Saber
Physical Modeling & Simulation

Lee Johnson
Saber Product Line
Synopsys, Inc.
The Challenge…

Electrification
Electrifying Automotive Systems

Driving the Need for Physical Modeling & Simulation

- System design includes integration of multiple physical domains – interactions cannot be ignored
- Increasing SW / HW interdependencies – earlier development requires platforms for mechatronic design
- Product quality & reliability remain critical – just more difficult to predict and manage
- Supply chain collaboration is a key to success – increased dependence on CAE tools & methods
Sabre Product Line

Physical Modeling & Simulation for Multi-Domain Systems

Sabre is...

A Proven Platform for Modeling & Simulating Physical Systems

Circuit & System Design
Modeling & Characterization
Simulation & Test
Analysis & Reporting

Sabre helps to...

Accelerate Electrification of Physical Systems

Increase HW Quality & Reduce Prototyping Iterations

Connect the Physical System to the rest of Electronics Design
Physical Modeling & Simulation

What is it?
• Lumped-element, conserved energy descriptions of multi-domain behaviors interacting as part of a system
  – e.g. an Electric Machine: electrical + magnetic + mechanical + thermal interactions within a single "component"

How is it used?
• Proven approach for predicting and optimizing the behavior of complex dynamic systems, without decoupling physical domains

Why is it important?
• Increasing demands on content, integration, efficiency, reliability, cost, time to market...physical prototyping alone is becoming intractable
Saber Tackles System Complexity
Unmatched Power Electronic & Electromechanical Simulation

Use standard languages & extensive behavioral libraries for modeling power electronics, electric machines, batteries and more...

Test, integrate, & optimize the complete multi-domain system – the path to higher reliability and robustness

Verify & calibrate embedded SW through integrations to C-code, Synopsys Virtual Prototypes and Simulink

History of success in design, integration and model exchange for electro-* systems
Supporting the Design of Real Systems

**Saber Differentiation**

### Modeling

**Base Set:**
- Behavioral Language(s)
- Multi-Domain Modeling
- Generic Model Libraries
- Modeling Assistants

**+ Saber:**
- Component Libraries
- Input Format Compatibility
- Component Characterization
- FE / Field Solver Extraction

### Simulation / Analysis

**Base Set:**
- Time-Domain Performance
- Freq-Domain Performance
- Scripting / Automation
- Embedded SW Connections

**+ Saber:**
- Design Optimization
- Design for Robustness
- Design for Reliability
- Grid / Parallel Computing

### Environment / Usability

**Base Set:**
- Windows-based IDE
- Documentation & Examples
- Support & Community
- Industry Standards

**+ Saber:**
- Linux, Unix Support
- Leading Performance
- Leading Robustness
- Supply Chain Success

© Synopsys 2011
Saber Strengths
Connecting Electronic Design with Physical Systems

Model Inputs
- MAST
- VHDL-AMS
- SPICE
- C/C++
- Simulink
- IBIS
- S-parameters

Circuit & System Design
- TCAD
  • Synopsys

Modeling & Characterization

Simulation & Test

Analysis & Reporting

Saber

Distributed Computing
- EST
- ChiasTek
- others…

Embedded SW / Algorithm
- Synopsys
- Mathworks
- others…

PCB
- Zuken
- Mentor
- others…

Wire Harness
- Synopsys
- Zuken
- Mentor
- others…

Digital Verification
- Synopsys
- Mentor
- others…

HW Integration & Test
- NI
- others…

EM & Multiphysics
- CST
- Infolytica
- others…

© Synopsys 2011
Automotive ECU Development

Conventional Flow

Virtual Environment

- Control Algorithm
- Functional Validation

Physical Environment

- Target SW
- HIL
- Physical Validation

- Plant Model
- Electronic Control Unit
- Vehicle System
Expanding Virtual ECU Development

- Synopsys solutions expand the virtual environment for ECU development
- Enables earlier integration, calibration and validation…without physical HW
Virtual Hardware-in-the-Loop

Virtual Prototypes + Saber

- SW integration, analysis, and debug—with a high-fidelity physical system model
- Enables greater test coverage → increased robustness and reliability
  - Account for system variability, faults, worst case
Saber Summary

• Proven solution for Physical Modeling & Simulation—addressing complexities of electrification & multi-domain integration

• Broad coverage in linking electronic design to the physical system

• Committed to customer partnership to advance power components through systems
Engineering Safe Mobile Systems
The omnipresent challenge for automotive engineering

Dr. Graham Hellestrand
CEO & Founder
What We will Cover in this Presentation

- Safety and Functional Safety in Mobile Systems and ISO 26262 Safety Standard
- High fidelity modelling with high performance simulation of mobile systems
  - The enabling modelling & simulation Eco-system exists and is on display today
- Verification of ECU + plant, Subsystems, Systems, and Systems of systems
  - Concurrent with design
- Example: High Fidelity Vehicle models simulated in a Traffic Scenario
- High effectiveness & efficiency of the model-driven system engineering process
  - Must drive the entire engineering supply chain
The ISO 26262 standard mandates govern automotive systems
- No harm to humans
- No damage to property, and
- No harm to the environment that causes harm to humans

Two Types of Safety:
- **Functional Safety** – monitoring of primary functions & timing (typ. Post engineering fix)
- **Safety** – intrinsic part of Requirements (eg. fault tolerance in autonomous control systems)

Automotive Functions/subsystems with potential to cause harm & damage
- Autonomous and semi-autonomous control
- Synchronization of communication between ECUs, Subsystems and Vehicles
- Any system producing emissions (engine, brakes, tyres), consuming energy

Sources with potential to cause harm to the Environment & thus to Humans
- Extraction, processing & transport of energy and materials
- Emissions – gaseous & particulate
Safe Mobile Systems Engineering
High Fidelity Models of Entire Vehicles, High Performance Simulation
Conceiving and Safely Engineering Mobile Systems Through Ecosystems - Partnerships & Interoperability

Kaizen Safety, Quality Reliability & Economy:
Interacting Cyber-Physical Systems

A. Modelling & Simulation
B. Networks
C. Communication Systems
D. Traffic Plant + Controls
E. Vehicle Plant + Controls
F. DoE and Optimization
G. Engineering Process

PARTNERS INTEROPERABILITY

Mathematical ODE/DAE/FSM/…

SYNOPSYS

© Synopsys 2011
Interoperability of Heterogeneous Models & Tools
ESSE Workbench: Any Model, Any Simulator, Any Tool

On Linux and Windows:
• Any Model + Simulator integrated into the ESSE DSE-SNF system
• Any Tool is supported by the ESSE system
ESSE Scalable, Distributed Simulation Engine vs Conventional Centralized Simulation Engine

Conventional Centralized Simulation Engine (all ECU/Plant models on a single core)
- Measured

ESSE Distributed Simulation Engine (multi-core, 1 ECU/Plant model per core)
- Measured
- Predicted
- Variance

Total Wall-Clock Simulation Time (seconds)

Number of ECU/Plant Models in Distributed System under Simulation
The ESSE – Systems Simulator
High Fidelity Systems Modelling and High Speed Systems Simulation

Integrated Traffic Simulation & Visualization
3D Map Data Base
Traffic Sensor & Controller
Visualization: Vehicles → Map

Route & Traffic Infrastructure Data
Vehicle Data

Vehicle Model
Transmission ECU Model
Braking ECU Model
Driver Model
Model of Driver Brain

Engine Model
Transmission Model
Brake Model

Vehicle Data
3D Map Data Base

ESSE Distributed Simulation Engines

© Synopsys 2011
 EST - Engineering Safe Mobile Systems
Modelling a Large City’s Infrastructure: Traffic: Infrastructure, Controllers, Energy & Emissions
The ESSE System – ITS Simulator
Large-scale Integrated Modelling & Simulation of Mobile Systems

100-1,000 full Vehicle models
1,000-50,000 GhostCar models
On 100-1,000 cores
Verification: ECU+Plant & Subsystems
Multi-ECU Virtual HIL System

ESSE CAN NETWORK

Vehicle Speed

Synopsys (VaST) Transmission Controller

Gear

ESSE Signals Network

Vehicle Speed

InvirTest

All Data

Vehicle Plant Model
(engine, transmission, ...)

Synopsys (VaST) Door-Lock Controller

Vehicle Speed

Spd > 20

Lock

InvirTech: Verification Results

Vehicle Speed

RPM

RPM

Static Tests & Results Repository
Safety & Mobility
Wirelessly Connected Vehicles – can they be safe?

V-ECU Transmission Controller
- Memory
- UART
- CAN Controller

Virtual Electronic Controller
(Synopsys/VaST + C++/SystemC)

Timer

Memory

CAN Network

DSRC

SILS GIPPS Driver + Radar + DSRC + Adaptive Cruise Ctrl + Collision Warning (Saber)

CarSim Vehicle Model

Signals Network

Radar

6-cars radar

6-cars radar + WiFi

Simulation

Simulation

Simulation

Simulation

Simulation

Simulation
Modelling & Simulation of Safe Mobile Systems
High Fidelity Vehicle, Control & DSRC Network Models + Experimental Active Cruise Control and GIPPS Driver Models

Legend
- **Red**: RADAR detects vehicle in front
- **Blue**: WiFi communication with vehicle in front, even when RADAR does not detect it.

(CarSim Visualization)
System Verification via Scenarios
ISO 26262 Compliance

Golden (Mathematical) Reference Vehicle Models

Operational Vehicle Models

Function & Timing Difference Detection & Analysis (Reference vs Operational)

ESSE Distributed Simulation Engine

Difference Reporting, Diagnosis, Analysis & Display

InvirTest: Verification Results

Scenario Selector

Traffic Scenario – Convoy Safety

Traffic Scenario – Highway Safety

Function Differences

Timing Differences

Responses

Stimuli

Report

Tests

Static & Fault Injection Tests

InvirTest

12 lanes, bi-directional traffic, 36m cars 1 lane broken, 3km, 80 km/h, 4 min. simulation
The Messages

- ISO 26262 standard is necessary and will govern automotive engineering
  - Bosch has committed to compliance for all new controllers after Q1, 2011
- ISO 26262 is inherently a system standard
  - All real-time control subsystems, as well as the system (vehicle) are subject to this standard
- Most efficient system (supply chain) engineering process is model-based
  - Executable Specification & system tests:  OEM → Tier1 → Tier2
  - Design verified to the specification:  Tier2 → Tier1 → OEM
- Model-based system engineering Eco-System exists – and works well
  - Today’s Synopsys sponsored seminar is evidence of this
- Scenario- with static test-based verification of vehicles in traffic
  - EST demonstrated this as a 1st in the world capability 2 years ago
  - Concurrent and regression style verification – with software development and hardware design
  - Selective fault injection, Forensic fault diagnosis
<table>
<thead>
<tr>
<th>Region</th>
<th>Company</th>
<th>Telephone</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Embedded Systems Technology Inc. (Si Valley, California)</td>
<td>+1-650-488-4571</td>
</tr>
<tr>
<td>Japan</td>
<td>Advanced Data Controls Corp (Tokyo, Japan)</td>
<td>+81-3-3576-5351</td>
</tr>
<tr>
<td>Australia</td>
<td>Embedded Systems Technology Pty Ltd (Sydney, Australia)</td>
<td>+61-2-9908-1227</td>
</tr>
</tbody>
</table>

Thank you Synopsys for being a Major Systems EcoSystem Player
Battery Pack Management Simulation in Electric Vehicle Using Saber

Rashed S. Rabaa
Senior Lead Engineer
Agenda

1. GM Li-Ion Battery Pack
2. Saber in the GM Design Flow
3. Virtual Hardware Model
4. Analysis objectives
Lithium Pack Battery Features

- Electronics
  - Used as Voltage Balancers
- Battery
- DC Cable
- Cell groups separated by thermal Fins
- Charge Port
How does Saber fit into our design flow?

Build…
Simulate…
Analyze…

**Build** Multiphysics Virtual Hardware Model

**Simulate** Performance

**Analyze** Results

Power Inverters

Li Cells

Discharge characteristics of 100Ah Li-ion cell.
Why Saber?

There are a myriad of tools… So why use Saber?

- Engineer-2-engineer… It is the best tool in its class:
  - Robust advanced mixed-technology mixed-signal simulator
  - Includes MAST (and VHDL-AMS) HDLs
  - Supports a large generic and component model library
  - Supports Simulink model to MAST model export tool
  - Supports Saber-Simulink cosimulation
  - Supports PSPICE model to MAST model import
  - Supports many model characterization & development tools (IGBTs, MOSFETs, diodes, motors, transformers, custom models, etc.)
Saber was the right tool for creating a Lithium Pack Battery model

Complex, large multi-hierarchical system

Spans multiple technologies such as
- Digital
- Electrical
- Thermal
- Control

Contains many non-linear elements
Can’t ignore software interaction

Model includes virtual hardware interaction with software

- Exported Simulink control model to Saber to capture the cells’ algorithms and calibrations
Battery simulation provides important output characteristics

**Typical Input**

<table>
<thead>
<tr>
<th>Drive Cycle: <strong>US06, NEDC</strong> <em>(NEFZ in German)</em></th>
<th>Initial Temperature</th>
<th>Initial SOC</th>
<th>Number or Li cells in any given pack</th>
<th>Maximum capacity (A-hrs)</th>
</tr>
</thead>
</table>

**Simulation**

**Typical Output**

<table>
<thead>
<tr>
<th>Voltage &amp; Current esp. variation across every cell</th>
<th>Power Consumption</th>
<th>Energy Consumption</th>
<th>SOC behavior</th>
<th>Temperature</th>
</tr>
</thead>
</table>

Battery simulation provides important output characteristics.
Voltage variations across cells based on different SOC

Battery pack
- Voltage & current
- Power
- Energy
Virtual prototyping opens doors to safe and efficient “what-if” investigation

**Intermediate-term** goals:
- Drive-cycle experiments
- Fault Analyses: hard and soft shorts
- Worst-case environmental temperature impacts
- Etc.
Saber enables system simulation

**Long-term** (big picture) goals:
- Saber has the model libraries, capability, & robustness to permit complete system study!
VIDEO

Chevrolet
V O L T
Saber enables GM virtual prototyping success with power electronics & machines

Hybrid / Electric Vehicle analysis requires a potent power electronics simulation tool

Power Electronics & Machines = Saber’s sweet spot
Acknowledgement

- John Novak (GMNA)
- Qinwei Sun (GMPT)
- Bryan Kelly (Synopsys)
- William C. Goodwin (GMPT)
- Mary Fortier (GMNA)
Thank you for your Attention!
Questions?
Challenges and Opportunities for Multicore Software Development in Powertrain and HEV Applications

Carl Bonfiglio, Senior Segment Marketing Manager, HEV and EV

Synopsys Automotive Solutions Seminar
November 10th, 2011
Introducing the new focus area "Mobility" reflects:

- Our leadership position in Automotive
- Rising importance of new mobility concepts (e.g. electro mobility) and
- Innovative public transportation solutions for traction & electronic tickets
Infineon Automotive Division

Product range

- **Sensors**: pressure, temperature, magnetic; wireless control ICs, radar
- **Microcontrollers**: 8-bit, 16-bit, 32-bit
- **Power**: MOSFETs, IGBTs, smart power ICs: voltage regulators, bridges, driver ICs, CAN / LIN / FlexRay™ transceiver, DC-DC converter, power system ICs, system-on-chip embedded power ICs
- **Hybrid & Electric Vehicle**: HybridPACK™1, HybridPACK™2, gate driver ICs, MOSFETs, IGBTs

Core competencies/Value proposition

- Fully **automotive commitment**: More than 40 years of automotive system and application expertise
- **Complete** automotive system provider
- **Hybrid and E-Mobility**: industry leading expertise and product portfolio
- **Worldwide** development, production and support sites for automotive semiconductors
- **Automotive Excellence**: most comprehensive quality program of the industry

Market positions

- **No. 1** in Automotive semiconductors worldwide
- **No. 1** in Europe
- **No. 2** in NAFTA
- **No. 2** in ROW

Source: Strategy Analytics (April 2010)

*FlexRay is a trademark licensed by FlexRay Consortium GbR*
Why Multicore?

Safety and Performance for Multicore

Multicore Software Development

Summary / Concluding remarks
Global CO₂ Targets

CAFÉ May 2\textsuperscript{nd} 2008
Cars + LD: 35 MPG by 2016

EU Dec. 17\textsuperscript{th} 2008
Cars: 95gCO₂/km by 2020

Overall tax: Car, Fuel and CO₂
- On the Fuel (≈90€ per gCO₂/km)
- On the Car (≈30€ per gCO₂/km)
- On the OEM (≈95€ per gCO₂/km)
- On the CO₂ (≈20€ per tone CO₂)

Car running 10 years
16,000 km/y or 10,000 miles/y

Conversion table for regular gasoline engine

<table>
<thead>
<tr>
<th>gCO₂/km</th>
<th>155</th>
<th>140</th>
<th>130</th>
<th>120</th>
<th>110</th>
<th>100</th>
<th>95</th>
</tr>
</thead>
<tbody>
<tr>
<td>L / 100km</td>
<td>6.72</td>
<td>6.08</td>
<td>5.65</td>
<td>5.21</td>
<td>4.78</td>
<td>4.34</td>
<td>4.13</td>
</tr>
<tr>
<td>MPG</td>
<td>35.00</td>
<td>38.69</td>
<td>41.66</td>
<td>45.13</td>
<td>49.24</td>
<td>54.16</td>
<td>57.01</td>
</tr>
</tbody>
</table>

http://www.wtrq.com/daily/crudeoilprice.html
Efficient mobility
Move from 160 gCO₂/km to 95 gCO₂/km

- Engine: 30%
- Hybridization
- Transmission: 12%
- Energy Management, Efficiency, On demand: 1-2%
- Friction reduction, TPMS: 2-3%
- Air drag reduction (S*Cv)

- Potential to reduce CO₂ by 45%
- Further CO₂ saving can be achieved by vehicle weight reduction, traffic management improvement and modified driving behavior.
Software-Enabled Functionality
Increased Microcontroller Performance

Semiconductor Industry provides a 30% to 60% annual performance increase at same cost

- Software platform, reuse of software modules across application and customers
- Migration of functions from hardware to software
- Hardware independent Software
- Wider use of automatic code generation
- Software standardization e.g. Operating system, Drivers with application level interfaces (OSEK, AutoSar, IEC61508, ISO26262...)
- Robust, transparent software e.g. encapsulation, software self test
- µC family concept with performance increase and easy migration path
Microcontroller Technologies

- ITRS CMOS logic
- MPU M1 half-pitch roadmap
- International Technology Roadmap of Semiconductors

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology (nm)</td>
<td>180</td>
<td>130</td>
<td>90</td>
<td>65</td>
<td>45</td>
<td>32</td>
<td>22</td>
<td>16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Automotive µC technology roadmap
- The smaller is the technology:
  - Processed Wafer cost
  - More expensive is mask cost
  - More sensitive to radiation
  - Higher is leakage current
  - Analogue do not shrink at same rate (Flash, ADC, Pad…)

Confidential
Microcontroller Performance Evolution

Performance benchmark based on the application code for a 4 cylinder direct injection engine management system

TC1767 100% 100%
TC1797 105% 113%
TC179x 113% 136%
TC1793 136% 152%
TC1798 152% 170%
Estimation 1xTC1.6E & 1xTC1.6P
Estimation 2x TC1.6P

Dual core utilization factor = 1.5
ISO26262 Requirements

Automotive Safety Integrity Level D (ASIL D) requires:
- Single point fault metric > 99%
- Latent fault metric (multiple faults) > 90%

Requires at least 2 computations of every function which has ASIL D safety goal:
- Multicores allow redundant calculations
  - Requires an independent comparator
- Lockstep cores implicitly have redundant calculation and comparator
Presentation Outline

- Why Multicore?
- Safety and Performance for Multicore
- Multicore Software Development
- Summary / Concluding remarks
Homogeneous Multicore?

CPU Type A

L1 MEM

CPU Type A

XBAR

FLASH w/ ECC

DEBUG ACCESS

SRAM w/ ECC

IRQ

SCU

BRIDGE

PER1

PER2

PERn

PERm
Heterogeneous Multicore?

DEBUG ACCESS

L1 MEM

CPU Type A

CPU Type B

XBAR

SRAM w/ ECC

FLASH w/ ECC

SCU

BRIDGE

PER1

PER2

PERn

PERm

IRQ
Dual Core Lockstep – Not Multicore?

- CPU Type A
- Peripheral CPU
- L1 MEM
- FLASH w/ ECC
- SRAM w/ ECC
- SCU
- BRIDGE
- PER1
- PER2
- PERn
- PERm

DEBUG ACCESS

IRQ
Need for ‘Safe’ Processing

- New 65nm technology can allow lockstep on several CPUs
  - Allows AutoSAR to run on Peripheral processor
  - Application processor is offloaded from real-time tasks
Low Cost ‘Safe’ Processing

■ For systems where only a small ASIL D processing load
  ‣ E.g. ETC

■ ASIL D tasks only run on peripheral core

■ Performance core is speed optimised
  ‣ E.g. engine controls
Presentation Outline

- Why Multicore?
- Safety and Performance for Multicore
- Multicore Software Development
- Summary / Concluding remarks
**SW issues & challenges**

- SW deadlocks, race conditions, priority inversion, non-reentrant code
- HW/SW Contention for shared resources, e.g. on shared memory allocations
- Cache trashing
- System Real Time behavior
- OSEK / AUTOSAR RTOS implementation
- Tools supporting Dual Core Architecture
  → e.g. OS task partitioning in order to optimize load balancing and synchronization overhead
- ...

---

**Multi Core Architecture Definition**

Challenge for Application SW

- EBU
  - PMU0
  - PMU1
  - SFI Bridge
  - DMA

SRI XBAR
Interconnect

6 Masters, 8 Slaves

Challenge for Microcontroller HW
Development Tool Chain Overview for the TriCore family

**SW Drivers, Libraries**
SW modules that will be utilized in the Application (Low Level driver for peripherals (incl AUTSAR), DSP Lib, Communication driver (CAN, TCPIP,..))

**Operating System**
Control SW that will be linked with the application. Main Functions: Task Mgmt. Interrupt Mgmt, Memory Mgmt

**Auto Coding**
Tool that generates application SW based on higher abstraction layers (i.e. graphical GUI)

**Compiler IDE**
Integrated Development Environment consisting of: Editor, C and C++ Cross Compiler, Assembler, Linker, Locator and partly including OS, Instruction Set Simulator and Source-Level Debugger.

**Programmer**
HW /SW to program the embedded Flash

**Programming Services**
Services to program microcontroller

**C-Model based simulation**
SW Models (C model) of the Microcontroller integrated in a simulation frameworks to test applications SW without HW

**Calibration**
Optimization of parameters (tables) of the app. SW

**Debugger**
Consist of a SW/HW system designed to help locate and correct bugs within a target application. Instead, the Debugger can be connected to a simulation engine.

**Starter Kits**
Evaluation boards including ready to run demonstration software

**Verification Tools**
Test environments to be used to perform functional and formal tests towards functional safety requirements
TriCore Family Tool Partner Vendors

- Multi-Core Tool Chain

- Multi Core Tool Chain

- C-Code Generation [Auto Coding]
- Integrated Compiler Tools
- Dual Core Device, Demonstrator or Simulator
  - Scheduling Analysis
  - Load Analysis
  - Debugger
- Static Dependency Analyzers
- Formal Verifiers
- WCET Worst Case Execution Time Analyzer
- Operating System
Multicore Software Development

- Multicore Software is not new
- Multicore for real-time automotive applications is new
- One possible approach for mapping runnables and data to cores and memories
  - Considering explicitly timing constraints and performance
  - Based upon a model for efficient exploration
  - With explicit support of AUTOSAR
- Metrics to compare different software mappings

- How to get multi-core on the road reliably and affordable?
Some assumptions for an automotive multi-core system

- Heterogeneous systems have advantages in cost and power consumption vs homogenous systems.

- Scheduling analysis on chip will become more complicated.

- The goal is not the ‘biggest’ multi-core system, but the system best suited for real time automotive applications including sophisticated multi-core debugging and tracing.
Challenges in Moving to Multicore

- Legacy software

- New hardware requires new compiler and new OS version

- Architectures are not always directly comparable to predecessors due to different clock rates, connectivity and memory characteristics

- Software mapping to cores uses new communication paths

- **Many changes at once! The practice: Why is the performance not as expected?**

- **Solution for investigation:**
  - Use a model based approach with explicit consideration of timing
  - Use communication cost catalogue
Modeling and Simulation Tools

- System Engineering Tools for Simulation and Modeling of Embedded Systems
  - Virtual Processor Models for Infineon TriCore devices
  - METeortm: software developer seat for development of software based on a virtual processor model
  - CoMET: hardware architecture developer seat (e.g. ECU development), includes functionality of METeortm.
  - Metrixtm: offers insight into the behavior and performance of the hardware and software components of virtual system prototypes
  - Peripheral Device Builder (PDB): rapid creation of peripheral devices
  - Synopsys TriCore® Partner:
    - Lauterbach
    - PLS
    - Altium
    - Hightec
Simulation Support and Virtual Prototyping for TriCore

- Core models from Synopsys, for usage in Synopsys CoMET and METeor Virtual Prototyping solution
- Peripherals usually modeled in SystemC
- Available platforms:
  - Current TriCore platforms available (all peripherals except FlexRay and MLI)
  - TriCore Multicore platform with one reference to TC1.6P, two references to TC1.6E (TC1.6P will be used as an approximation to TC1.6E), interrupt system, PMU, LMU, SCU, SRI and bridges, Ports, STM, DMAs, MSC, SMU, GTM model v1.3, QSPI, ASCLIN, HSM and tested tool chain (compiler debugger, available only directly from tool vendor)
  - Further Multicore peripheral models are subject to clarification
Fast and accurately timed simulation in COMET and METEOR

- Simulator can be used like real silicon in interaction with debugger (Lauterbach and PLS tested)
- Coupling of own SystemC models (e.g. for stimuli generation) possible
- The Multicore C-model platform is also being used strongly by the Infineon MCAL developers
- Generation of vcd files easily possible
- Coupling to Simulink available
- MCD API supported for TC1.6P core model from April 2011 on
- Simulation speed usually 1.. 50 MHz for one core with peripherals, strongly depending on use case
Development flow for multi-core software development

- **Hardware meta-model with ECU, core, memory types**
- **Software meta model with runnables and tasks**
- **Generic communication benchmark**
- **Model of the application with mapping of runnables, tasks and variables**
- **Application metric, supported with dedicated charts in SymTA/S**
- **Timing information about computational load**
- **Application with mapping of runnables, tasks and variables**
- **Verification**
- **Hardware or virtual prototype**

**SYMTA VISION**

Automatic import
MCDS – Multicore Debug System

**MCDS Goals**
- Non-intrusive observation
- Watching all cores and busses concurrently
- System state extraction
- Tracing
- Profiling
- Real-time visibility of Program Flow, Data Flow
- Performance Measurements

**Abbreviations:**
- **POB:** Processor Observation Block
- **BOB:** Bus Observation Block
- **DMC:** Debug Memory Controller
- **MCX:** Multi Core Cross-Connect
- **DAP:** Device Access Port
- **EMEM:** Emulation Extension Chip SRAM

**MCDS Goals**
- Non-intrusive observation
- Watching all cores and busses concurrently
- System state extraction
- Tracing
- Profiling
- Real-time visibility of Program Flow, Data Flow
- Performance Measurements
Conclusion

- Powertrain and xEV are growing in complexity and require an increasing amount of microcontroller complexity.

- Semiconductor vendors are using multicore architectures to meet the challenges.

- The biggest challenge in the adoption of multicore is software development.

- Advances in development tools is the single most important factor to achieve efficient multicore software.
Innovative semiconductor solutions for energy efficiency, mobility and security
Virtual HIL and Process Integration
Virtual Testing of Embedded Software
Robert Kiessel, CEO
Invirtech, LLC
Purpose

• Thank you to Synopsys for hosting this seminar

• Overview of an Industry first Virtual Testing Product – InvirTest – and review capabilities and benefits.

• Discussion/ demo of recent evolution of InvirTest to include multi-ECU and CAN message capability (John and Dan)
Our Premise

- Software is now THE defining technology in Automotive – cars are now an electronic product.
- The company that leads in SW development capability will lead the industry (feature innovation, speed & quality)
- Software has grown into a disruptive technology which is causing the breakdown of traditional approaches
- Successfully meeting this challenge requires both a multi-faceted approach and effective integration between these approaches
- We’ve Developed a Solution To Help You Tackle the Some of the Significant Challenges in this arena.
Invirtech Solutions for:

- Development & Testing of Embedded Control Systems—with InvirTest
  - Implements a Virtual Lab for the validation & verification of distributed embedded control systems
  - Puts Industry-leading products in the hands of typically skilled engineers.
- Your Partners with Supporting Services –
  - Embedded systems development consultation
  - Integration of development & PLM systems
- We bring over 100 years of experience in automotive software development management & IT expertise.
InvirTest - Purpose

Product Development V

- Specification
- MBD
- MIL
- Auto Code Generation
- Limited HIL
- Reduced Vehicle Testing

Virtual Lab
Virtual HIL Supporting Embedded Software Verification & Validation At Subsystem & System Levels
InvirTest – What is it?

- A Virtual Lab supporting all components for Virtual HIL test

- Provide an easy-to-use **interface** which simplifies the complexities of underlying integrated tools (Synopsys Virtualizer aka CoMET, Mathworks Simulink)
InvirTest—Operation

- ECU
- Plant Model
- Target Code
- Test Parameters

Auto-Build ➔ Execute Tests ➔ Analyze
InvirTest - Benefits

• Earlier SW testing even before HW is available
• More efficient & extensive testing in a lower cost environment
• Easy-to-use interface allows typical product or test engineers to run tests. This permits specialists to focus on modeling & setup for improved efficiency.
• Weekly or daily build allows more frequent testing—shortens cycle of issue identification
• Earlier software maturity, improved quality & reduced development and launch costs
InvirTest – Benefits (continued)

• Easy replication for a global workforce & supply base/value chain for efficient shared development

• Easily resurrected test environment to permit quick testing of in-cycle changes. (30 sec vs. 3 mos.)

• Increased visibility & control with repeatable results for easier debugging.

• Given the growth of distributed functions and interactive systems, a more thorough approach to managing test complexity.
InvirTest – Features

• Automated test wiring, setup, execution and report-generation.
• Automated import of requirements
• Auto-generation of test vectors
• Automated signal extraction from plant & hardware models
• Activity-based User Interface—streamline UI to suit the task being performed (typical vs. power user).
Invirtech – The Product
Product Evolution Shown Today!

- InvirTest extension into Multi-ECU capability with EST’s ESSE is demonstrated today.
- We have industry partnerships to ensure compatible, integrated products (Our host--Synopsys, and EST).
Multi ECU Configuration

InvirTest Environment

Test

Distributed Function Subsystem

Test

Ad Hoc

? 

Test

ECU

Results

Test

ECU

Results
Today’s Demo - Auto Door Lock

InvirTest/ESSE Environment

ESSE CAN BUS
- Vehicle Speed
- Vehicle Speed

Vehicle Plant Model
- Vehicle Speed
- Vehicle Speed
- Vehicle Speed
- Vehicle Speed

InvirTest
- All
- Throttle

Trans Controller
- Speed
- Gear

Body Controller
- Door Lock Command

If speed > 20;
Lock Door
Our Invitation

We are ready to engage with your team:

– Demonstrate InvirTest as a solution for your project
– Partner with you to ensure InvirTest integrates with your development process

Are there any questions?
Appendix
Principals

• Robert Kiessel – Chief Executive Officer
  – 30 years at Ford
  – Chief Engineer, Electrical & Electronic Systems Engineering
  – MBA, MS-Electrical Engineering, BS- Electrical Engineering from University of Michigan

• Martin Baker – Advisor and Partner
  – 26 years at Ford
  – Senior Global Manager, Embedded Software, Modeling, CAE, and Electrical/Electronic Process, Methods, Tools and Information
  – MBA, MSEE
Principals

• Dan Wilczak – Chief Technical Officer
  – 30 years at Ford
  – Manager CAE Systems – Vehicle Structures, Test Data and Engineering Systems, Test Systems development and implementation
  – BS Computer and Communications Sciences

• Mike Conley – Chief Information Officer
  – 17 years at Ford
  – IT technical and solutions architect for Ford finance systems, CAD application and data management systems, and Project leader for test data acquisition and management systems
  – MBA, BS in Mathematics
Principals

• John Stawarz – Chief Product Strategy Officer
  – 35 years at Ford
  – Strategic Electrical Portfolio Manager, CAD Systems project manager, Test data systems development and implementation
  – BSME, BS – Physics, BA – Music
Invirtech – Our Customers’ Challenge

1. What to test

- Product
- Specifications
- Requirements
- Test Plan
- Procedures
- Test Vectors

2. Plan the test

- ECU / ECU Model
- Plant Model
- Assemble Test Environment
- Software
- Wiring
- Applications
- Parameters

3. Assemble test environment

4. Run test and analyze results

- Run Test
- Analyze Results
- Save

5. Iterate as needed

- Data Intensive
- Data is usually dispersed in multiple systems
- Data is of varied type
- Requires application specific knowledge
InvirTest – Capabilities

• **Data Management:**
  – Specifications
  – Requirements
  – Test plans
  – Test procedures and vectors
  – Target software
  – Parameters

• **Model Management:**
  – ECU Models
  – Plant models

• **Test Management:**
  – Automated Wiring and Setup
  – Execution
  – Results

• **Interoperability Services**
  – Parameter and Specification Import
  – PDM/PLM connectivity
InvirTest – What is it?

• **Virtual Lab** supporting all components for Virtual HIL test
• **Data manager** supporting parameters, ECU hardware models, plant models, embedded software, test vectors, test procedures, and test configuration
• **Test manager** supporting automated test wiring, setup, execution, and report generation
• An **easy-to-use, single interface** that abstracts & simplifies the complexities of underlying integrated tools (Synopsys/CoMET, Mathworks/Simulink)
• Designed for typically skilled Test/D&R Engineers
Virtual Testing Dataflow

Product Info
- Features
- Requirements
- Specs
- Data Dictionary
- Signals
- etc.
Invirtech – Product Features

Test Management/Execution
- Execute appropriate mode (Automated batch, Regression, Interactive/Ad-Hoc, Environmental)
- Manage Test Execution
- Capture and report results

Auto-generate Test Environment
- Auto-generate Test Vectors
- Assemble Drive and Procedure Files
- Assemble and Network Virtual ECUs
- Define Test Control and Management
- Select Target Software
- Select Plant Models
- Auto-configure the Test Rig, i.e., virtual wiring

Assemble Feature/Specification Pair
- Requirements Management
- Data and State Flow Diagrams
- Assignment of Hardware, Plant Models
- Assignment of Target Software
- Assignment of Parameters

Model Management
- Virtual Hardware Models
- Plant Models
- Software Models
- Advocate Models

Parameter Management
- Data/Signals
- DTCodes
- Test, Procedure, Drive
- Operational and System
InvirTest Flow Diagram

InvirTest virtual testing dataflow

ECU model creation

ECU simulation

Test execution

Test setup and control

Plant model simulation

Data management

Target code

Test definition

Analysis and reports

InvirTest

coMET

Simulink