Ahmed Barakat, Field Application Engineer (TASKING GmbH)
AURIX™ After-Lunch-Seminar
TASKING: A Full Set of TriCore/AURIX Development Tools

Compiler - Optimized to leverage critical features of target processors, and developed with ASPICE CL2 processes to qualify for safety-critical applications. Includes Debugger and Linker/Locator.

Embedded Profiler - More than performance measurement, it shows you the source of the performance bottleneck and how optimize your code.

LAPACK Performance Libraries - Provides rapid and accurate solutions to complex mathematical operations - all within a safety-critical environment.

Embedded Debugger - At an accessible price, your team can finally own enough debuggers so you can verify code functionality without workflow delay.

Safety Checker - You can ensure freedom from interference with static safety analysis of your code.

Compiler Qualification Kit - Required documentation pertaining to the TASKING toolset helps streamline your ISO 26262 safety certification process.
• Maintaining embedded software quality becomes more difficult as the complexity of intelligent devices increases.

• Many companies have failed to efficiently test and deliver reliable products due to the rising cost of verifying additional software.

• The number of defects rises proportionally with more lines of code. Engineering managers are looking for ways to mitigate the time and money spent on maintaining the quality of these complex systems so they can reduce time to market and retain budget for competitive features.

• Many companies still need to find additional methods of identifying problems sooner.
● Co-developed with Infineon — Expert knowledge about AURIX architecture built in

● Fully exploit the performance potential of AURIX devices whether novice or expert

● Identifies inter- and intra-core bottlenecks

● Measures performance of instruction pipelines, instruction caches, data caches and memory including inter-core effects.

● Measurements for whole application or selected functions only
World’s first Smart Performance Optimization tool for AURIX.
Developed in cooperation with Infineon.
Non-intrusive analysis will be done on the chip without having to buy special hardware.
Identifies the exact location and nature of the Performance bottleneck and gives you hints how to fix it.

What causes performance issues
Stalls
For example memory stalls are common on the AURIX due the amount and connection of memories and can only be prevented with a good layout.

Pipeline hazards
Data hazards if instructions with a data dependence modify data in different stages of a pipeline.
A structural hazard occurs when a part of the processor’s hardware is needed by two or more instructions at the same time.
Control hazards or branching hazards when the processor does not know the outcome of the branch.
TASKING Embedded Profiler

- Finds location and nature performance bottlenecks, allowing non-expert users to find and fix problems like an expert
- Displays the problem, shows a detailed list of errors and gives the user precise suggestions to solve the problem
- Compare results before and after to see the optimization effects — including possible unwanted side effects
- Profile analysis levels
Profiler Analysis Type

Performance Analysis
This type of analysis traces instructions and performance events. It measures the CPU clock count and it finds branch misses, cache misses and stalls due to memory access delays or pipeline hazards. You can run this type of analysis on the whole application or select specific functions.

Memory Access Analysis
This type of analysis traces function calls, function returns and data accesses. You can run this type of analysis on the whole application or select specific functions.

Function-level Analysis
This type of analysis traces all function calls and function returns. This is the fastest analysis.
Useful links

• For more information and webinars please visit
  www.tasking.com/webinars
  www.tasking.com/white-papers

• To request a free trial, please visit
  www.tasking.com/trial

• Contact our sales department at
  www.tasking.com/buy-now
TASKING: A *Full Set* of TriCore/AURIX Development Tools

**Compiler** - Optimized to leverage critical features of target processors, and developed with ASPICE CL2 processes to qualify for safety-critical applications. Includes Debugger and Linker/Locator.

**Embedded Profiler** - More than performance measurement, it shows you the source of the performance bottleneck and how optimize your code.

**LAPACK Performance Libraries** - Provides rapid and accurate solutions to complex mathematical operations - all within a safety-critical environment.

**Embedded Debugger** - At an accessible price, your team can finally own enough debuggers so you can verify code functionality *without* workflow delay.

**Safety Checker** - You can ensure freedom from interference with static safety analysis of your code.

**Compiler Qualification Kit** - Required documentation pertaining to the TASKING toolset helps streamline your ISO 26262 safety certification process.
Performance Libraries LAPACK & BLAS

TASKING LAPACK Libraries

- TASKING is only supplier for TriCore/AURIX optimized LAPACK — only LAPACK library implemented in C
- LAPACK with BLAS available for AURIX and AURIX 2nd Generation and later
- LAPACK libraries facilitate development of high-performance signal processing and ADAS applications
- Library accuracy proven over decades
Performance Libraries LAPACK & BLAS

• Highly optimized numerical library
• Full LAPACK/BLAS single-precision support
• Compatible with MathWorks model-based design software
• Easily port existing mathematical software to embedded devices
• ISO26262 & ASIL-B compatible
• C source code available (Beside pre-compiled binary files, makefiles, user documentation and examples)
Compact with Matlab Generated Code

Generated Code *.C, *.h

LAPACK Pre-compiled
lapack.h *.a

C,C++ Compiler
lapack.h *.a

Integrate Tasking LAPACK

OR

*.h,*.C

Tasking

*.*.elf

Compiler

Makefile

Linker

LAPACK Source

*.h,*.C

*.a

LAPACK

Matlab Simulink
**BLAS - Basic Linear Algebra Subprograms**

**SSYMM**

Performs one of the matrix-matrix operations

\[ C = \alpha A^T B + \beta C, \]

or

\[ C = \alpha B^T A + \beta C, \]

where \( \alpha \) and \( \beta \) are scalars, \( A \) is a symmetric matrix and \( B \) and \( C \) are \( m \) by \( n \) matrices.

**SGEMM**

Performs one of the matrix-matrix operations in pseudo code:

\[ C = \alpha \text{op}(A)^T \text{op}(B) + \beta C, \]

where \( \text{op}(X) \) is one of

\[ \text{op}(X) = X \text{ or } \text{op}(X) = X^T, \]

where \( \alpha \) and \( \beta \) are scalars, and \( A, B \) and \( C \) are matrices, with \( \text{op}(A) \) an \( m \) by \( k \) matrix, \( \text{op}(B) \) a \( k \) by \( n \) matrix and \( C \) an \( m \) by \( n \) matrix.

**SDOT**

Forms the dot product of two vectors.

**SGEMV**

Performs one of the matrix-vector operations

\[ y := \alpha A x + \beta y, \text{ or } \]

\[ y := \alpha A^T x + \beta y, \]

where \( \alpha \) and \( \beta \) are scalars, \( x \) and \( y \) are vectors and \( A \) is an \( m \) by \( n \) matrix.

---

**Graphs**

- **SGEMM & SGEMV Performance**
  - Tricore TC29x at 300 MHz

- **SDOT & SSYM Performance**
  - Tricore TC29x at 300 MHz
Q&A

Learn more about Hitex at www.hitex.com