# Lauterbach Ltd

Many in this audience have to meet specifications such as – DO178C / ISO26262 / IEC62304

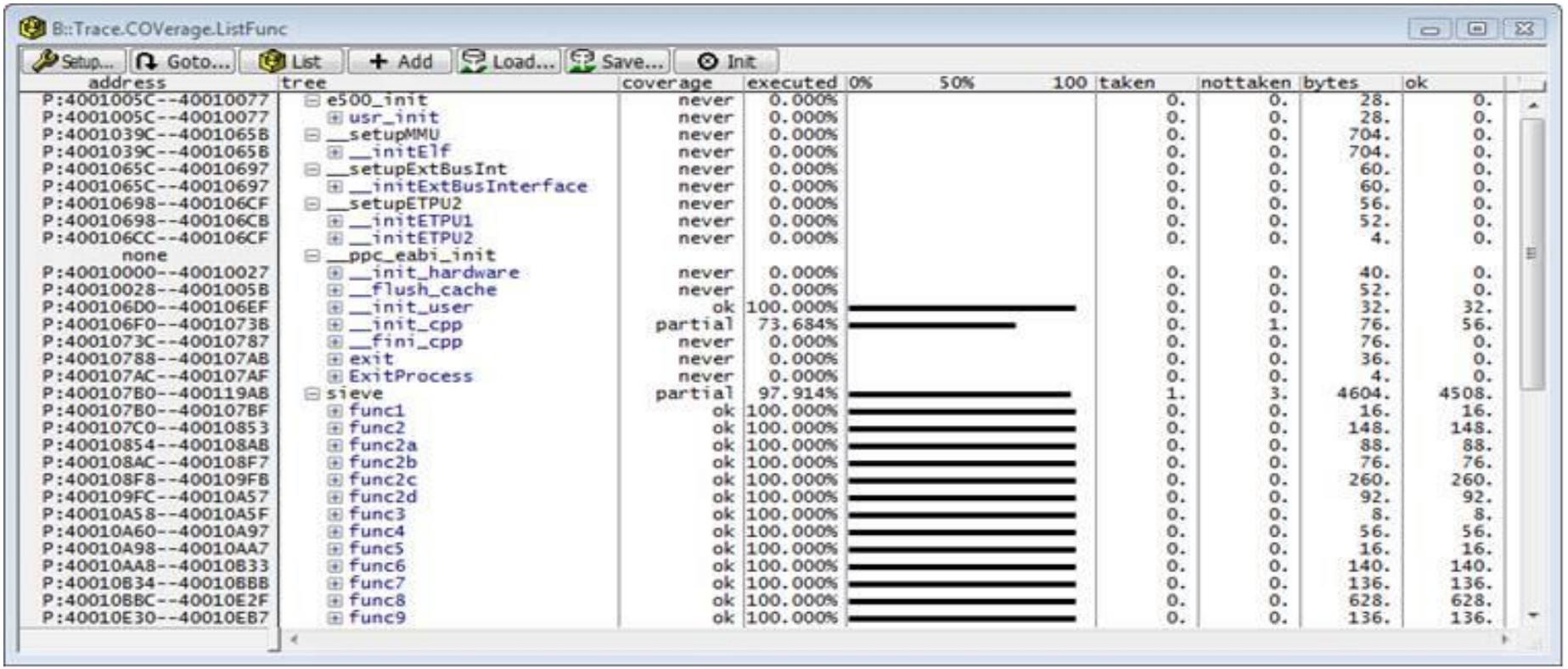
For many years we have been able to provide analysis of your software with :-
NO INSTRUMENTATION
CODE RUNNING IN REAL TIME

This includes :-

- Code coverage
- Best and worst case timings
- Core loading
- Cache usage

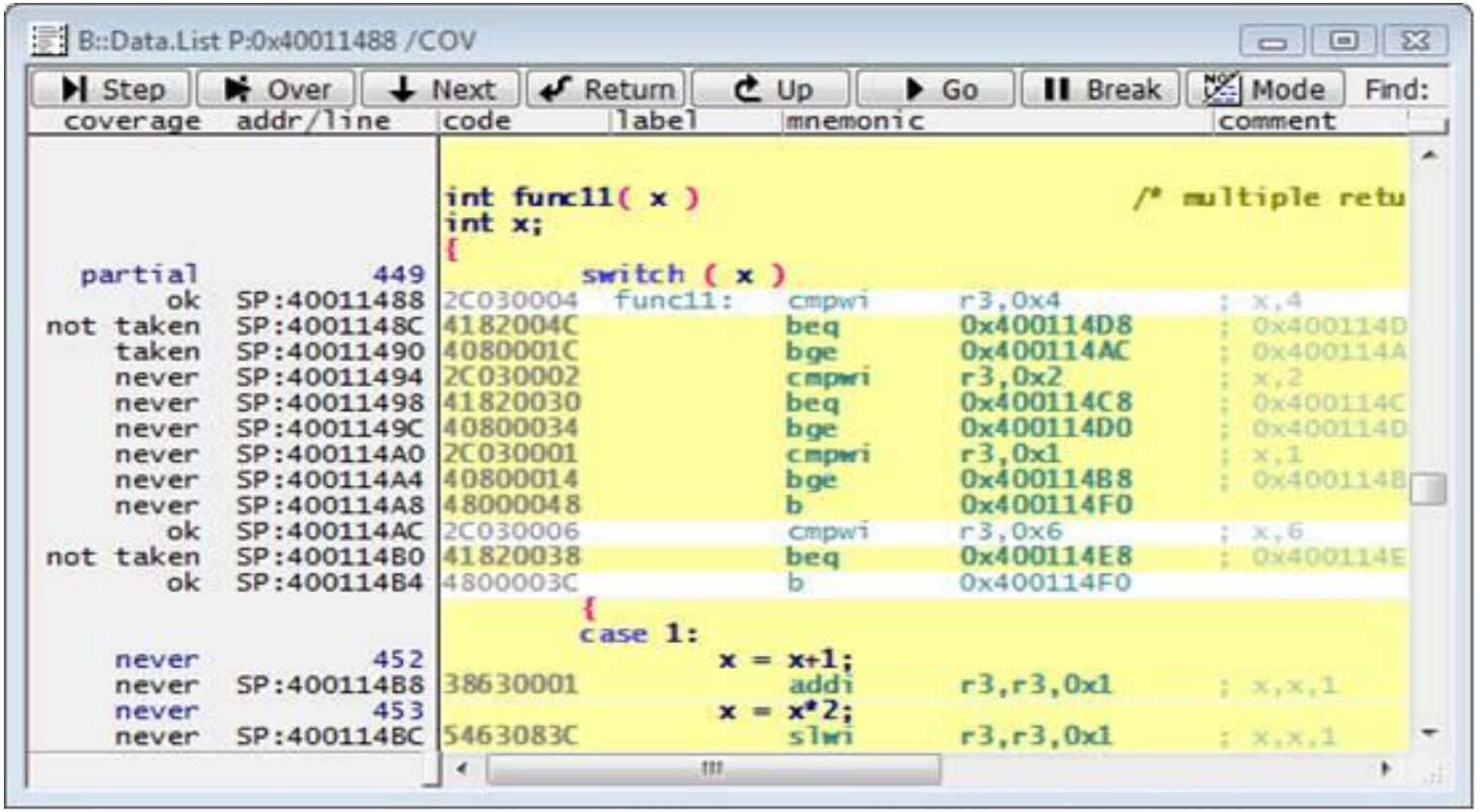
From mid 2014 we will offer a tool kit you can use to produce reports for most safety  specifications :-

- A manual on how to use the tools to produce the reports
- Tool qualification details for each safety standard
- Operational requirements
- Tool verification and validation cases
- Records of test results

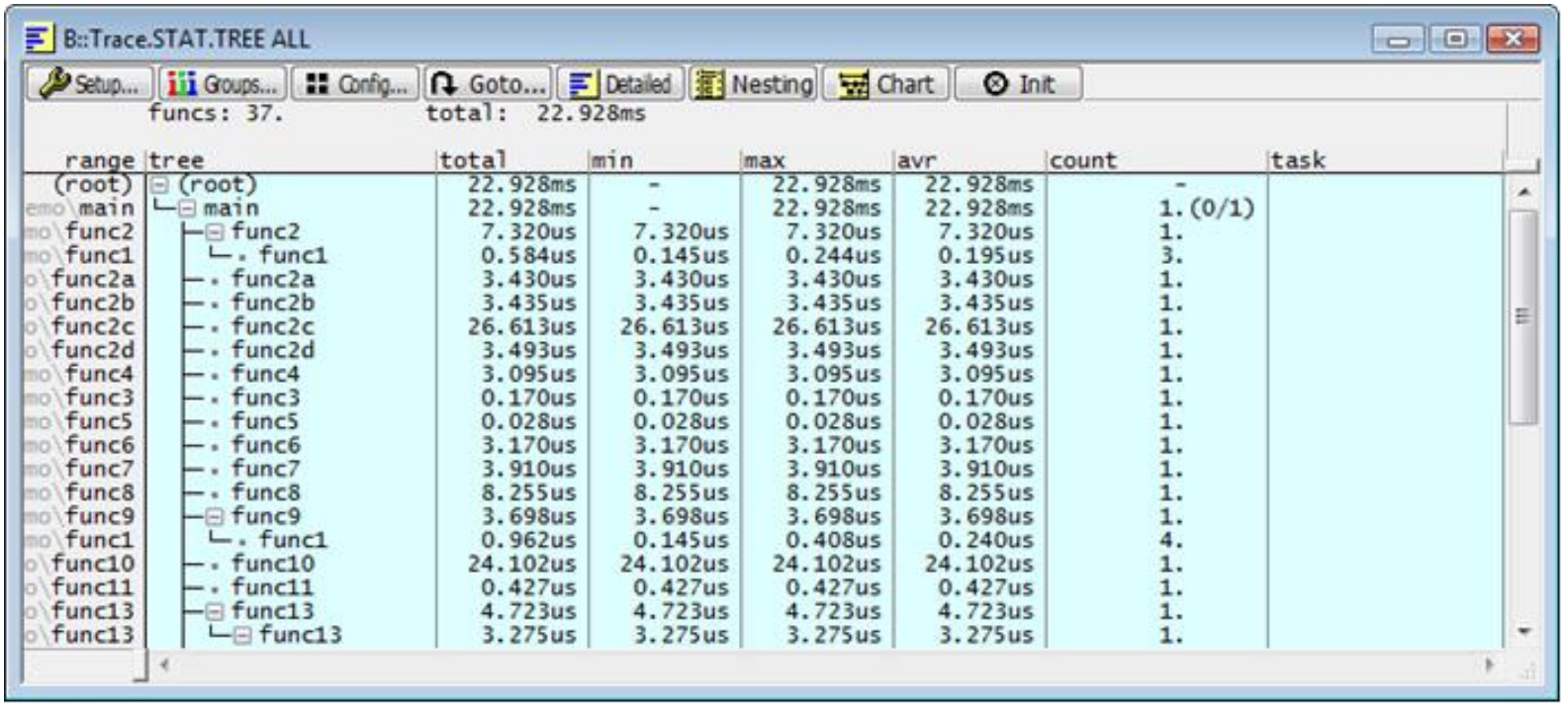



| address | tree | coverage | executed | 0% 50% 100 | taken | nottaken | bytes | ok |
|---|---|---|---|---|---|---|---|---|
| P:4001005C--40010077 | e500_init | never | 0.000% | | 0. | 0. | 28. | 0. |
| P:4001005C--40010077 | usr_init | never | 0.000% | | 0. | 0. | 28. | 0. |
| P:4001039C--4001065B | __setupMMU | never | 0.000% | | 0. | 0. | 704. | 0. |
| P:4001039C--4001065B | __initElf | never | 0.000% | | 0. | 0. | 704. | 0. |
| P:4001065C--40010697 | __setupExtBusInt | never | 0.000% | | 0. | 0. | 60. | 0. |
| P:4001065C--40010697 | __initExtBusInterface | never | 0.000% | | 0. | 0. | 60. | 0. |
| P:40010698--400106CF | __setupETPU2 | never | 0.000% | | 0. | 0. | 56. | 0. |
| P:40010698--400106CB | __initETPU1 | never | 0.000% | | 0. | 0. | 52. | 0. |
| P:400106CC--400106CF | __initETPU2 | never | 0.000% | | 0. | 0. | 4. | 0. |
| none | __ppc_eabi_init | | | | | | | |
| P:40010000--40010027 | __init_hardware | never | 0.000% | | 0. | 0. | 40. | 0. |
| P:40010028--4001005B | __flush_cache | never | 0.000% | | 0. | 0. | 52. | 0. |
| P:400106D0--400106EF | __init_user | ok | 100.000% | | 0. | 0. | 32. | 32. |
| P:400106F0--4001073B | __init_cpp | partial | 73.684% | | 0. | 1. | 76. | 56. |
| P:4001073C--40010787 | __fini_cpp | never | 0.000% | | 0. | 0. | 76. | 0. |
| P:40010788--400107AB | exit | never | 0.000% | | 0. | 0. | 36. | 0. |
| P:400107AC--400107AF | ExitProcess | never | 0.000% | | 0. | 0. | 4. | 0. |
| P:400107B0--400119AB | sieve | partial | 97.914% | | 1. | 3. | 4604. | 4508. |
| P:400107B0--400107BF | func1 | ok | 100.000% | | 0. | 0. | 16. | 16. |
| P:400107C0--40010853 | func2 | ok | 100.000% | | 0. | 0. | 148. | 148. |
| P:40010854--400108AB | func2a | ok | 100.000% | | 0. | 0. | 88. | 88. |
| P:400108AC--400108F7 | func2b | ok | 100.000% | | 0. | 0. | 76. | 76. |
| P:400108F8--400109FB | func2c | ok | 100.000% | | 0. | 0. | 260. | 260. |
| P:400109FC--40010A57 | func2d | ok | 100.000% | | 0. | 0. | 92. | 92. |
| P:40010A58--40010A5F | func3 | ok | 100.000% | | 0. | 0. | 8. | 8. |
| P:40010A60--40010A97 | func4 | ok | 100.000% | | 0. | 0. | 56. | 56. |
| P:40010A98--40010AA7 | func5 | ok | 100.000% | | 0. | 0. | 16. | 16. |
| P:40010AA8--40010B33 | func6 | ok | 100.000% | | 0. | 0. | 140. | 140. |
| P:40010B34--40010BBB | func7 | ok | 100.000% | | 0. | 0. | 136. | 136. |
| P:40010BBC--40010E2F | func8 | ok | 100.000% | | 0. | 0. | 628. | 628. |
| P:40010E30--40010EB7 | func9 | ok | 100.000% | | 0. | 0. | 136. | 136. |

Code coverage overview showing object, function and source line with branches taken or not taken and lines executed or not
We now extract reports in the correct format to include in your certification package




The quick view whilst developing the code
Yellow code is code not executed or branches not taken – click for more detailed view



B::Trace.STAT.TREE ALL

Setup... Groups... Config... Goto... Detailed Nesting Chart Init

funcs: 37. total: 22.928ms

| range | tree | total | min | max | avr | count | task |
|---|---|---|---|---|---|---|---|
| (root) | (root) | 22.928ms | - | 22.928ms | 22.928ms | - | |
| emo\main | main | 22.928ms | - | 22.928ms | 22.928ms | 1.(0/1) | |
| mo\func2 | func2 | 7.320us | 7.320us | 7.320us | 7.320us | 1. | |
| mo\func1 | . func1 | 0.584us | 0.145us | 0.244us | 0.195us | 3. | |
| o\func2a | . func2a | 3.430us | 3.430us | 3.430us | 3.430us | 1. | |
| o\func2b | . func2b | 3.435us | 3.435us | 3.435us | 3.435us | 1. | |
| o\func2c | . func2c | 26.613us | 26.613us | 26.613us | 26.613us | 1. | |
| o\func2d | . func2d | 3.493us | 3.493us | 3.493us | 3.493us | 1. | |
| mo\func4 | . func4 | 3.095us | 3.095us | 3.095us | 3.095us | 1. | |
| mo\func3 | . func3 | 0.170us | 0.170us | 0.170us | 0.170us | 1. | |
| mo\func5 | . func5 | 0.028us | 0.028us | 0.028us | 0.028us | 1. | |
| mo\func6 | . func6 | 3.170us | 3.170us | 3.170us | 3.170us | 1. | |
| mo\func7 | . func7 | 3.910us | 3.910us | 3.910us | 3.910us | 1. | |
| mo\func8 | . func8 | 8.255us | 8.255us | 8.255us | 8.255us | 1. | |
| mo\func9 | func9 | 3.698us | 3.698us | 3.698us | 3.698us | 1. | |
| mo\func1 | . func1 | 0.962us | 0.145us | 0.408us | 0.240us | 4. | |
| o\func10 | . func10 | 24.102us | 24.102us | 24.102us | 24.102us | 1. | |
| o\func11 | . func11 | 0.427us | 0.427us | 0.427us | 0.427us | 1. | |
| o\func13 | func13 | 4.723us | 4.723us | 4.723us | 4.723us | 1. | |
| o\func13 | func13 | 3.275us | 3.275us | 3.275us | 3.275us | 1. | |

**Performance analysis in numbers**

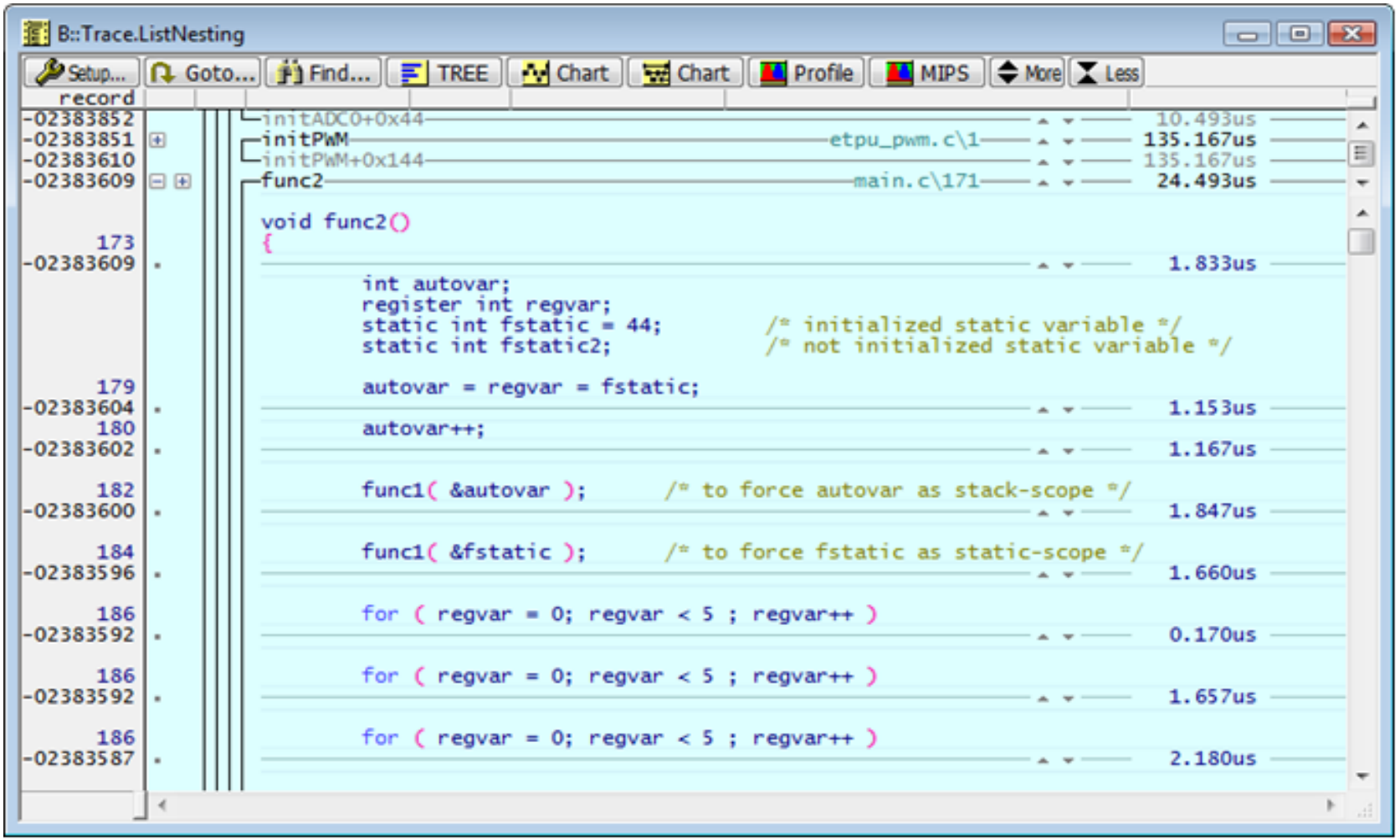
Call nesting

Time in each function

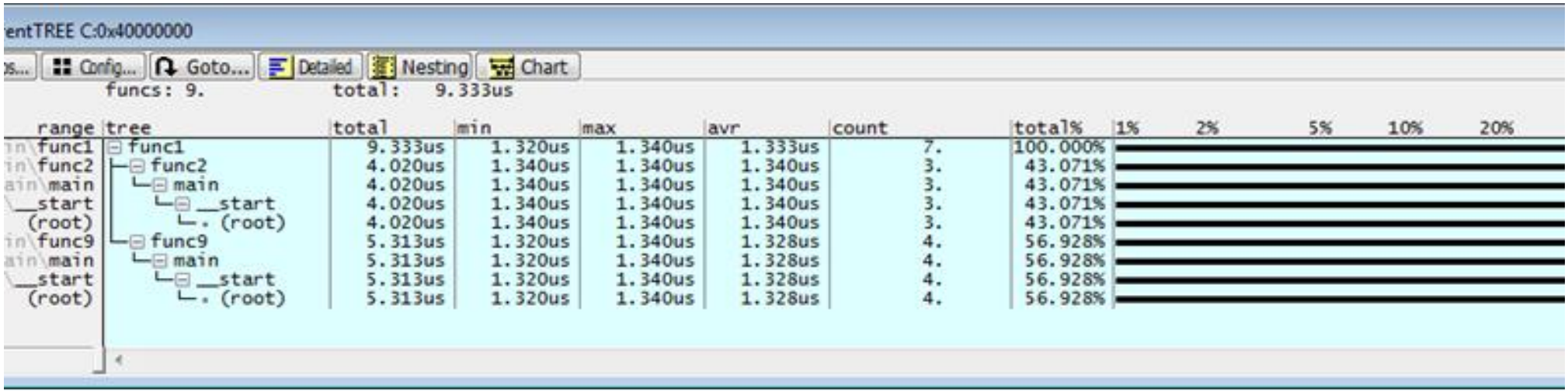
Time for interrupt handlers

Time for task switching

max min and mean runtimes




Expand to see high level language and timings




| range | tree | total | min | max | avr | count | total% | 1% 2% 5% 10% 20% |
|---|---|---|---|---|---|---|---|---|
| in\func1 | func1 | 9.333us | 1.320us | 1.340us | 1.333us | 7. | 100.000% | |
| in\func2 | func2 | 4.020us | 1.340us | 1.340us | 1.340us | 3. | 43.071% | |
| ain\main | main | 4.020us | 1.340us | 1.340us | 1.340us | 3. | 43.071% | |
| __start | __start | 4.020us | 1.340us | 1.340us | 1.340us | 3. | 43.071% | |
| (root) | . (root) | 4.020us | 1.340us | 1.340us | 1.340us | 3. | 43.071% | |
| in\func9 | func9 | 5.313us | 1.320us | 1.340us | 1.328us | 4. | 56.928% | |
| ain\main | main | 5.313us | 1.320us | 1.340us | 1.328us | 4. | 56.928% | |
| __start | __start | 5.313us | 1.320us | 1.340us | 1.328us | 4. | 56.928% | |
| (root) | . (root) | 5.313us | 1.320us | 1.340us | 1.328us | 4. | 56.928% | |

Showing the call tree for each function with timings and ratios

# What do you need ? A core with off chip trace

With the right tools it can provide full code history from reset or start and record for hours or days

Most core types have variants with trace ports
ARM, MPC, Qorivva, QoriQ, Tricore, Mips, Atom, SH, etc