

Application Note for Trace.List

Release 09.2023

Application Note for Trace.List

TRACE32 Online Help

TRACE32 Directory

TRACE32 Index

TRACE32 Documents 

Trace Application Notes 

Trace Analysis 

Application Note for Trace.List **1**

Overview **4**

List items 4

Record items 5

Pre-defined Item Groups 5

Item reference: List items **6**

List.Asm Show disassembled mnemonics 6

List.ADDRESS Show program address 7

List.CODE Show program opcode 7

List.Comment Show comments to disassembled mnemonics 7

List.DIAG Disassembler-related diagnostic information 8

List.Dummy Show dummy cycles 8

List.EXEC Execution information about conditional branches 9

List.Hll Show source code 9

List.Label Show associated label 9

List.Mix Show disassembled mnemonics and HLL code 10

List.NoDummy Suppress the display of dummy cycles 10

List.NoFetch Suppress the display of program fetches 10

List.SOURCE Display associated source file name 11

List.SOURCEFILE Display associated source file path and name 11

List.sYmbol Show debug info for every linear program block 11

List.TASK Display OS Awareness information 12

List.Tlme Display time information for HLL lines 12

Item reference: Record items **13**

Address Start address contained in trace record 13

CPU Set of items Run, Address, CYcle, Data and sYmbol 13

CYcle Show bus cycle type of trace record 13

Run Core and execution information 14

SIGNALS Trace-port signals 15

sYmbol Debug symbol of start address in trace record 15

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

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”.

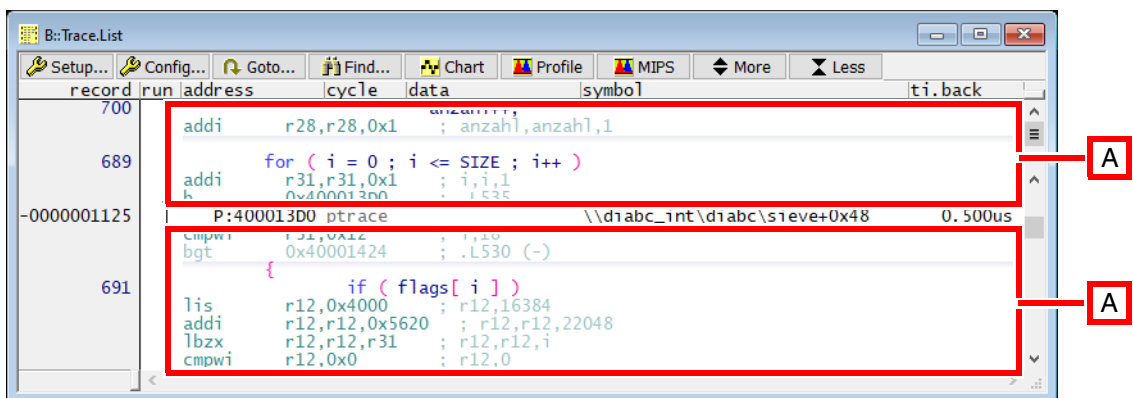
When a trace recording is processed, the resulting information about the trace messages and of the reconstructed program flow is displayed in various windows, e.g. in [Trace.List](#) or [Trace.FindAll](#) windows. Trace.List items are the pieces of information of a certain type, which represent certain 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:

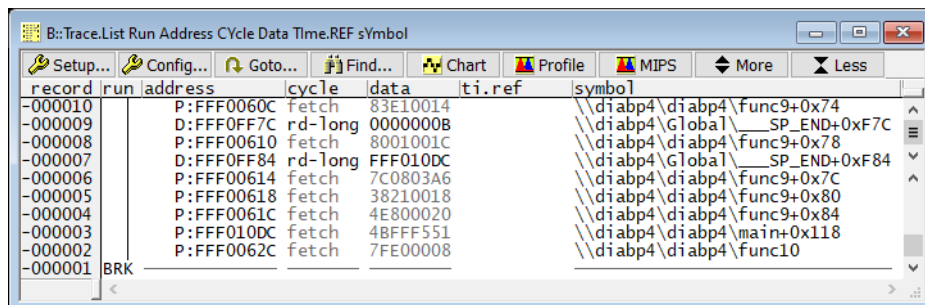


A List Items

Record 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:



The screenshot shows a window titled "B::Trace.List Run Address Cycle Data Time.REF sYmbol". The window contains a table with the following columns: record, run, address, cycle, data, ti.ref, and symbol. The data is as follows:

record	run	address	cycle	data	ti.ref	symbol
-000010		P:FFF0060C	fetch	83E10014		\\diabp4\diabp4\func9+0x74
-000009		D:FFF0FF7C	rd-long	0000000B		\\diabp4\Global__SP_END+0xF7C
-000008		P:FFF00610	fetch	8001001C		\\diabp4\diabp4\func9+0x78
-000007		D:FFF0FF84	rd-long	FFF010DC		\\diabp4\Global__SP_END+0xF84
-000006		P:FFF00614	fetch	7C0803A6		\\diabp4\diabp4\func9+0x7C
-000005		P:FFF00618	fetch	38210018		\\diabp4\diabp4\func9+0x80
-000004		P:FFF0061C	fetch	4E800020		\\diabp4\diabp4\func9+0x84
-000003		P:FFF010DC	fetch	4BFFF551		\\diabp4\diabp4\main+0x118
-000002		P:FFF0062C	fetch	7FE00008		\\diabp4\diabp4\func10
-000001	BRK					

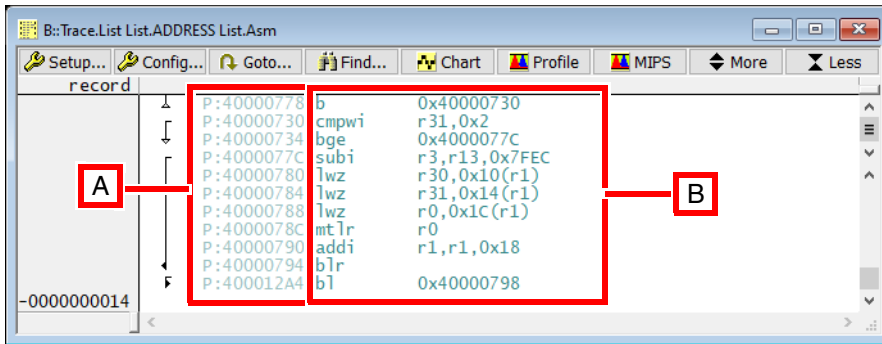
Pre-defined Item Groups

The following pre-defined item groups are available:

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

Show disassembled mnemonics.

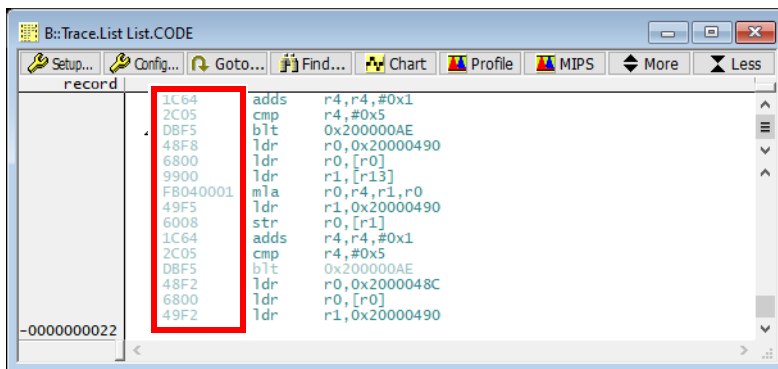
Shows the program address of the current instruction.



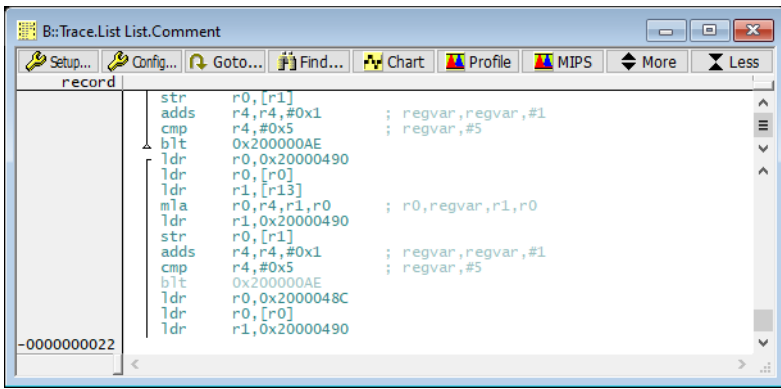
A Program address column

B Disassembled mnemonic at the address

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



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).



List.DIAG

Disassembler-related diagnostic information

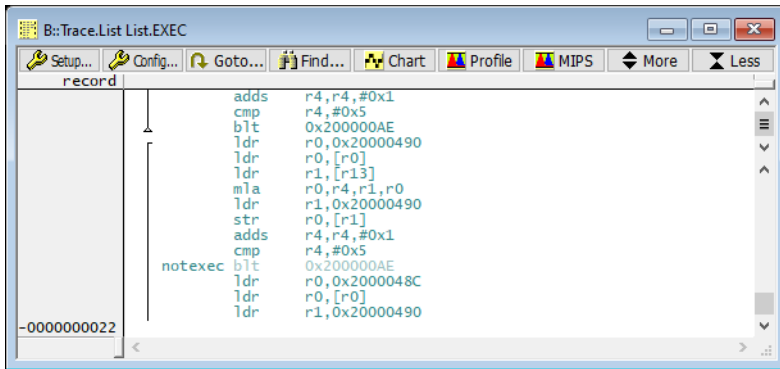
For internal use.

List.Dummy

Show dummy cycles

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

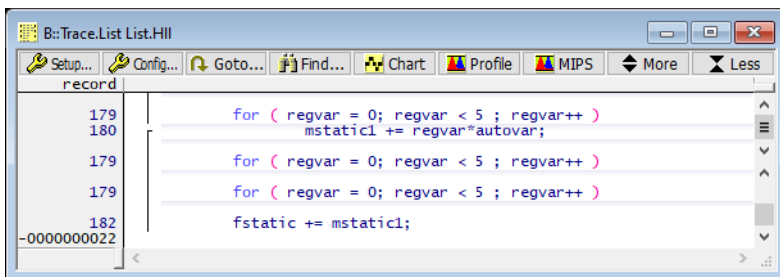
Adds a column to the Trace.List window, that show if a conditional instruction, e.g. a conditional branch, was executed.



List.HLL

Show source code

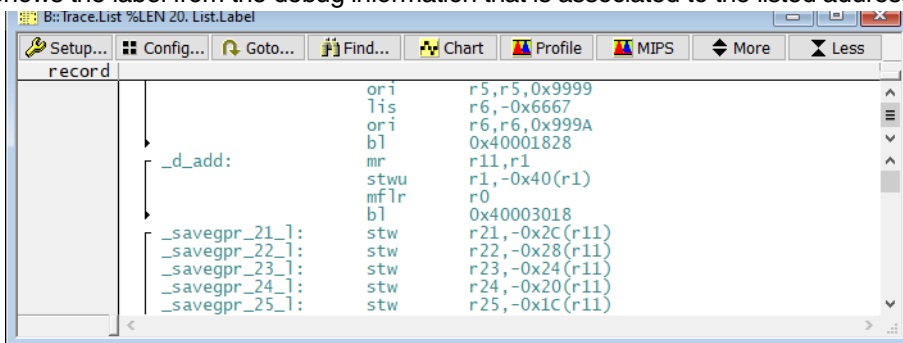
Show the source code lines associated to the executed address-



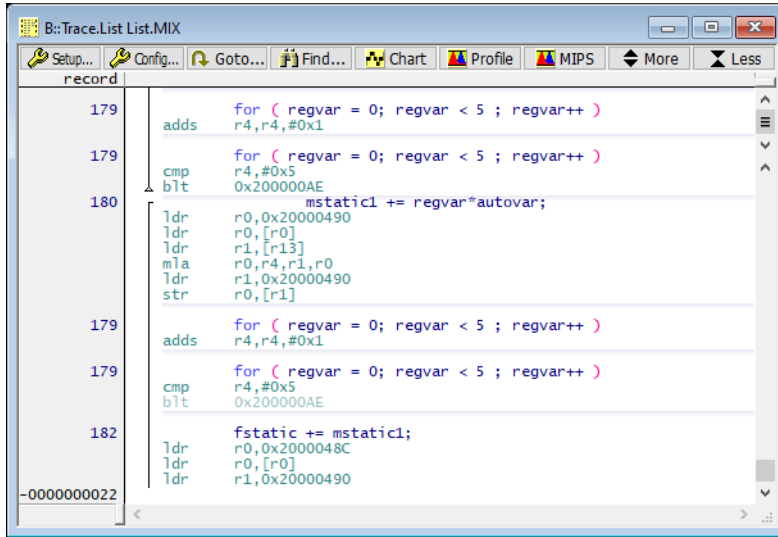
List.Label

Show associated label

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



Show disassembled mnemonics. Overrides List.ASM



List.NoDummy

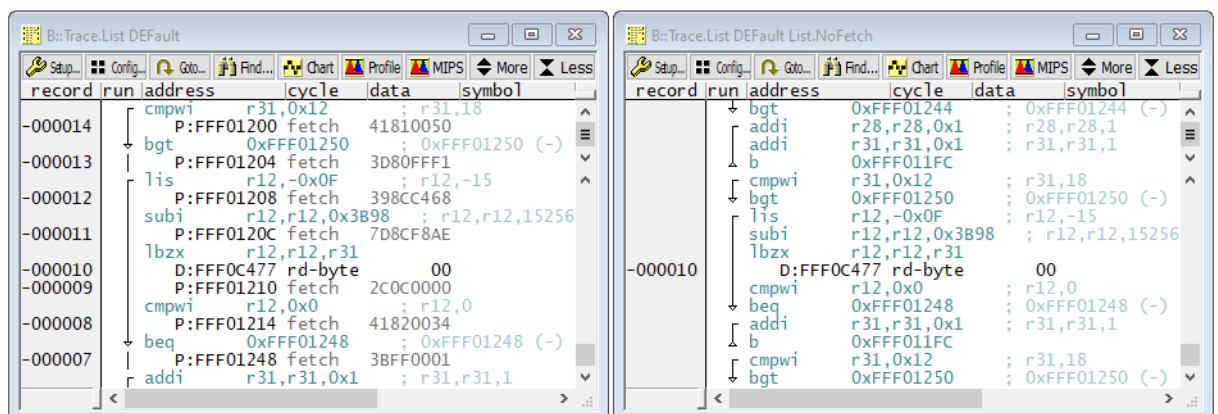
Suppress the display of dummy cycles

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.



List.SOURCE

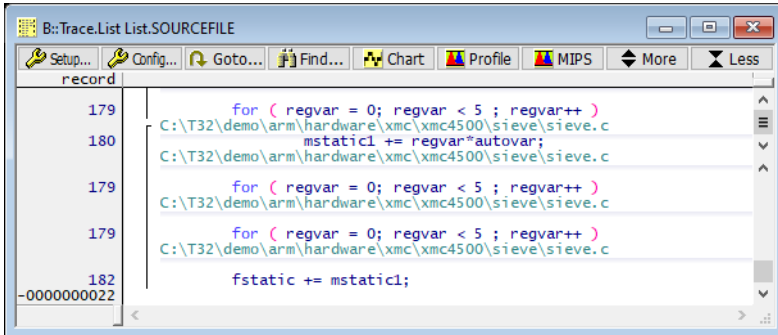
Display associated source file name

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

List.SOURCEFILE

Display associated source file path and name

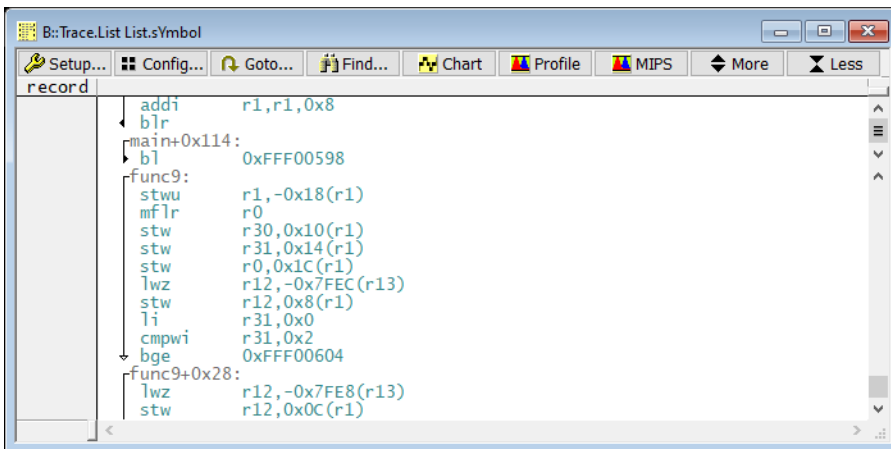
Displays source file path and name for each HLL block shown in the trace listing.



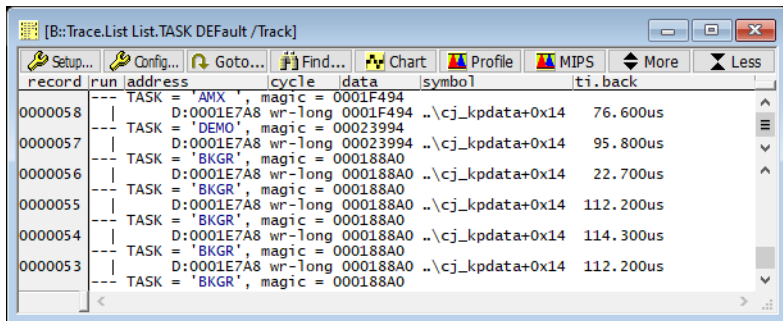
List.sYmbol

Show debug info for every linear program block

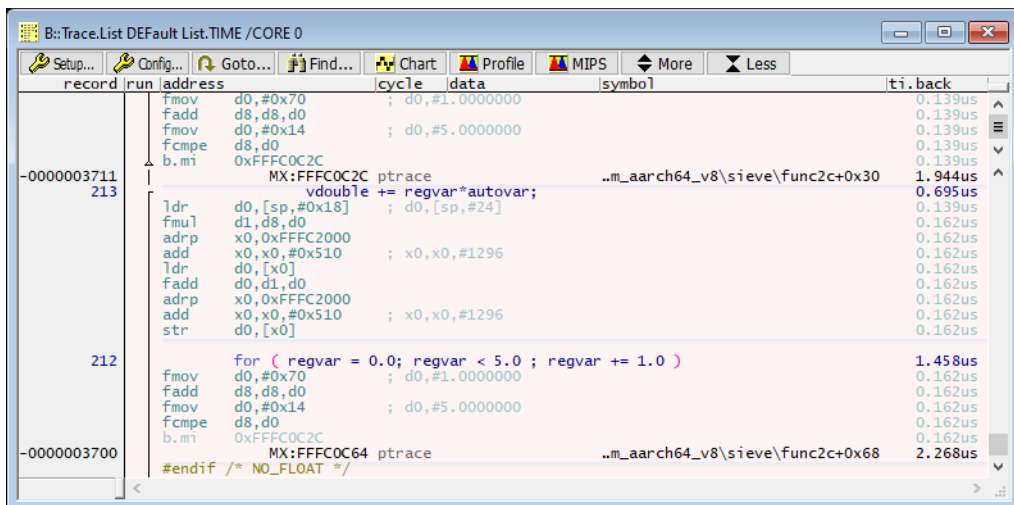
If this item is used, the Trace.List window will show debug symbol and offset before every linear program execution block.



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



Displays time information in assembler or HLL lines.



Item reference: Record items

Address

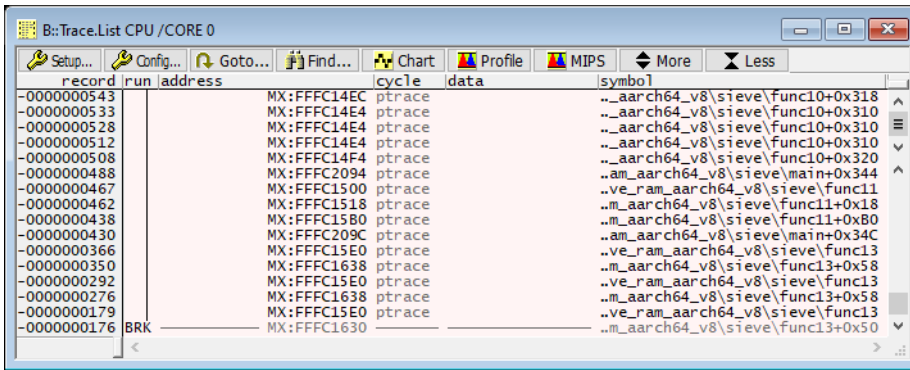
Start address contained in trace record

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 accessed with the function [Trace.RECORD.ADDRESS\(\)](#).

CPU

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

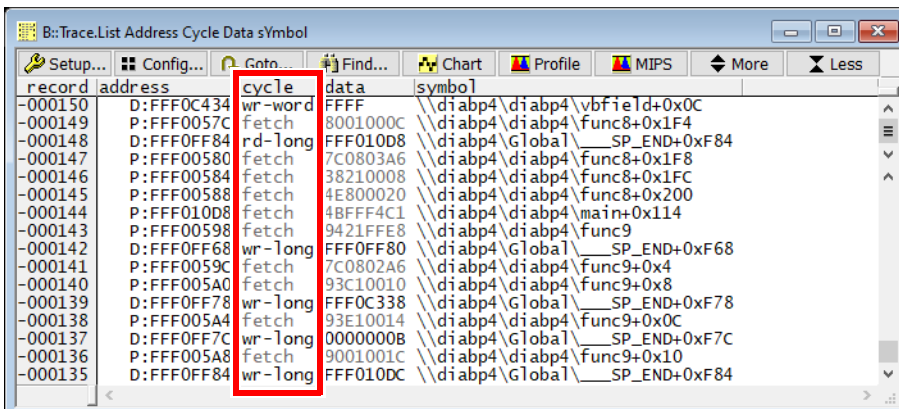
This item is a set and will be expanded to items Run, Address, CYcle, Data and sYmbol.



CYcle

Show bus cycle type of trace record

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



Some common cycle types are listed below.

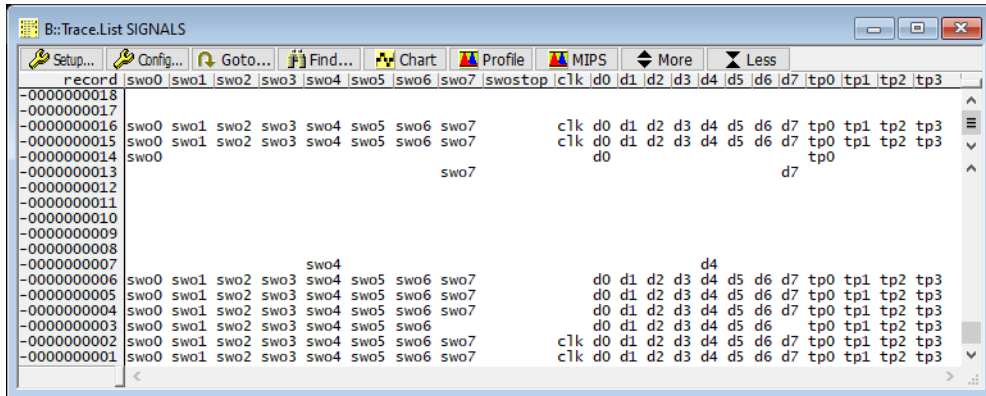
ptrace	The trace record contains information about the program flow.
fetch	The trace record contains information about instruction fetches.
read rd-<width>	The trace record contains information about a data read access.
write wr-<word>	The trace record contains information about a data write access.
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, ...).

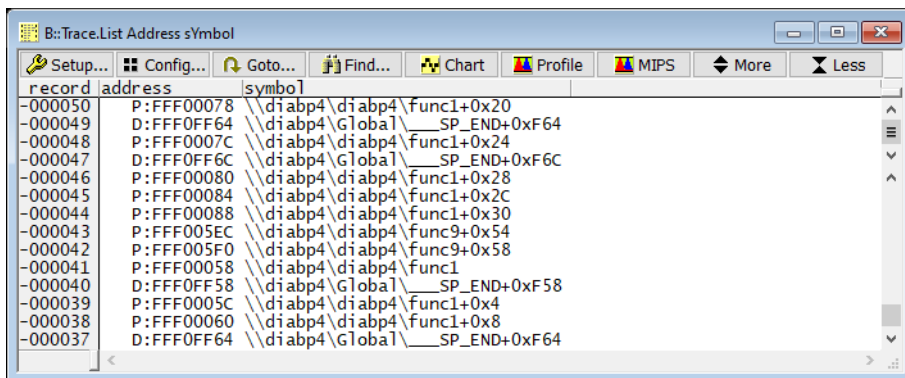
Shows the state of the trace port signals, as recorded by the trace module. The names of the individual signals depend on the trace port protocol. In some cases, the recorded trace port signals may not directly correlate to physical trace port signals. Intended for low-level diagnostic purposes.



sSymbol

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.



Item reference: CLOCKS items

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**.

record		ti.back	clocks.b	clocks.z	ref
720	vtriplearray[1][0][0] = 2;	0.397us	99.	-323.	-1090.
721	vtriplearray[0][1][0] = 3;	0.397us	99.	-224.	-991.
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 variable */				
159	autovar = regvar = fstatic;	0.767us	191.	191.	-576.
160	autovar++;	1.023us	256.	447.	-320.
162	func1(&autovar); /* to force autovar as stack-scope */	0.767us	192.	639.	-128.
148	static void func1(int * intptr) /* static function */	0.512us	128.	767.	0.
149	{	0.617us	154.	921.	154.
150	(*intptr)++;	1.542us	385.	1306.	539.
163	func1(&fstatic); /* to force fstatic as static-scope */	0.926us	232.	1538.	771.
	static void func1(int * intptr) /* static function */				

- 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.

record	run	address	cycle	data	symbol	ti.back	clocks.t
433		for (v14 = 0 ; v14 < 3 ; v14++)					
		add w20,w20,#0x1 ; i,i,#1					
		cmp w20,#0x2 ; i,i,#2					
		b.l e 0xFFFC14B4					
-0000000021		MX:FFFC14B4 ptrace			--_aarch64_v8\sieve\func10+0x2E0	1.188us	-297.
434		v17 += v14;					
433		add w19,w19,w20 ; v17,v17,i					
		for (v14 = 0 ; v14 < 3 ; v14++)					
		add w20,w20,#0x1 ; i,i,#1					
		cmp w20,#0x2 ; i,i,#2					
		b.l e 0xFFFC14B4					
T0000000000		MX:FFFC14B4 ptrace			--_aarch64_v8\sieve\func10+0x2E0	1.188us	0.
434		v17 += v14;					
433		add w19,w19,w20 ; v17,v17,i					
		for (v14 = 0 ; v14 < 3 ; v14++)					
		add w20,w20,#0x1 ; i,i,#1					
		cmp w20,#0x2 ; i,i,#2					
		b.l e 0xFFFC14B4					
+0000000005		MX:FFFC14C4 ptrace			--_aarch64_v8\sieve\func10+0x2F0	1.188us	297.
435		for (v15 = 0 ; v15 < 3 ; v15++)					
		mov w20,#0x0 ; i,#0					
		b 0xFFFC14D4					
		cmp w20,#0x2 ; i,i,#2					
		b.l e 0xFFFC14CC					
+0000000010		MX:FFFC14CC ptrace			--_aarch64_v8\sieve\func10+0x2F8	0.652us	460.

CLOCKS.Zero

Number of clocks relative to global zero point

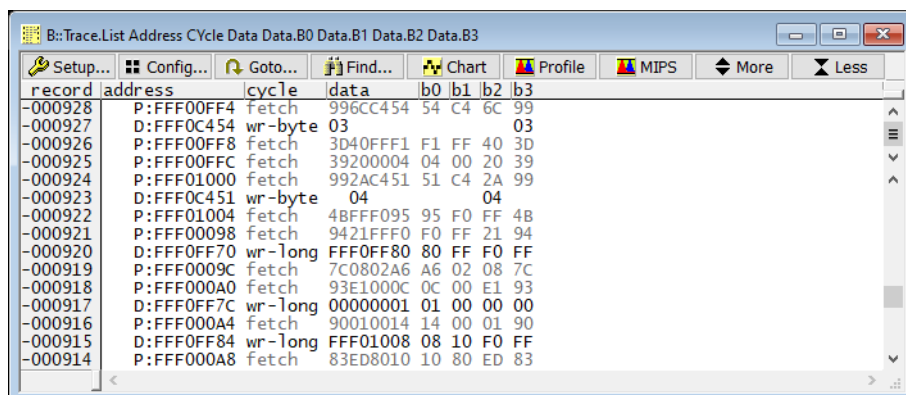
This item displays the number of core clock ticks that passed since the ZERO point in the trace recording. For tool-generated timestamps, the ZERO point default is set to the start of the first debug session since starting PowerView.

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.

Item reference: Data items

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.



record	address	cycle	data	b0	b1	b2	b3
-000928	P:FFF0FF4	fetch	996CC454	54	C4	6C	99
-000927	D:FFF0C454	wr-byte	03				03
-000926	P:FFF0FF8	fetch	3D40FFF1	F1	FF	40	3D
-000925	P:FFF0FFC	fetch	39200004	04	00	20	39
-000924	P:FFF01000	fetch	992AC451	51	C4	2A	99
-000923	D:FFF0C451	wr-byte	04				04
-000922	P:FFF01004	fetch	4BFFF095	95	F0	FF	4B
-000921	P:FFF00098	fetch	9421FFF0	F0	FF	21	94
-000920	D:FFF0FF70	wr-long	FFF0FF80	80	FF	F0	FF
-000919	P:FFF0009C	fetch	7C0802A6	A6	02	08	7C
-000918	P:FFF000A0	fetch	93E1000C	0C	00	E1	93
-000917	D:FFF0FF7C	wr-long	00000001	01	00	00	00
-000916	P:FFF000A4	fetch	90010014	14	00	01	90
-000915	D:FFF0FF84	wr-long	FFF01008	08	10	F0	FF
-000914	P:FFF000A8	fetch	83ED8010	10	80	ED	83

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.

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.

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.

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.

record	address	cycle	data	symbol
-000964	P:FFF01384	fetch	3FC0FFF1	..diabp4\Global__init_main+0x9C
-000963	P:FFF01388	fetch	3BDEC338	..diabp4\Global__init_main+0xA0
-000962	P:FFF0138C	fetch	022E8000	..diabp4\Global__init_main+0xA4
-000961	D:FFF0C3D0	wr - long	FFF0C338	\\diabp4\Global__env
-000960	P:FFF01390	fetch	70002C00	..diabp4\Global__init_main+0xA8
-000959	P:FFF04048	fetch	7C0802A6	\\diabp4\Global__init
-000958	P:FFF0404C	fetch	0431FF00	\\diabp4\Global__init+0x4
-000957	D:FFF0FF60	wr - long	FFF0FFA8	\\diabp4\Global__SP_END+0xFA8
-000956	P:FFF04050	fetch	00000000	\\diabp4\Global__init+0x8
-000955	D:FFF0FFAC	wr - long	FFF01394	\\diabp4\Global__init_main+0xAC
-000954	P:FFF04054	fetch	00000000	\\diabp4\Global__init+0x0C
-000953	D:FFF0FFA0	rd - long	FFF01394	\\diabp4\Global__init_main+0xAC
-000952	P:FFF04058	fetch	90210000	\\diabp4\Global__init+0x10
-000951	P:FFF0405C	fetch	7C0803A6	\\diabp4\Global__init+0x14
-000950	P:FFF04060	fetch	4E800020	\\diabp4\Global__init+0x18

Item reference: ENERGY items

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.

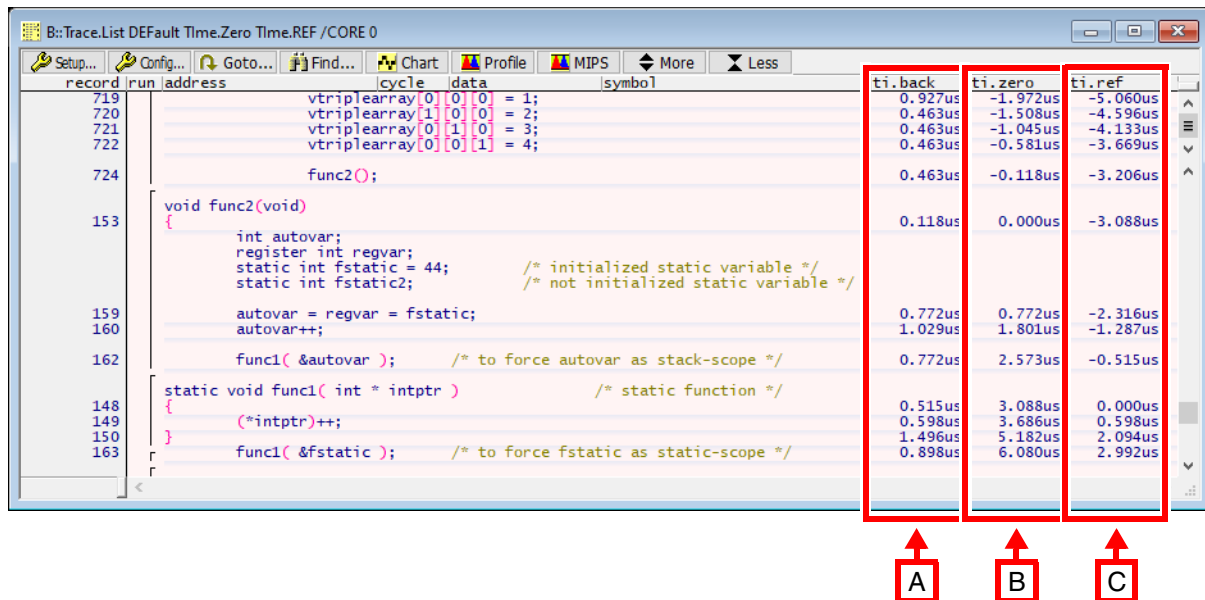
It is possible to align the ZERO time of unrelated trace sources. See command **ZERO** for details..

Item reference: Time items

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**.



- 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.

Usage: Trace.List Default Tlme.AddressFore <address>

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

Tlme.Back

Time relative to previous record

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

Tlme.Fore

Time relative to next record

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

Tlme.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).

record	run address	cycle	data	symbol	ti.back	ti.func
153	void func2(void)				0.119us	30.016us
	int autovar;					
	register int regvar;					
	static int fstatic = 44; /* initialized static variable */					
	static int fstatic2; /* not initialized static variable */					
159	autovar = regvar = fstatic;				0.889us	
160	autovar++;				1.185us	
162	func1(&autovar); /* to force autovar as stack-scope */				0.889us	
148	static void func1(int * intptr) /* static function */				0.593us	2.936us
149	{				0.587us	
150	(*intptr)++;				1.468us	
163	func1(&fstatic); /* to force fstatic as static-scope */				0.881us	
148	static void func1(int * intptr) /* static function */				0.628us	2.848us
149	{				0.570us	
150	(*intptr)++;				1.424us	

Tlme.FUNCX

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).

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.

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.

This item displays the time that passed since the trace reference point. See [Trace.REF](#) for more information.

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.

record	run	address	cycle	data	symbol	ti.back	ti.trigger
-0000000032		b.le 0xFFFC14B4		ptrace	MX:FFFC14B4	1.200us	-2.376us
434		v17 += v14;			__aarch64_v8\sieve\func10+0x2E0		
		add w19,w19,w20		: v17,v17,i			
433		for (v14 = 0 ; v14 < 3 ; v14++)					
		add w20,w20,#0x1		: i,i,#1			
		cmp w20,#0x2		: i,#2			
-0000000021		b.le 0xFFFC14B4		ptrace	MX:FFFC14B4	1.188us	-1.188us
434		v17 += v14;			__aarch64_v8\sieve\func10+0x2E0		
		add w19,w19,w20		: v17,v17,i			
433		for (v14 = 0 ; v14 < 3 ; v14++)					
		add w20,w20,#0x1		: i,i,#1			
		cmp w20,#0x2		: i,#2			
T0000000000		b.le 0xFFFC14B4		ptrace	MX:FFFC14B4	1.188us	0.000us
434		v17 += v14;			__aarch64_v8\sieve\func10+0x2E0		
		add w19,w19,w20		: v17,v17,i			
433		for (v14 = 0 ; v14 < 3 ; v14++)					
		add w20,w20,#0x1		: i,i,#1			
		cmp w20,#0x2		: i,#2			
+0000000005		b.le 0xFFFC14C4		ptrace	MX:FFFC14C4	1.188us	1.188us
435		for (v15 = 0 ; v15 < 3 ; v15++)			__aarch64_v8\sieve\func10+0x2F0		

Time.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.