Model-Based Design of Embedded Signal Processing Systems Using Simulink®

Altera SOPC World 2004

<Name of presenter here>
Agenda

- Model-Based Design of Embedded Systems
  - Challenges in DSP system design

- Simulink and Blocksets
  - Quick Simulink demo
  - Video surveillance demo

- Hardware Implementations
  - Implementation on DSPs and Altera FPGAs
The MathWorks at a Glance

Headquarters: Natick, Massachusetts USA

USA: California, Michigan, Washington DC, Texas

Europe: UK, France, Germany, Switzerland, Italy, Spain, Benelux

Asia-Pacific: Korea

Worldwide training and consulting

Distributors in 25 countries

Earth’s topography on an equidistant cylindrical projection, created with the MATLAB Mapping Toolbox
Key Industries

Core
- Aerospace and Defense
- Automotive
- Communications, Electronics, Semiconductor, Computers and Office Equipment
- Education

Emerging
- Biotech, Pharmaceutical and Medical
- Financial Services
- Industrial Equipment and Machinery
- Instrumentation

Ongoing
- Chemical and Petroleum
- Earth and Ocean Sciences
- Utilities and Energy
The MathWorks Product Family

DAQ cards
Instruments
Databases

Blocksets
Stateflow

SIMULINK

MATLAB

Core Platform for Signal Processing System Design

Code Generation for Rapid Prototyping and Production Deployment

Desktop Applications Automated Reports
Challenges in Embedded System Design

- Handle Increasing complexity
- Design team integration
- Reduce Time-to-Market
Problems with Traditional Development

Requirements and Specifications
- Text-based
  - Prevents rapid iteration

Design
- Physical prototypes
  - Incomplete and expensive

Implementation
- Manual coding
  - Introduces human error

Test and Verification
- Traditional testing
  - Errors found too late in the process
Advantages of Model-Based Design

Requirements and Specifications
- Unambiguous
- Only “one truth”

Design
- Simulation
  - Reduces “real” prototypes
  - Systematic “what-if” analysis

Implementation
- Automatic code generation
  - Minimizes coding errors

Test and Verification
- Test with Design
  - Detects errors earlier

Executable models
- Continuous verification

Model elaboration
Model-Based Design with Simulink
Model-Based Design Allows You to Overcome Design Challenges:

- **Handle Design Complexity**
  - Reuse IP, Simulate at high speeds, collaborate with multiple design teams, utilize system-level design abstraction

- **Design team integration**
  - Analog/Mixed-Signal, digital hardware, DSP S/W, control S/W designed in one model
  - Co-design and partition HW and SW components

- **Reduce Time-to-Market**
  - Generate code automatically for HW and SW
  - Accelerate verification using executable specification
# User stories: RealTek

RealTek Gains 50% of Market Share with a New Audio Chip Designed with MathWorks Tools

<table>
<thead>
<tr>
<th>Challenge</th>
<th>To unify different engineering disciplines on a single development platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution</td>
<td>Standardize on MathWorks tools to streamline the design process and enable analog and DSP designers to work together</td>
</tr>
</tbody>
</table>
| Results            | • 50% market share in first year of product release.  
                    | • High return on investment.  
                    | • Improved collaboration and reduced design time. |

**Products Used**
- MATLAB
- Simulink
- DSP Blockset
- Fixed-Point Blockset
- Optimization Toolbox
- Signal Processing Toolbox
Simulink Tutorial: Model Construction

- Drag and drop
- Connect

- Digital
  - Fast frame-based simulation

- Analog
  - Variable-step numerical integration solvers
  - Zero-crossing detection
Signal Processing Blockset

- Streaming data
- Multi-rate systems
- Transforms, filters, estimators
- Enables frames in Simulink
- Fixed- and Floating-Point Support

Example 1 - Envelope Detection

```matlab
>> dspwwvlib.mdl
>>dspwwv
```
Example 2 - Video Surveillance Systems using the Video and Image Processing Blockset

Fixed-point video surveillance system based on Sum of Absolute Differences motion detection
Example 2: Video Surveillance System

Source Frames
High rate: 30 fps

Recorded Frames
Aperiodic rate (<< 30 fps)

Total # of Recorded Frames

>> vipsurveillance_win32
Motion Estimation: Sum of Absolute Differences

Compare Threshold

Record frames & update count

Estimate motion

Motion Estimation:
Sum of Absolute Differences
Example 2: Video Surveillance System

Source Frames
High rate: 30 fps

Recorded Frames
Aperiodic rate (<< 30 fps)

Total # of Recorded Frames

>> vipsurveillance_win32
Motion Estimation: Sum of Absolute Differences

Estimate motion

Compare Threshold

Record frames & update count
Motion Detection with Thresholding

- Motion levels
- Video Motion Estimate
- Detection Threshold
- Captured Video Frames

The diagram shows a graph with red stars indicating motion events. The x-axis represents time, and the y-axis represents motion levels. The threshold is set at a certain value, and the captured video frames are highlighted.
Embedding Signal Processing Applications on DSPs and FPGAs
Production Code Generation

- Based on Real-Time Workshop code generation engine
- Generated code is ANSI C – efficient, readable, editable
- Supports and utilizes Real-Time Workshop Embedded Coder

- Proven automatic code generation technology for critical applications
Production Code Generation
Successful in Automotive and Aerospace industries

“Visteon Powertrain has demonstrated that model-based software development can generate quality software in less time, and the automatic code ROM & RAM sizes are equal to or better than hand written code.”

Table 1: Code Size comparison between a fixed-point hand code and auto code.

<table>
<thead>
<tr>
<th></th>
<th>Code Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Code</td>
<td>926</td>
</tr>
<tr>
<td>Auto Code</td>
<td></td>
</tr>
<tr>
<td>No overflow/underflow check</td>
<td>904</td>
</tr>
<tr>
<td>Check OF/UE everywhere</td>
<td>1562</td>
</tr>
<tr>
<td>Check only where necessary</td>
<td>934</td>
</tr>
</tbody>
</table>

*Based on Tasking Compiler for ST110

Table 2 ROM and RAM comparison between a floating-point hand code and auto code.

<table>
<thead>
<tr>
<th></th>
<th>Hand Code</th>
<th>Auto Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>6408</td>
<td>6192</td>
</tr>
<tr>
<td>RAM</td>
<td>132</td>
<td>112</td>
</tr>
</tbody>
</table>

* Multi-Target Modeling for Embedded Software Development for Automotive Applications
Grantley Hodge, Jian Ye and Walt Stuart, Visteon Corporation
2004 SAE World Congress, Detroit, MI. March 8-11, 2004
Embedded Target for TI C6000 DSP is…

... 1. a tool for production code generation
   - Processor-specific, optimized
   - Simulink blocks and optimized libraries (FIR, FFT, …)

... 2. a means for project automation
   - Processor-specific, automatic
   - APIs for CCS IDE, Compiler/Linker

... 3. a platform for rapid prototyping
   - Target-specific, integrated
   - Simulink hardware blocks and device drivers (ADC, DAC, RTDX, daughter cards)
1. Production Code Generation

a) How good is the generated code for TI C6000?

- Code generation philosophy for C6000 DSPs:
  - Generates efficient, portable, readable, editable code
  - Supports code profiler to help identify code performance bottlenecks
    - code segments that provide highest return on optimization
1. Production Code Generation

b) How can I further optimize the code?

- Provides alternate methods for code optimization
  - Manual optimization by user
    Click on link in profile report to jump to relevant code section
  - Target-specific blocks
  - Engineering services
1. Production Code Generation

Target-Specific Blocks

- C-callable assembler libraries
  - Simulate bit-true in Simulink
  - Generate calls to hand-optimized assembler libraries
  - Highly optimized implementation of core functionality
- C62x and C64x fixed-point DSPs
1. Production Code Generation

- Recap of code generation with Embedded Target for TI C6000
  - Quickly create a complete, working code base
  - Utilize code profiler to help identify any performance bottlenecks in generated code
  - Choose from several approaches for code optimization

...optimize only when and where necessary...
2. Project Automation

- Create and populate CCS project
- Automate compile/link/download
2. Project Automation

- Utilize CCS to:
  - debug, test, and verify code
  - add and customize code

![Breakpoints and Watch Window](image)
# 3. Rapid Prototyping – Scenarios

<table>
<thead>
<tr>
<th>Use Supported DSK/EVM#</th>
<th>Use Custom Board + Emulator</th>
<th>Use TI DSP Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. PC + DSK/EVM</td>
<td>1. PC + Custom Board + Emulator</td>
<td>1. PC</td>
</tr>
<tr>
<td>2. Emulator Optional</td>
<td>2. Connect through JTAG port</td>
<td>2. Simulator: Very Slow</td>
</tr>
<tr>
<td>3. Connect through USB, Parallel, PCI, or JTAG ports</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Generate code  
2. H-I-L prototyping*  
3. RTDX  
4. Access hw devices – ADC, DAC, daughter cards, etc.

• Test and verify in CCS  
• Test and verify using MATLAB and Link for CCS

1. Generate code  
2. H-I-L prototyping*  
3. RTDX

• Test and verify in CCS  
• Test and verify using MATLAB and Link for CCS

1. Generate code  
2. S-I-L prototyping*  

• Test and verify in CCS  
• Test and verify using MATLAB and Link for CCS

# Supports 6701 EVM, 6711, 6713, and 6416 DSKs. In v.2.1, DM642 EVM will also be supported.

* From MATLAB
Steps to Target the TI C6416 DSK

**RTDX blocks**

**DIP Switch**

>> c6416dsksurveil_hsrtdx
Select target options (DSP/BIOS, compiler settings, etc)

- High-Speed RTDX
- Incorporate DSP-BIOS
- Inline Signal Processing Blockset functions
Steps to Target the TI C6416 DSK – 3

- Build process
  - Auto-generate ANSI C and ASM code
  - Integration of RTOS and scheduler
  - Create full CCS project in IDE Invoke compiler, linker, and download code
  - Run target
Target the TI C6416 DSK – 4

- Automatic profiling of program executing on DSP

System profiling
Includes entire DSP application code

Subsystem profiling

```matlab
c=ccsdsp;
profile(cc,'report')
```
Design Verification: Real-time Visualization

- Host-side visualization using Link for Code Composer Studio

Log and plot estimates over time (scrolling data)

Input video frames

Captured frames

>> c6416dsksurvey_hsrtdx
Embedding
Signal Processing Systems in Altera FPGAs Using the DSP Builder
DSP Builder

- Creates HDL Code
- Creates Simulation Test Bench
- HDL Synthesis
- Provides Model Technology

- Download Design to Development Board
- Verify in Hardware
- Creates Process or Plug-In

- QUARTUS® II
- Synplicity
- SignalTap® II

The MathWorks
MATLAB & SIMULINK
Custom Co-Processor Development

**SOPC Builder**

- Embedded Processor
- Memory

**DSP Builder**

- Embedded Processor
- IQ Map
- NCO
- Memory

- Processor + Co-Processor

- Dedicated Hardware Architecture
DSP Builder Library Components

- Arithmetic
- Bus Manipulation
- Complex Signals
- Logical Components
- SOPC Ports
- Storage
- MegaCore® IP
- Rate Change
- State Machine
- Altera Library
- DSP Board
Other MathWorks Products for HDL Code Generation and Verification
Filter Design HDL Coder for Digital Filters

- **Description**
  - Design IIR fixed-point filter
  - Generate synthesizable VHDL or Verilog
  - Verify implementation through co-simulation

- **What you will see**
  - Fixed-point filter design
  - Automatic HDL generation with Filter Design HDL Coder
  - Verify implementation with Link for ModelSim
Link for ModelSim

Generate Synthesizable VHDL or Verilog code

Co-simulate with ModelSim

Fixed-point support across MATLAB and Simulink

Design Fixed-Point Filters
Simulink and Model-Based Design Produce Results Across Industries

TOYOTA

Standard for Powertrain Controls Production Code Development

LOCKHEED MARTIN

JSF Flight Control System

zyray wireless

W-CDMA Baseband Processors

TEXAS INSTRUMENTS

Specialty Chipsets for DSP Customers
MATLAB Central

- www.matlabcentral.com
- Over 1,500 MathWorks- and user-contributed files
- MATLAB files and Simulink models for download
Summary

- Simulink brings:
  - Model-Based Design to large-scale projects
  - More comprehensive coverage of embedded system development
  - New domains and applications

- Visit the web for more
  www.mathworks.com/r14

- Visit us at the booth to see more product demonstrations