Timing Analysis for Multicore ECUs
Agenda

Options for TA tool suite
TA Designing
Trace Log Measurements
Summary
Q&A
Multicore Effects are challenging the software architecture and integration

Introducing

Defining a software architecture and integrating it in a way that ensures:

- Deadline fulfillment
- Load balancing
- Guarantee of system constraints and requirements
- Reduce cross-core communication overhead
- Reduce memory access overhead and blocking effects
- Reduce OS Overheads
Solution

TA Tool Suite in the development process

**TATS:: ECU SW Integration**
- Define Timing Requirements & Constraints
- Analyze timing behavior (by simulation) of software architecture
- Evaluate and optimize timing & performance behavior (by simulation and model analysis)

**TATS:: ECU SW Verification**
- Verify target behavior for fulfilling specified timing requirements
- Understand timing behavior by interactive visualizations
- Evaluate timing and performance properties for project management
Solution

TA Tool Suite Tools Interfaces - SW Integration

Use Case: SW Integration with target

- SWC Detailed Design
- DaVinci Developer
- RTE & BSW Configuration
- DaVinci Configurator Pro
- Define Requirements, Visualise & Analyse SW architecture (TA.Design)
- Simulate ECU Model (TA.Simulation)
- Measurement Execution Times
- Measurement Runtime (AMD RTM)
- CANoe

Integration phase
Verification phase
Vector tool
Use Case: SW Verification
Current status of the product

Timing Architects Tool Suite Interfaces with other Tools

- **SWC Detailed Design**
  - PREEvision

- **RTE & BSW Configuration**
  - DaVinci Configurator Pro

- **Define Requirements, Visualise & Analyse SW architecture**
  - **TA.Design**
    - System Extract of System Description
    - ECU Configuration Description / Direct Interface

- **Simulate ECU Model**
  - **TA.Simulation**
    - Update ECU Configuration Description

- **Verify ECU target behavior**
  - **TA.Inspection**

- **Measurement Execution Times & Activation**

- **Measurement and Testing tools**
  - **Trace (btf)**
  - **CANoe**

- **RTE & BSW Configuration**
  - **3rd party tool**

Integration phase
Verification phase
Vector tool
3rd party tool
TA.Design enables design and statical analysis of software application

- Visualize functional composition and communication flow
- Define and visualize requirements, event-chains and constraints
- Define and verify data age constraint potential violation
TA.Simulation enables simulation for timing and performance analysis

- Schedulability analysis
  - Consideration of HW- and OS- effects
- Verification of timing requirements fulfillment
  - Cause-effect analysis
- In-depth system design evaluation of timing behavior and resource efforts
  - Forecast analysis of system utilization and dimensioning of the hardware
TA.Inspection enables:

- Verification of system implementation via ECU hardware measurement
  - Hardware-based measurements verification
  - Comparison on system implementation and simulated results
  - Enrich timing model with (specific) time information from traces

Import of traces format:
- Best Trace Format (BTF)
- iSYSTEM Traces
- Lauterbach Traces
- PLS Traces
- GLIWA OT1
Solution

Direct Interface to DaVinci Configurator Pro

- Workflow enables the automation of the several steps with the TA Options
  - Provides direct interface to DaVinci Configurator Pro
  - Job nodes allow different processing of the timing model (e.g. simulation, evaluation, trace import etc.)
  - Also accessible via command line

Example workflow for creation of Timing architecture model directly from DaVinci project, completing with runtime information. The model is simulated and evaluated and a configured Evaluation Report is generated.
Use-Case 1: Design of dynamic software architecture

**Goal**
Timing evaluation and analysis of software architecture during early design phase for efficient development of robust multi-core architecture

**Solution**
Evaluate and analyze an application model according to defined timing requirements under different system utilization scenarios

**Benefits**
Requirements engineering for tasks, ISR and timing event-chains
Early recognition of system timing behavior problems and possibility for their repair
Use-Case 1 Solution: Model Creation

TA Tool Suite Selected Use-Cases

- *.CSV
  - Part of SW application description

- System/ECU-C
  - Part of SW application and core allocation

- Trace/Profiling
  - Runtime information

Static TA timing model
Dynamic TA timing model

TA Tool Suite model creation

TA Tool Suite

TA Explorer – Workflow Editor
Use-Case 1 Solution: Timing Requirements

- A threshold for a quality criteria/metric
  - E.g. Deadline – upper limit of task response time

- Imported via API-import interfaces in the TA Tool Suite

- TA Explorer allows tool-based creation and modification of requirements and event-chain
Timing Architecture Model (TAM)

- Composition of Timing Architecture Model
- TA Tool Suite support import/export of AUTOSAR (>4.x), ASAM MDX, AMALTEA-model, OSEK OIL, OT1-model and CSV files format
TA Tool Suite Selected Use-Cases

Use-Case 2 Solution: Evaluation and Verification Results

- TA Simulation graphical presentation
Overview

1. Configure MICROSAR BSW with Tracing Hooks (OS and RTE)
2. Configure the Trace Debugger
3. Import Trace into TA Tool Suite (iSYSTEM, Lauterbach, PLS and GLIWA)
4. Evaluate the Traces against Requirements

Completely automatable for Regression Tests and Troubleshooting
Use-Case 1 Solution: Evaluation and Verification Results

- **Stress test analysis example**
  - Analysis of timing event-chains duration and requirement fulfillment for different load utilization levels
  - Load utilization increased by increasing the runtimes of some of the most loaded tasks
  - Identification of system design improvements
Example with iSYSTEM

1. Configure MICROSAR BSW - Tracing Hooks OS and RTE
2. Configure iSYSTEM Trace Debugger - iSYSTEM Profiler XML for OS Awareness
3. Import Trace into TA Tool Suite
4. Evaluate the Traces against Requirements

Completely automatable for Regression Tests and Troubleshooting
Use-Case 3: Automated Target Verification

**Goal**
Automatically evaluate and verify the implemented system for fulfillment of the specified timing requirements

**Solution**
Evaluate and analyzing trace measurements according to formalized system requirements

**Benefits**
Complete automation of the trace verification process into the build process
Recognition of system timing behavior problems and possibility for their repair
Verification of software application regarding system utilization of resources
Detect problems by reporting evaluation results directly to the Jenkins build server via Junit test
Use-Case 2: Overview

- Specification of event-chains and definition of their timing requirements
- Automated evaluation and requirement verification
- Integration of TA Inspector into Jenkins build environment (via console API)
- HTML and XML reporting for internal quality audits
Automation – Workflow Editor (Editing)

- Workflow enables the automation of the several steps with the TA Options
  - Job nodes allow different processing of the timing architecture model
  - Also accessible via command line
  - Provides direct interface to DaVinci Configurator Pro

Example workflow for creation of Timing architecture model directly from DaVinci project, completing with runtime information. The model is simulated and evaluated and a configured Evaluation Report is generated.
Summary

- Timing analysis for multi core ECU systems
- Increased flexibility of SW partitioning in multi core systems
- SW integration in multi core systems more complex
- Efficient verification of partitioning decisions necessary after SW integration