“Living in a Virtual World” – Microcontroller and Full System Simulation
Renesas Technology & Solution Portfolio

Enabling the Smart Society

Microcontrollers
No.1 Market Share Worldwide

Advanced and Proven Technologies

System LSIs

Extensive, High-quality Portfolio

Analog & Power
Microcontroller and Microprocessor Line-up

2010

1200 DMIPS, Superscalar
- Automotive & Industrial, 65nm
  - 600μA/MHz, 1.5μA standby

500 DMIPS, Low Power
- Automotive & Industrial, 90nm
  - 600μA/MHz, 1.5μA standby

165 DMIPS, FPU, DSC
- Industrial, 90nm
  - 242μA/MHz, 0.2μA standby

25 DMIPS, Low Power
- Industrial & Automotive, 150nm
  - 190μA/MHz, 0.3μA standby

10 DMIPS, Capacitive Touch
- Industrial & Automotive, 130nm
  - 350μA/MHz, 1μA standby

2013

1200 DMIPS, Performance
- Automotive, 40nm
  - 500μA/MHz, 35μA deep standby

165 DMIPS, FPU, DSC
- Industrial, 40nm
  - 242μA/MHz, 0.2μA standby

Embedded Security, ASSP
- Industrial, 90nm
  - 1mA/MHz, 100μA standby

44 DMIPS, True Low Power
- Industrial & Automotive, 130nm
  - 144μA/MHz, 0.2μA standby
Modeling & Simulation is **Necessity** Not Luxury

“The Space Station consists of 70 separate major components and hundreds of minor ones, all of which will be assembled for the first time in space.” pbs.org

Intel Core 2 Duo
291M transistors

PCB circuit simulation
Today’s Situation Requires Modeling & Simulation

- More products which are more complicated
- Fewer resources
- More software
- Shorter development cycles
- Dislocated development teams
- More safety critical features
Developing the Smart Society Products

- Microcontrollers have made gigantic leaps in performance. Requirements on engineers have escalated on many fronts. Debugging and developing require smarter methods.

- The Solution is to leverage microprocessor and full system modeling and simulation to provide more capability and increase efficiency. Renesas has a proven record, and is pursuing modeling and simulation environments on many micros.
Agenda

- Debug History of High End Embedded Microcontrollers
- Exploiting Simulation and Modeling with Microcontrollers
- Benefitting from Simulation & Modeling
- The Rest of the Story
- Carpe Diem
Debug History of High End Embedded Microcontrollers
The Beginning

- Microcontrollers
  - Single core
  - 16 bit datapath
  - < 40MHz clock
  - < 500K code
  - Address/data bus usually available

- Primary Debug Methods/Tools
  - Instrumented code, pins, LEDs
  - ROM monitor
  - Run control debuggers
  - Scope

- Limitations
  - Cannot determine exact execution path unless single stepping
  - Cannot determine read/write data unless single stepping
  - Very difficult to see hardware triggers and events
  - Impossible to cause internal faults
  - Debug process interferes with core execution
The Last Decade

Microcontrollers
- 1-2 cores
- 32 bit datapath
- < 150MHz clock
- < 2M code
- Address/data bus sometimes available

Primary Debug Methods/Tools
- Run control debuggers
- Logic analyzers
- Trace debuggers
- Scope

Limitations
- Exact execution path requires reconstruction from trace data, might have gaps
- Determination of most read/write data requires single stepping
- Very difficult to see hardware triggers and events
- Impossible to cause internal faults
- Debug process may interfere with core execution
The Present Difficulties

- Microcontrollers
  - 3+ dual issue cores
  - 32 bit datapath
  - > 300MHz clock
  - > 6M code with safety critical software
  - Address/data bus often not available

- Primary Debug Methods/Tools – more of the same
  - Run control debuggers
  - Dual logic analyzers
  - Trace debuggers
  - Multiple scopes

- Limitations – no improvements
  - Exact execution path requires reconstruction from trace data, might have gaps
  - Determination of most read/write data requires single stepping
  - Very difficult to see hardware triggers and events
  - Impossible to cause internal faults
  - Debug process may interfere with core execution
Yet Even More Control/Performance is Required

- In-cylinder pressure transducers
  - Optimized combustion events
- Finer motor control
  - Improved efficiency
- Faster, better graphics
  - Realistic, 3D vision, 4 displays

234 billion texture fills / sec?!
300W?!
Exploiting Modeling & Simulation
You Get it All

- See every detail clearly
  - See every node
  - No trace reconstruction

- Precise control and operation
  - Execution not modified by instrumentation
  - Events and faults perfectly timed

- No need for instrumentation
  - No short circuits
  - No mislabeling
  - No channel limits
Why Not Now?  (Rhetorical)

- Proven Tools Exist
  - ASTC’s Vworks / VLAB simulators
  - Synopsys’ Virtualizer virtual prototyping
  - EST’s high performance system simulation
  - Accellera modelling standards and tools (formerly Open System C Initiative)
  - Imperas simulation and analysis tools

- Fast Simulation Times
  - Instruction set simulation faster than real time
  - Cycle accurate micro simulation ~100:1

- Good Business Cases
  - New structuring for reasonable tool costs
  - Semiconductor companies including core and peripheral models upfront

ASTC, Synopsys and Imperas are here today!

Attend the simulation courses, labs, demos at DevCon!
Benefiting from Modeling & Simulation
Complete Visibility

- Every internal node is accessible on every clock
  - All data, even internal to micro, is available to speed understanding and debug

- Full, synchronized program, data and I/O trace
  - No gaps or lost synchronization
  - No limits due to micro or tool packaging
  - All data all the time (EDPN: Entertainment and Data Programming Network)

- Full software task / function analysis with zero code instrumentation and zero overhead
  - Instant analysis with any software
Perfect Consistency

- Simulations and tests are always exactly duplicated
  - Faster debug and controlled testing
- Automation and regression tests are simpler and more controlled
  - Faster test and verification development
  - Faster release verification
- Better code coverage analysis from consistent fault injection
  - Faster verification because of predictable behavior
Fast Simulation

- **Exact** target micro instruction set is simulated
  - No unique processes or compilers required

- Target instruction set is translated to x86 (native Intel PC) equivalent instructions
  - Instruction set simulation can execute *faster than real time*

- Cool, vendor specific innovations
  - Cycle accurate simulation can *execute at ~100:1* on typical PC

- I/O Channel models are behavioral level, event based models
  - I/O adds *minimal simulation time*
Hardware Independent Development

- More development can be done concurrently
  - Software development can be done earlier
- Only software and a PC is required
  - Development environment is completely portable
- Software is more mature, earlier
  - Fewer ECU hardware redesigns
- Hardware development benches only required for verification in unpredictable, real world situations
  - Less development cost
  - Hardware/software comparisons built without designing/building physical hardware
  - Architecture variations can be easily evaluated
Easy, Global Fault Injection

- Faults can be injected both external and internal to ECU
  - White box testing and black box testing are simple
- Faults can be injected internal to micro
  - Only method for testing fault tolerant, error handling code
- Any signal/value can be changed at any time
  - Only method for verify safety critical operation
Portable Environment

- Any (newer) PC in the world can run simulations
  - Dislocated development teams can use coherent environment
  - Issues can be easily duplicated anywhere
- Environment is entirely software
  - Environment can be tracked with version control
  - Environment can be globally distributed overnight
  - No (ECU and instrumentation) hardware interaction issues
Quick System Configuration Changes

- Easily change micro configuration
  - Execute “curiosity” experiments requiring additional flash, additional RAM, different cache, or high clock speed **not available on micro**
  - Execute investigative experiments with I/O channel from another micro

- Easily change ECU configuration
  - Evaluate throughput/latency of sharing a serial bus instead of separated busses implemented in the ECU
  - Evaluate performance of digital/analog expanders
Reduced Development Costs

- **Shorter development cycles**
  - Faster to market and better return on investment
- **Fewer hardware development benches**
  - Less to purchase, less to maintain, less to upgrade
- **Fewer evaluation/investigation ECU designs**
  - Less capital investment
- **Fewer ECU redesigns**
  - Less waste, less re-engineering
Shorter Development Schedules

- Software development done independently of hardware
  - More development can be done concurrently
- Development environment coherent and easily deployed
  - Concurrent, lock-step development
- Complete visibility and perfect repeatability
  - More complete analysis leads to better decisions
  - More efficient debug
  - Better optimizations
- Simple, improved regression testing
  - Find issues sooner
The Rest of the Story
Reasons Not to use Modeling

- Models are sometimes unavailable
- Modeling only as good as models
- Model test vectors often not the same as silicon test vectors
- HW (PCB) development is cheap and easy
- Improved trace capabilities has made current debugging methods reasonably good
Carpe Diem
Developing the Smart Society Products

- Embedded microcontroller debug and development can be far more complicated and challenging, with many external constraints.
- Simulation and Modeling can provide significant advantages to ease the pain:
  - Complete visibility & Perfect consistency
  - Fast simulation & Hardware independent development
  - Easy, global fault injection with a Portable environment
  - Quick system configuration changes
  - Reduced development costs & Shorter development schedules
- Full ECU Simulation is Realistic:
  - Full engine controllers are already being simulated
  - Full cell phones are already being simulated
- Modeling and Simulation are commonplace, not cutting edge:
  - “Be there, or be square”
  - “Seize the day”
Questions?