hitex

AURIX KNOWLEDGE LAB 2024

ETAS HIGHTEC imagimob An Infineon Technologies Company

Infineon pls Development Tools TRUST IN SOFT

Debugging of Multicore Applications

Jens Braunes

pls Development Tools

Master the Challenge of Debugging Heterogeneous Multicore Applications

Jens Braunes

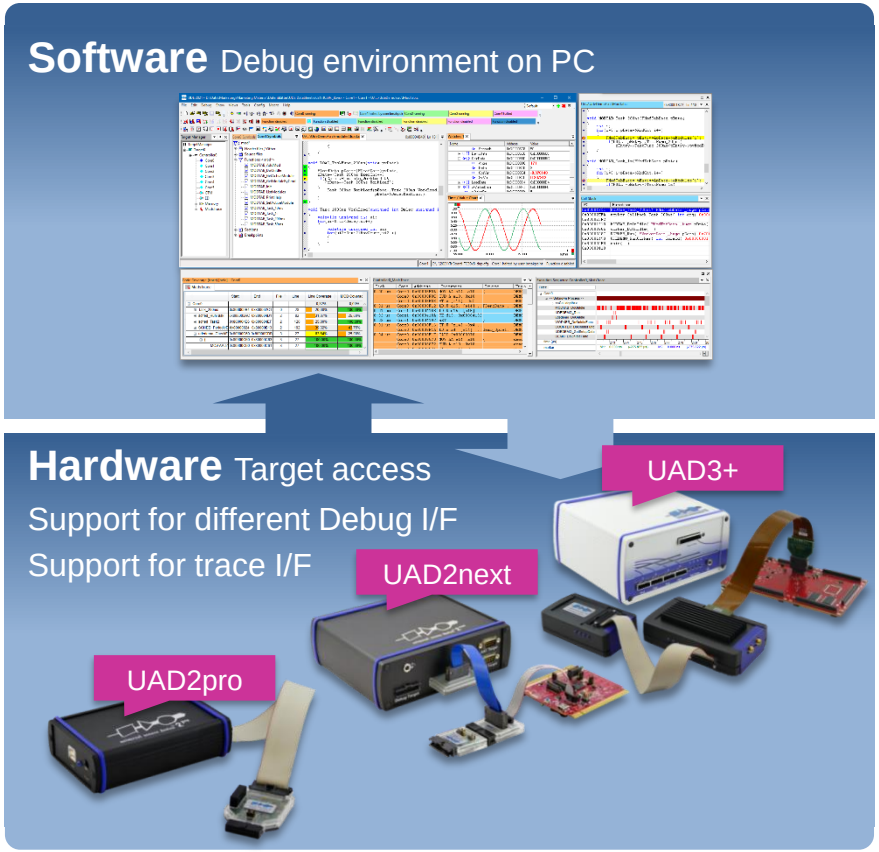
pls Development Tools

UDE Universal Debug Engine

Complete tool set for debugging and trace

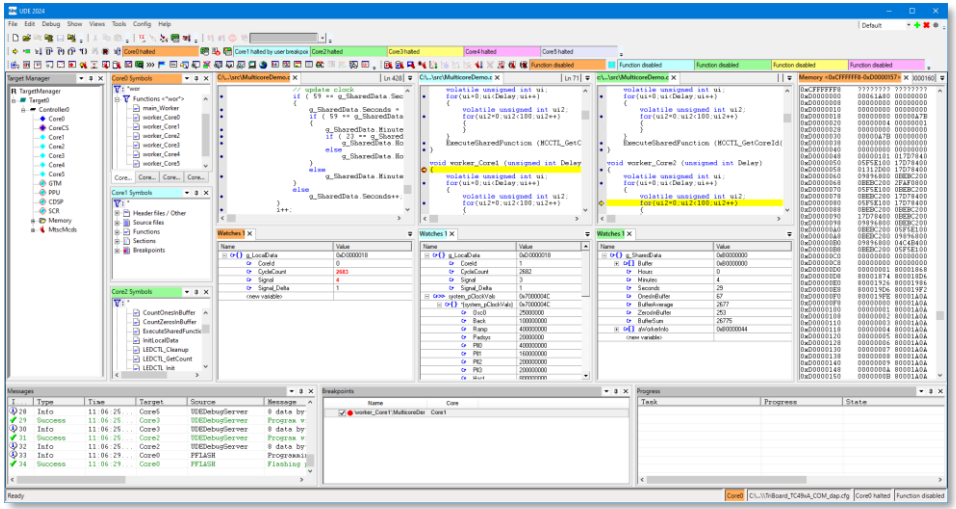
- On real target hardware
  - Microcontrollers, embedded systems, multicore SoCs
- On virtual prototypes
  - Simulators

- Real multicore debugging
  - System centric debugger environment, not core centric
  - One common user interface
  - Integrated Core Debuggers instead of separate debugger UI instances for cores

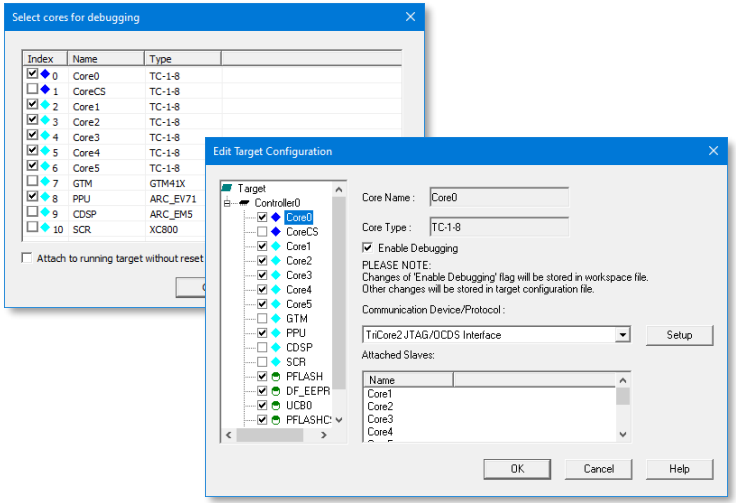


Multicore Debugging with UDE

- **Complete**
  - Debugger environment for system, not for core
  - All cores (main cores, special cores) visible
  - One common user interface
  - No separate debugger instances
  - Core coloring

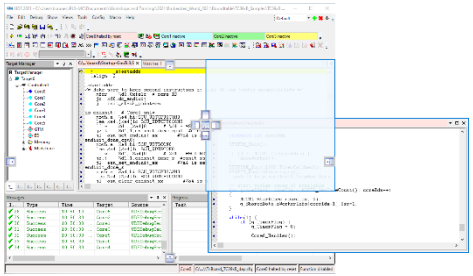


- **Easy-to-use**
  - User interface guides to setup process
  - Predefined configurations for Architectures, Devices, Boards
  - Select cores for debugging



Intuitive User Interface for Multicore Debugging

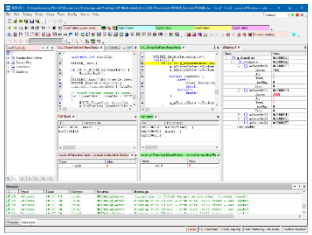
- Modern and state-of-the-art Window application
- Fully flexible
  - Arrange and group windows as needed



- Settings are persistent
  - Continue work where you left off, get the same environment the next day

Multi-Screen Support

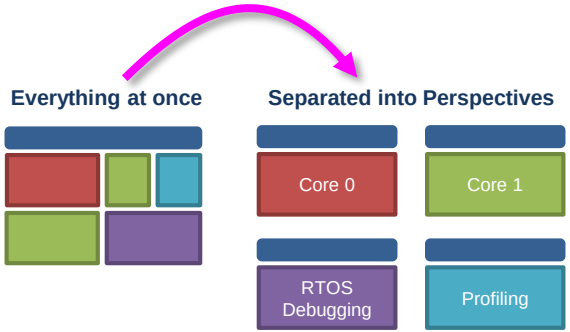
- Individual windows can be detached from main UDE window
- Detached windows can be grouped into docking containers



- Individual windows and dock containers can be moved to secondary screens

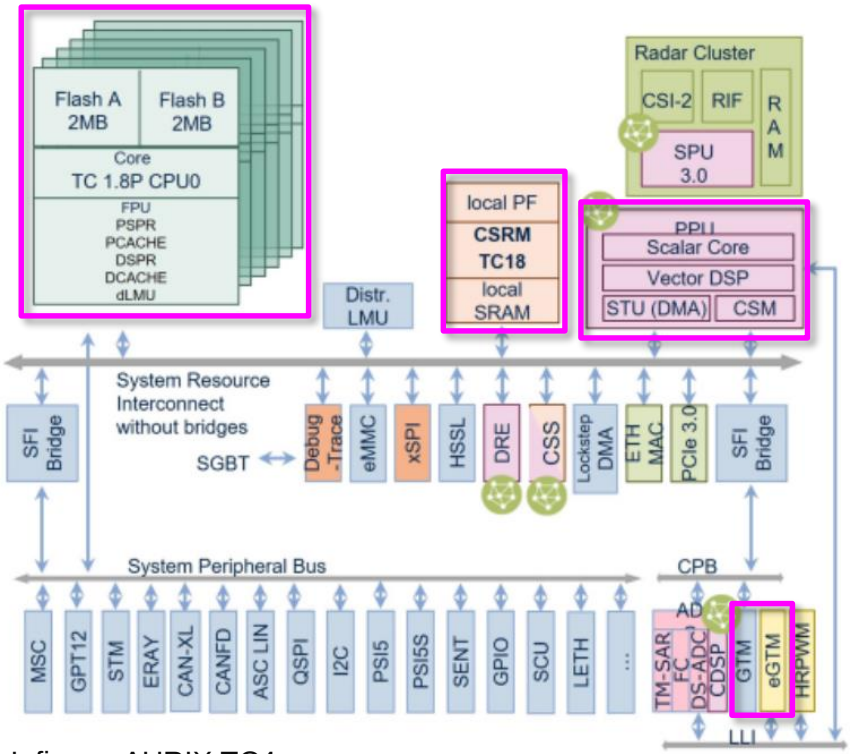
Perspectives

- Help to focus on the essentials
- One debugger session → multiple debug and test tasks → Perspectives
- Add perspectives as you like
- Freely configurable
  - In Layout
  - Contained windows (not only specific to one Core)



The Need for Multicore Debugging for TC4x

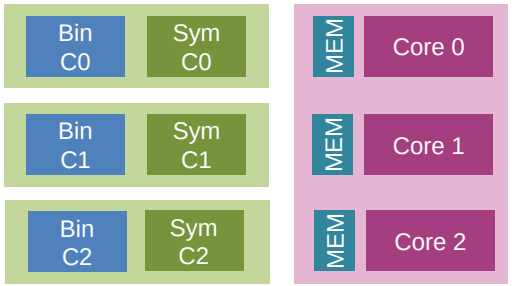
- Homogeneous core architectures
  - Up to 6 **TriCore** cores
- Heterogeneous core architectures
  - **SCRM** (Security Module)
  - **GTMv4** (Generic Timer Module)
  - **PPU** (Parallel Processing Unit based on Synopsys ARC architecture)
  - **SCR** (Standby Controller)



Infineon AURIX TC4x
(Infineon website)

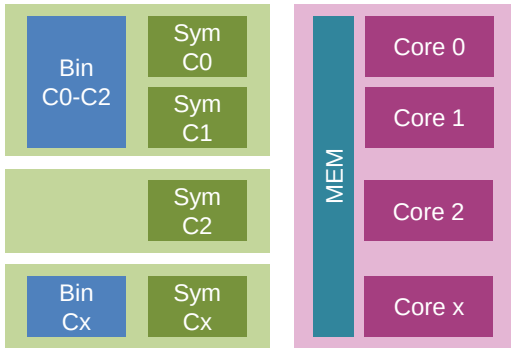
Multicore Software Architectures & Debug Scenarios

Several independent applications



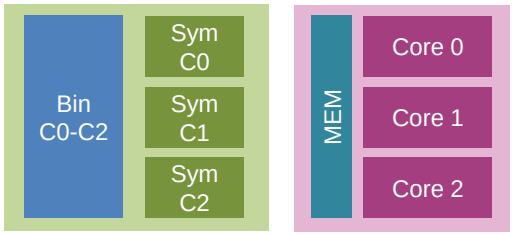
- Multiple ELF/HEX files
- One application per core
- Independent run control
- **Break / Step / Go only for one core**

Heterogeneous code



- One/multiple HEX / multiple ELF files
- E.g. different code base for heterogeneous cores
- Partially synchronized / fully synchronized run control
- **Break / Step / Go at the same time for selected cores**

Monolithic application

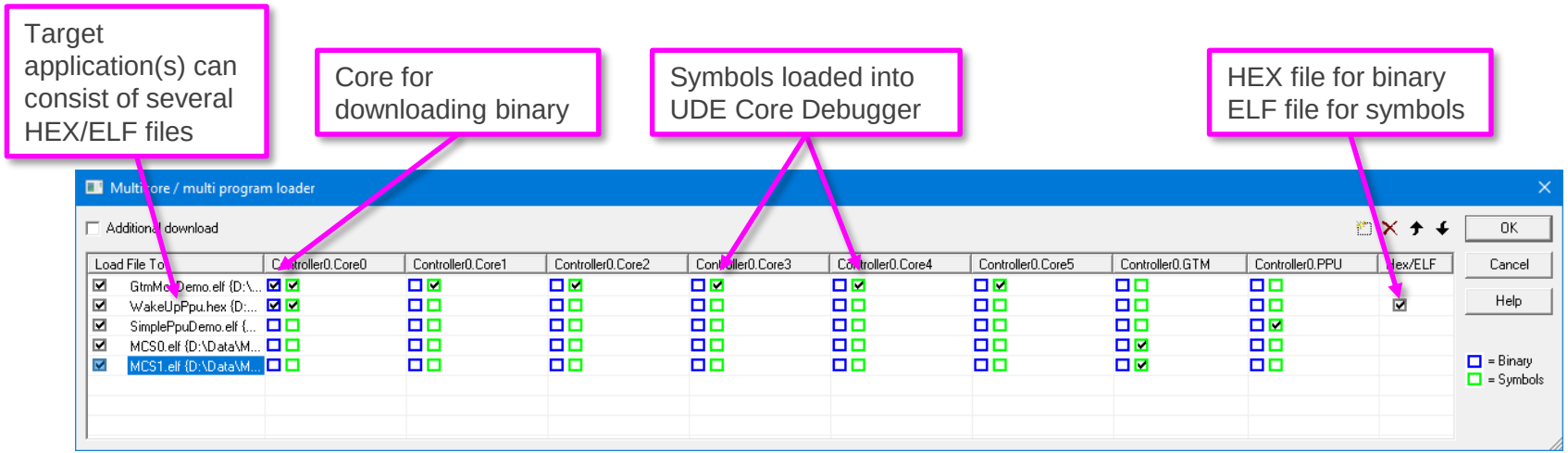


- One ELF/HEX file
- Tasks distributed over several cores
- Close dependencies between tasks (data, timing)
- Synchronous run control
- **Break / Step / Go at the same time for all cores**

pls Development Tools

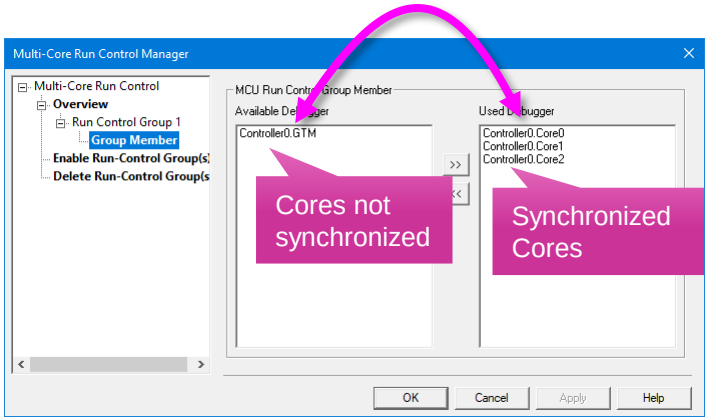
Managing TC4x Multicore Applications

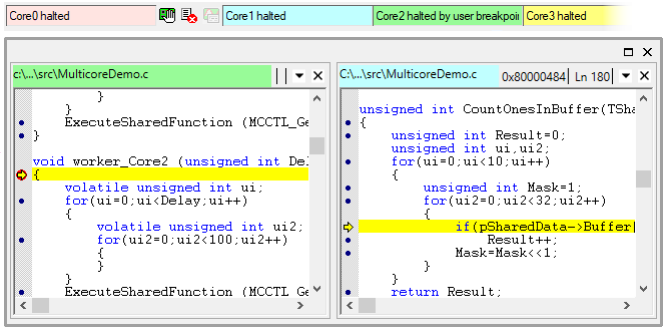
- Maximum flexibility in terms of software architecture
- Support for
  - Monolithic applications (single ELF/HEX)
  - Multiple HEX/ELF files, to be load to different cores
- Separate handling of binaries (patterns) and symbols
- HLL support for
  - C/C++
  - Rust
  - OpenCL
  - Compilers from HighTec, Synopsys, Tasking
- Assembler



Debug Synchronization / Multicore Run Control

- In general run-control is core related
  - Go/Break/Step effective only for active core debugger
  - Reset effective for all cores
- Debug Sync of cores by UDE run-control group
  - Allows to synchronize cores for Go/Break/Step
  - Go/Break/Step effective for all core debuggers in group
- Multicore Run Control Manager for configuration
  - Cores can be added or removed from run-control group

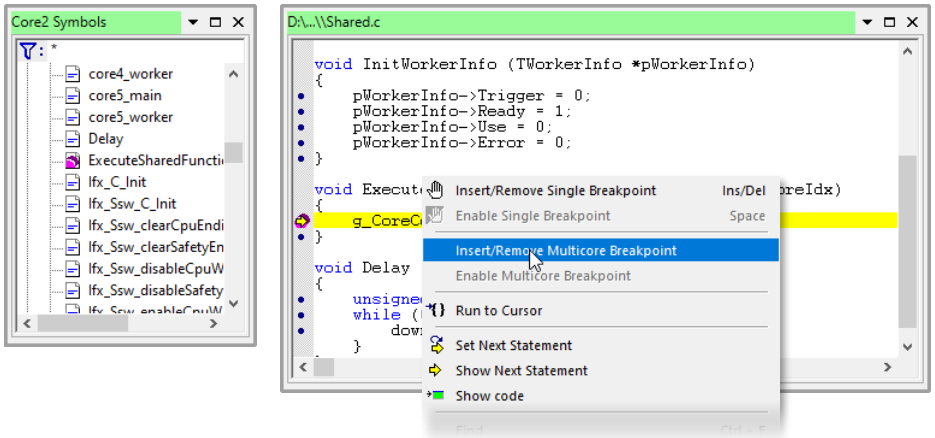




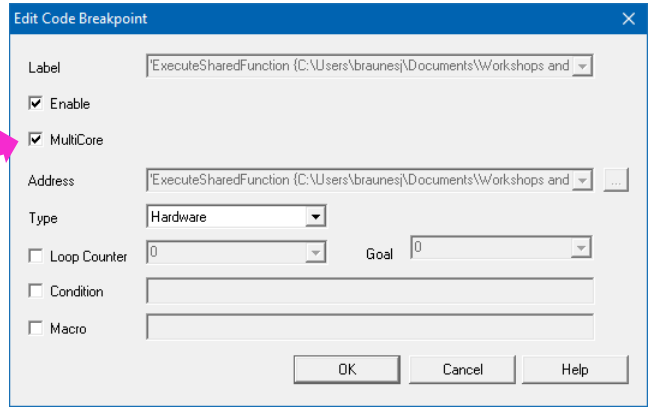
Synchronized Run-Control
All / several cores halted if one core hits a breakpoint

Multicore Breakpoints

- Useful to set breakpoints in shared code
- Effective in all cores of a corresponding run-control group

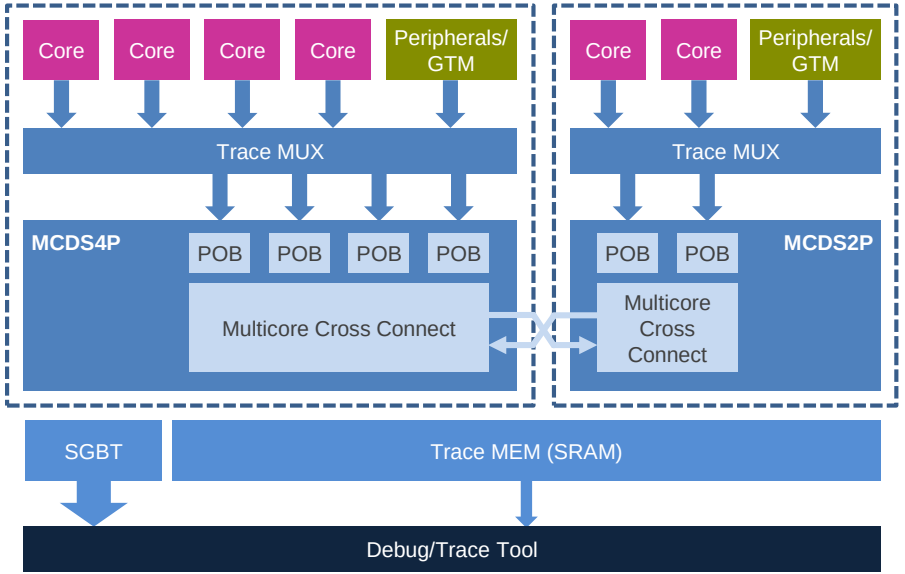


- Single-core breakpoint can be changed to Multicore Breakpoint



On-chip Trace for Debugging / Non-intrusive Analysis

- MCDS – Multicore Debug Solution
  - Two instances possible for TC4x (MCDS4P/2P)
- Parallel trace of multiple Clients (subset of clients)
  - Cores
  - Busses
  - Peripherals
- Trigger and filters to focus trace output on relevant information
- Cross-Triggers to distribute events
  - E.g enable trace on core Y if core X enters an ISR
  - Between MCDS instances MCDS4P/2P
- Trace output
  - On-Chip trace memory (SRAM of TC4x)
    - Into UDE via DAP
  - SGBT Serial Gigabit Trace
    - UAD2next
    - UAD3+



Analyzing and Visualizing Trace Data

Trace Based Debugging

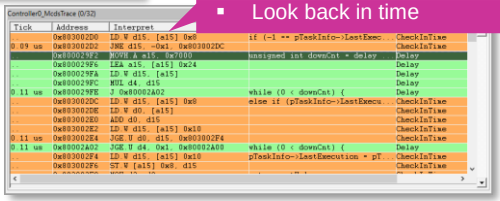
- Find bugs
- Look back in time
- Investigate bugs caused by parallel execution
  - Deadlocks, race conditions, timing issues

Trace Based Measurement and Runtime Analysis

- Profiling
- Call graph analysis
- Visualization of program execution
- Performance analysis
- Code coverage

Trace Window

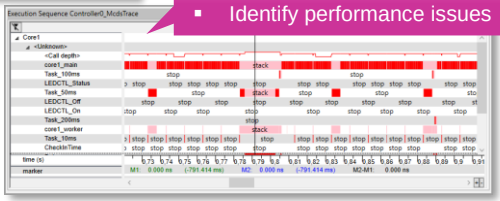
- Debugging
- Look back in time



- Sequential, time aligned list of recorded trace data
  - Program flow
  - Data transfers
  - Misc. data

Execution Sequence Chart

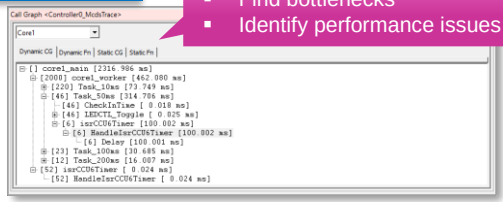
- Find bottlenecks
- Identify performance issues



- Graphical visualization of executed functions
- Task changes over time for (RTOS support)
- Visualization of AUTOSAR OS execution

Call Graph

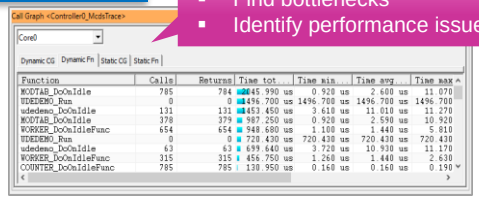
- Find bottlenecks
- Identify performance issues



- Call relationships of functions

Call Graph / Profiling

- Find bottlenecks
- Identify performance issues



- Profiling information

Defining Trace & Debug Tasks

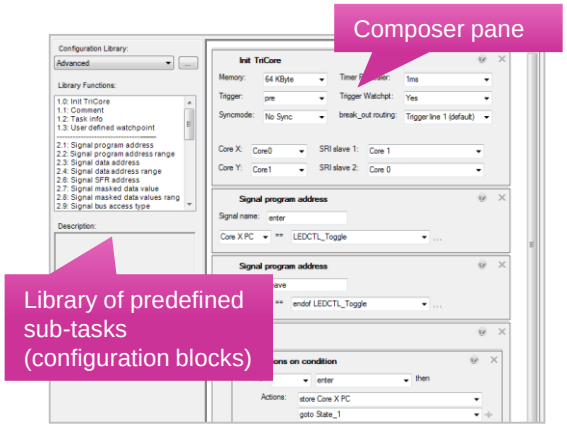
Define a trace / debug task
- What has to be recorded
- When it has to be recorded

Run the trace

Analyze the trace result

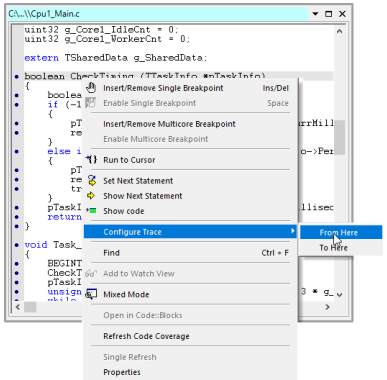
UEC graphical configuration tool

- Manual configuration
- Maximum flexibility
- Simplifies the complexity introduced with two MCDS instances
- Compose trace and debug tasks



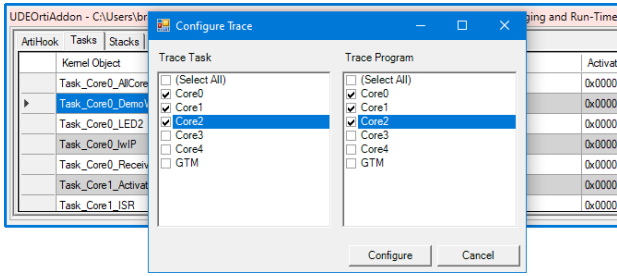
UDE SimplyTrace

- “One-click” trace configuration of every day’s tasks
- Context sensitive
- Device independent
- Simplifies trace usage
- Faster results
- Low training effort



RTOS / AUTOSAR trace

- Configuration of task trace (+ program trace) directly from RTOS / AUTOSAR Add-In



UDE Debug and Trace Support for TC4x

Main Cores	Generic Timer Module (GTM)	Parallel Processing Unit (PPU)	Cyber Security Real-Time Module (CSRM)	Stand-By Controller (SCR)
▶ Architecture ▪ TriCore TC1.8 ▶ Supported Languages ▪ C, C++, Rust, Assembler ▶ Run-Control ▪ HW and SW Breakpoints ▪ Single stepping ▪ Synchronization via UDE Run-Control Group ▶ Trace ▪ Program, Data, Watchpoints, etc. ▪ Parallel trace of all cores (as selected)	▶ Architecture ▪ Bosch GTM 4.1 ▶ Supported Languages ▪ C, Assembler ▶ Run-Control ▪ HW Breakpoints ▪ Single stepping ▪ Synchronization via UDE Run-Control Group ▶ Trace ▪ MCS code trace ▪ Data trace ▪ Trace of signals	▶ Architecture ▪ Synopsys ARC EV ▶ Supported Languages ▪ C, C++, Assembler ▶ Run-Control ▪ HW and SW Breakpoints ▪ Single stepping ▪ Synchronization via UDE Run-Control Group ▶ Trace ▪ Program trace	▶ Architecture ▪ TriCore TC1.8 ▶ Supported Languages ▪ C, C++, Assembler ▶ Run-Control ▪ HW and SW Breakpoints ▪ Single stepping ▪ Synchronization via UDE Run-Control Group	▶ Architecture ▪ XC800 (8-bit) ▶ Supported Languages ▪ C, Assembler ▶ Run-Control ▪ HW and SW Breakpoints ▪ Single stepping ▶ Trace ▪ Not supported by HW ▶ Debug Interface ▪ Separate SCR DAP I/F connected to some P33 port pins ▪ Main SoC DAP port (also used for AURIX debugging)

pls Development Tools

Any Questions?

ETAS HIGHTEC imagimob An Infineon Technologies Company
Infineon pls Development Tools TRUST IN SOFT