goto 🎁 Find 🚺 Chart 🛛 🌉 Profile 🛛 🌉 MIPS 🛛 🗢 More 🛛 👗 Les				-
cord 720	vtriplearray[1][0][0] = 2;	ti.back 0.397us	clocks.b 99.	<u>clocks.z</u> -323.	ref -1090
721	vtriplearray[0][1][0] = 3;	0.397us	99.	-224.	-1050
722	vtriplearray[0][0][1] = 4;	0.397us	99.	-125.	-892
724	func2();	0.397us	100.	-25.	-792
153	void func2(void)	0.100us	25.	0.	-767
	int autovar; register int regvar; static int fstatic = 44; /* initialized static variable * static int fstatic2; /* not initialized static variab	/ le */			
159	autovar = regvar = fstatic;	0.767us	191.	191.	-576
160	autovar++;	1.023us	256.	447.	-320
162	<pre>func1( &amp;autovar ); /* to force autovar as stack-scope */</pre>	0.767us	192.	639.	-128
	<pre>static void func1( int * intptr ) /* static function */</pre>	0.512	120		
148 149	(*intptr)++;	0.512us 0.617us	128. 154.	767.	0 154
150		1.542us	385.	1306.	539
163	<pre>func1( &amp;fstatic ); /* to force fstatic as static-scope */</pre>	0.926us	232.	1538.	771
	<pre>static void func1( int * intptr ) /* static function */</pre>				
<					
		•			

- A Number of clocks relative time to previous record
- B Number of clocks relative to global zero point
- C Number of clocks relative to reference point

#### CLOCKS.Back Number of clocks relative time to previous record

This item displays the number of core clock ticks that passed since the previous record.

#### **CLOCKS.Fore**

Number of clocks relative time to next record

This item displays the number of core clock ticks that passed between this and the upcoming record.

This item displays the number of core clock ticks that passed since the trace reference point. See **Trace.REF** for more information.

## CLOCKS.Trigger

#### Number of clocks relative to trigger point

This item displays the number of core clock ticks that passed since the trigger point in the trace recording. The trace record with the Trigger is record number 0 and is marked with letter 'T'.

A trigger can be issued by TRACE32 logic analyzers, complex trigger units and also external sources. See **TrBus**, **Trace.TCount** and **Trace.TDelay** for more information.

🖉 Setup 🌽	🖓 Config 🔒	Goto 🛐 Find	Chart	🔼 Profile	🔼 MIPS	More	Less				
	run addres		cycle		S	mbol			ti.back	clocks.t	
433	1000	for ( v14 = 0 ;									
	add	w20,w20,#0x1	; 1,1,4	#1							
	cmp b.le	w20,#0x2	; i,#2								
000000021	A D. Ie	0xFFFC14B4 MX:FFFC14B4	ataaca			anab64 w		nc10+0x2E0	1.188us	-297.	
434		v17 +=				aar crio4_vo	s/sieve/iu	ICTO+0X2E0	1.10005	-297.	
434	add		: v17.	17.i							
433	uuu	for ( v14 = 0 ;									
	add	w20,w20,#0x1	: 1.1.4								
	cmp	w20,#0x2	; i,#2								
	b.le	0xFFFC14B4									
00000000000		MX:FFFC14B4	ptrace			_aarch64_v8	3\sieve\fu	nc10+0x2E0	1.188us	0.	
434	Γ	v17 +=									
433	add	w19,w19,w20	; v17,v								
433	add	for (v14 = 0; w20.w20.#0x1									
	Cmp	w20, #0x1	; i,i,4	+ <b>⊥</b>							
	b, le	0xFFFC14B4	1 1,#2								
+0000000005	0.10	MX:FFFC14C4	ptrace			aarch64 v8	sieve\fu	nc10+0x2F0	1.188us	297.	
435		for ( v15 = 0 ;		: v15++ )							
	mov	w20,#0x0	; i,#0	e e							
	∱ þ	0xFFFC14D4									
	r cmp	w20,#0x2	; i,#2								
	b.le	0xFFFC14CC									
+000000010		MX:FFFC14CC	ptrace			_aarch64_v8	sieve\fu	nc10+0x2F8	0.652us	460.	

## CLOCKS.Zero

#### Number of clocks relative to global zero point

This item displays the number of core clock ticks that passed since the ZERO point in the trace recording. For tool-generated timestamps, the ZERO point default is set to the start of the first debug session since starting PowerView.

For trace sources without tool-generated timestamps (like on-chip trace, but also trace data loaded with **Trace.FILE**), the zero point can be either at the start or end of the trace recording.

It is possible to align the ZERO time of unrelated trace sources. See command ZERO for details.

The Data items shows any data that is stored in a trace record. The data is usually data from read/write accesses, but can also contain fetched program code, depending on the CYcle type.

🔡 B::Trace.L	ist Address CYcle D	ata Data.B0 [	Data.B1 Data.B	2 Da	ta.B3					[		×
🔑 Setup	. 🏭 Config 🕻	Goto	🛐 Find		Chart		Profile	🔼 MI	PS	\$ More	🗶 Less	
	address	cycle	data		b1 b	2  b	3					
-000928	P:FFF00FF4		996CC454	54	C4 6	C 9	9					~
-000927	D:FFF0C454					0	3					≡
-000926	P:FFF00FF8		3D40FFF1	F1	FF 4	03	D					-
-000925	P:FFF00FFC		39200004		00 2	03	9					~
-000924	P:FFF01000			51	C4 2	A 9	9					$\sim$
-000923	D:FFF0C451				0							
-000922	P:FFF01004					F 4	B					
-000921	P:FFF00098					19						
-000920	D:FFF0FF70					0 F	F					
-000919	P:FFF0009C					87	C					
-000918	P:FFF000A0					19						
-000917	D:FFF0FF7C					00	0					
-000916	P:FFF000A4					19						
-000915	D:FFF0FF84											
-000914	P:FFF000A8	fetch	83ED8010	10	80 E	D 8	3					$\sim$
	<											>

#### Data.any

## Data value stored in trace record (auto width)

Shows the data associated to the current trace record in the suitable width for the processor architecture in use.

## **Data.B<x>** Data value stored in trace record (byte lanes)

Shows the data associated to the current trace record for the specified byte lane. E.g. for a 32-bit architecture, the allowed byte lane numbers are 0, 1, 2 and 3. For 8-bit architectures, the Data.B item is used without index.

#### Data.L<x>

## Data value stored in trace record (long lanes)

Shows the data associated to the current trace record for the specified long (32-bit) lane. If there is only one lane, Data.L is used without index.

## Data.Oct

## Data value stored in trace record (oct-word)

Shows the data associated to the current trace record as oct-word (128 bit value).

Shows the data associated to the current trace record for the specified quad (64-bit) lane. If the architecture does not support more than 64-bit wide accesses, Data.Q is used without index.

#### Data value stored in trace record (tbyte)

Shows the data associated to the current trace record for the specified TBYTE (24-bit) lane. If there is only one lane, Data.T is used without index.

#### Data.W<x>

#### Data value stored in trace record (word)

Shows the data associated to the current trace record for the specified word (16-bit) lane. If there is only one lane, Data.W is used without index.

#### Data.sYmbol

#### Debug symbol associated to data value

If the data stored in the current trace record is an address, then this item will show the debug symbol that is associated to the address stored in the data field.

This item can e.g. show debug symbol of addresses that are written to or read from the program stack.

🔑 Setup	. 🔡 Config 🖡	Goto	🛉 Find	- Chart	📕 Profile	MIPS	More	🗶 Less			
record a		cycle	data	symbol				symb			
-000964	P:FFF01384		3FC0FFF1							init_main+0x9C	
-000963	P:FFF01388		3BDEC 338							init_main+0xA0	
-000962	P:FFF0138C		03008000							init_main+0xA4	
-000961	D:FFF0C3D4		FFF0C338	\\diabp4\	Global\_	_env		\\di	abp4\Global	\environ	. 1
-000960	P:FFF01390		TOUDECDJ							init_main+0xA8	1
-000959	P:FFF04048		7C0802A6						abp4\G]oba]		
-000958	P:FFF0404C			V 12 1 4V	c1   1\		0 = 1 0			\init+0x4	
-000957	D:FFF0FF6			\\d1abb4\	Global\_	SP_END+	UXFA8			SP_END+0xF68	•
-000956	P:FFF0405(			) ) dd - b - 4)	Clabal)		0.10			\init+0x8	
-000955	D:FFF0FFAC P:FFF04054		FFF01394	\\d1abp4\	GIODAI\_	_1n1t_ma1	n+0xac			SP_END+0xFAC	e
-000954 -000953			FFF01204	\\ dd alaw (\	clahal)	2				\init+0x0C \SP_END+0xFAC	
-000955	D:FFF0FFA( P:FFF04056			\\dTabb4\	GTODAT\_	_Init_mai	n+UXAC			\SP_END+0xFAC	1
-000952	P:FFF04056		7C0803A6							$\underline{10111+0x10}$	
-000950	P:FFF04060		4E800020					\\di	abp4 (Global	$\underline{1011}$	

The ENERGY item group is used to show the energy measurement results in the Trace.List window. For more information, check the **ETA** command group.

## **ENERGY.Abs** Energy consumption since start of trace data

This item shows the energy consumption from the oldest trace record in the trace buffer until the current record.

#### **ENERGY.Back** Energy consumption since previous record

This item shows the relative energy consumption since the last trace record.

#### ENERGY.Fore

Energy consumption until next record

This item shows the relative energy consumption until the next trace record.

# **ENERGY.REF** Energy consumption since reference record

Energy consumption since the trace reference point. See **Trace.REF** for more information.

# **ENERGY.Trigger** Energy consumption since trigger point

Energy consumption since the since the trigger point in the trace recording. The trace record with the Trigger is record number 0 and is marked with letter 'T'.

A trigger can be issued by TRACE32 logic analyzers, complex trigger units and also external sources. See **TrBus**, **Trace.TCount** and **Trace.TDelay** for more information.

#### ENERGY.Zero

#### Energy consumption since global zero point

Energy consumption since the ZERO point in the trace recording. For tool-generated timestamps, the ZERO point default is set to the start of the first debug session since starting PowerView.

For trace sources without tool-generated timestamps (like on-chip trace, but also trace data loaded with **Trace.FILE**), the zero point can be either at the start or end of the trace recording.

It is possible to align the ZERO time of unrelated trace sources. See command ZERO for details.

The function of the TIme item group is similar to the function of the **CLOCKS** item group, with the difference that the measurement unit is seconds instead of core clock ticks.

TIME and CLOCKS are closely related. In the case of a tool-generated timestamp, the base data is TIME and CLOCKS will be derived from it. In the case of processor-generated timestamps (e.g. cycle-accurate trace), the base data is in CLOCKS and TIME is derived from it.

The conversion from CLOCKS to TIME depends on the setting of the command Trace.CLOCK.

B::Trace.List DEFault Time.Zero Time.REF /CORE 0				x
🥔 Setup 🔑 Config 😱 Goto 🏥 Find 🔥 Chart 🗮 Profile 🔛 MIPS 🔷 More 🗶 Less				
record run address     cycle     data     symbol       719     vtriplearray[0][0][0] = 1;	ti.back 0.927us	ti.zero -1.972us	ti.ref -5.060us	1
720 $vtriplearray[1][0][0] = 2;$	0.463us		-4.596us	^
721 vtriplearray[0][1][0] = 3;	0.463us		-4.133us	=
722 vtriplearray[0][0][1] = 4;	0.463us	-0.581us	-3.669us	~
724 func2();	0.463us	-0.118us	-3.206us	^
void func2(void)				
153 {	0.118us	0.000us	-3.088us	
int autovar; register int regvar;				
static int fstatic = 44: /* initialized static variable */				
static int fstatic2; /* not initialized static variable */	r -			
159 autovar = regvar = fstatic;	0.772us	0.772us	-2.316us	
160 autovar ++ egyar = istatic;	1.029us		-1.287us	-
				-
162 func1( &autovar ); /* to force autovar as stack-scope */	0.772us	2.573us	-0.515us	
<pre>static void func1( int * intptr ) /* static function */</pre>				
148 f	0.515us	3.088us	0.000us	
149 (*intptr)++;	0.598us		0.598us	
150 } 163 func1( &fstatic ); /* to force fstatic as static-scope */	1.496us	5.182us 6.080us	2.094us 2.992us	
163 r funct( &rstatic); /* to force fstatic as static-scope */	0.898us	6.080us	2.992us	- v
		<u> </u>		•
	<b>▲</b>	<b></b>	<b></b>	
	A	В	C	

- **A** Time relative to previous record
- B Time relative to global zero point
- **C** Time relative to reference point

## **TIme.AddressBack**Time relative to previous occurrence of address

Usage: Trace.List DEFault TIme.AddressBack <address>

This item shows the time that has passed since the specified address was last executed.

Time relative to previous record

Usage: Trace.List DEFault TIme.AddressFore <address>

This item shows the time that will pass until the specified address will be executed next.

## TIme.Back

This item displays the time that passed since the previous record.

TIme.Fore

#### Time relative to next record

This item displays the time that passed between this and the upcoming record.

#### TIme.FUNC

#### Time spent to execute a function

This item shows the time that the core executed the current function, including external code (e.g. function calls).

	ònfig 📭 Goto 🏥 Find 🙌 Chart 🛛 🌉 Profile 🔛 MIPS 🔷 More 🗶 Less		
record ru	n address cycle data symbol	ti.back	ti.func
	void func2(void)		
153		0.119us	30.016u
	int_autovar;		
	register int regvar;		
	<pre>static int fstatic = 44;  /* initialized static variable */ static int fstatic2;  /* not initialized static variable */</pre>		
	static menstaticz, / not interatized static variable /		
159	autovar = regvar = fstatic;	0.889us	
160	autovar++;	1.185us	
162	<pre>func1( &amp;autovar ); /* to force autovar as stack-scope */</pre>	0.889us	
	<pre>static void func1( int * intptr )</pre>		
148	static volument includer ) / static function /	0.593us	2.936u
149	(*intptr)++;	0.587us	
150		1.468us	
163	<pre>func1( &amp;fstatic ); /* to force fstatic as static-scope */</pre>	0.881us	
	<pre>static void func1( int * intptr ) /* static function */</pre>		
148	(	0.628us	2.848u
149	(*intptr)++;	0.570us	
150	}	1.424us	

## TIme.FUNCEX

## Time spent outside the current function

This item shows the time that the core was executing code outside the current function.

This item shows the time that the core executed the current function, excluding external code (e.g. function calls).

## TIme.MARK<x>BACK

#### Time relative back to the last marker

This item shows the time that passed since the specified marker last appeared in the trace recording. Valid marker names are A, B, C and D.

## TIme.MARK<x>FORW

#### Time relative forward to the next marker

This item shows the time that passed from the current record to the record in which the specified marker appears next in the trace recording. Valid marker names are A, B, C and D.

#### TIme.REF

Time relative to reference point

This item displays the time that passed since the trace reference point. See **Trace.REF** for more information.

#### Tlme.Trigger

Time relative to trigger point

This item displays the time that passed since the trigger point in the trace recording. The trace record with the Trigger is record number 0 and is marked with letter 'T'.

A trigger can be issued by TRACE32 logic analyzers, complex trigger units and also external sources. See **TrBus**, **Trace.TCount** and **Trace.TDelay** for more information.

Setup 2	🌽 Config 🛛	🗘 Goto 🎁 Find	Chart	K Profile	MIPS	More	Less				
record	run addr		cycle	data	syn	nbol		1	ti.back	ti.trigger	
-000000032 434	L b. le	0xFFFC1484 MX:FFFC1484 v17 += w19,w19,w20				aarch64_v8	\sieve\fu	inc10+0x2E0	1.200us	-2.376us	
433	add cmp b.le	<pre>for ( v14 = 0 ; w20,w20,#0x1 w20,#0x2</pre>	v14 < 3	; v14++ )		aarch64_v8	\sieve\fu	inc10+0x2E0	1.188us	-1.188us	
434 433	add add cmp	v17 += w19,w19,w20 for (v14 = 0; w20,w20,#0x1 w20,#0x2	v14; v14 < 3	v17,i ; <b>v14++ )</b> #1	-	_					
T000000000 434	L b.le	MX:FFFC14B4 v17 +=				aarch64_v8	\sieve\fu	inc10+0x2E0	1.188us	0.000us	
433	add cmp b.le	<pre>for ( vi4 = 0 ; w20,w20,#0x1 w20,#0x2 0xFFFC1484</pre>	v14 < 3	; v14++ )							
+000000005		MX:FFFC14C4 for (v15 = 0 :		· v15++ )		aarch64_v8	\sieve\fu	inc10+0x2F0	1.188us	1.188us	
433		101 ( 115 - 0 ,	112 ( )	, , ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,							

#### Tlme.Zero

## Time relative to global reference

This item displays the time that passed since the ZERO point in the trace recording.

For tool-generated timestamps, the ZERO point default is set to the start of the first debug session since starting PowerView. For trace sources without tool-generated timestamps (like on-chip trace, but also trace data loaded with **Trace.FILE**), the zero point can be either at the start or end of the trace recording.

It is possible to align the ZERO time of unrelated trace sources. See command ZERO for details.