Communication System Design for Software Defined Radio

Mike McLernon
Manager, Communications Development
The MathWorks
June 15, 2006
Outline

Overview of Software Defined Radio (SDR)
- Working Definition of SDR
- MathWorks Activities in SDR
- Design Flows for SDR Development

MathWorks Tools for SDR Designs
- Demo: SDR Reference Waveform – FM3TR
  - Simulations
  - Fixed-Point System Design
  - Automatic Code Generation
- Example Design Flows for Target Platforms

Conclusion
A Working Definition of SDR

- SDR is a collection of hardware and software technologies that enable reconfigurable system architectures for wireless networks and user terminals.
- Embedded, portable, reusable software
  - Across diverse software and hardware platforms
  - Across teams, projects, and in time
  - For multistandard support
  - For reduction of development cost and time
- Defense industry driving the technology via JTRS
The MathWorks Activities in SDR

- Active member of the SDR Forum since 2002
- Participation in work groups:
  - Design Process and Tools
  - Hardware Abstraction Layer
  - Commercial Technology
- Delivered workshops on code portability and embeddable transceiver code generation
Design Flows for SDR Development
Traditional Design Flow

- **Client Requirements**
  - Specification
  - Simulation results

- **Test**
  - Test spec
  - Simulation test vectors
  - Client test vectors

- **Implementation**
  - Code
  - System docs

- **Design**
  - SW design tool

- Completed system

The MathWorks Aerospace & Defense Conference 2006
Waveform Design (by SDR Forum Tools Work Group)

- Requirements Specifications (MS Word), MAT Reference Designs
- Requirements Feedback: (Visio Diagram, PowerPoint, Excel, MS Word)
- Feed Forward: (Spectral Masks, BER Performance, Processing Requirements)
- Platform Definition: (Specification, Device Configuration, Data Transport)
- Platform Requirements

Core Technologies can feed each stage

- Visio Diagrams, UML Models, Connectivity Specification (MS Word)
- Algorithms (MS Word), MATLAB M-files, Simulink® files
- Preliminary Design Docs (MS Word, Visio, Excel), Prototypes, UML Models
- Assembly Description, C/C++ Source Code, “Golden Waveform”
- Detailed Design Docs (MS Word, Visio, Excel), UML Models
- Waveform Partitioning (Deployment Model)
- FPGA Code
- DSP Code
- GPP Code
- VHDL, Bit Files, Pin Out Specifications, Timing Diagrams
- C/C++/Assembly Object Code, Timing Specifications
- C/C++ Object Code, Timing Specifications
- Real-Time Waveform Implementation Real-Time Debugging

Technologies Demonstrations
- Acceptance Test Reports
- Requirements Management System, Traceability Matrix: DOORS

Bug Reports
Overview of Model-Based Design

Links to paper-based specs

Iterate, Iterate, Iterate!

Floating point to Fixed point

Import your own custom code
BAE Systems Achieves 80% Reduction in Software-Defined Radio Development Time with Model-Based Design

The Challenge
To develop a military standard SDR waveform for satellite communications

The Solution
Use Simulink and Xilinx System Generator to rapidly design, debug, and automatically generate code for an SDR signal processing chain

The Results
- Project development time reduced by 80%
- Problems found and eliminated faster
- Clocking and interfacing simplified

“Using Simulink and Xilinx System Generator we designed and developed the signal processing chain of the SDR and achieved a 10-to-1 reduction in development time.”

Dr. David Haessig,
BAE Systems
One SCA Rapid Development Project

- Develop glue code and wrappers that encapsulate signal processing code and interface it to SCA core framework
- Automatic code generation using MathWorks Real-Time Workshop® and Xilinx System Generator™
- TI Code Composer Studio™
- Zeligsoft CE for Automatic XML Profile Generation
- BAE expertise in SCA
- Virginia Tech Glue Code Development
Demo: SDR Reference Waveform – FM3TR
**FM3TR Reference Waveform**

- Future Multiband, Multiwaveform, Modular, Tactical Radio Waveform (Reference waveform for SDR Forum)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency range</td>
<td>30-400 kHz</td>
</tr>
<tr>
<td>Channel spacing</td>
<td>25 kHz</td>
</tr>
<tr>
<td>Modulation type</td>
<td>CPFSK</td>
</tr>
<tr>
<td>Modulation rate</td>
<td>25 kbps</td>
</tr>
<tr>
<td>Frequency hopping</td>
<td>250-500 hops/second</td>
</tr>
<tr>
<td>Framing, packetization</td>
<td>Switched, packet</td>
</tr>
<tr>
<td>CVSD voice coder</td>
<td>16 kbps</td>
</tr>
<tr>
<td>Coding</td>
<td>Reed-Solomon</td>
</tr>
</tbody>
</table>
The MathWorks FM3TR
Software Defined Radio Example

- Design, simulation, performance analysis
- Fixed-point analysis
- Automatic code generation
- Flexible system partitioning into components
Software Defined Radio Example

Software Defined Radio - FM3TR Reference Waveform
Fixed-Point MSK Modulation/Demodulation

MSK Rimoldi-Style Modulator/Demodulator

References:
SDR FSDRF-01-10056-V0.00
Description of the FM3TR Proposed Reference Waveform
Modulation Format

H. Leib and S. Pasupathy, "Error-Control Properties of MSK,"
IEEE Communications Magazine, Jan. 1993

B. Rimoldi, "A Decomposition Approach to CPM,"
IEEE Transactions on Information Theory, Mar. 1998
Fixed-Point System Design
Fixed-Point Design in Simulink®

- **Simulink Fixed Point**
  - Enables fixed-point support in:
    - Simulink
    - Signal Processing Blockset
    - Stateflow®

- **Fixed-point settings at:**
  - Block level
  - Subsystem level
  - Model level

- **Control over**
  - Inputs
  - Outputs
  - Internal values
Tool: Fixed-Point Settings

![Fixed-Point Settings](image)

- **Select current system:** MSK Demodulator1
- **Logging mode:** Min, max and overflow
- **Data type override:** Use local settings

**Simulation data logged for current system**

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Min</th>
<th>Max</th>
<th>Data Type</th>
<th>Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix Sum : Accumulator</td>
<td>-69.2</td>
<td>61.39</td>
<td>S32</td>
<td>V=Q*2^-11</td>
</tr>
<tr>
<td>Matrix Sum : Output</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add1</td>
<td>-32</td>
<td>32</td>
<td>S16</td>
<td>V=Q*2^-10</td>
</tr>
<tr>
<td>Product1</td>
<td>-14.13</td>
<td>13.99</td>
<td>S32</td>
<td>V=Q*2^-26</td>
</tr>
<tr>
<td>Data Type Conversion1</td>
<td>-13.23</td>
<td>13.99</td>
<td>S16</td>
<td>V=Q*2^-11</td>
</tr>
<tr>
<td>Matrix Sum1 : Accumulator</td>
<td>-69.88</td>
<td>75.37</td>
<td>S32</td>
<td>V=Q*2^-11</td>
</tr>
<tr>
<td>Matrix Sum1 : Output</td>
<td>-66.88</td>
<td>75.37</td>
<td>S16</td>
<td>V=Q*2^-8</td>
</tr>
<tr>
<td>Add2</td>
<td>-102.8</td>
<td>107.7</td>
<td>S32</td>
<td>V=Q*2^-24</td>
</tr>
<tr>
<td>Data Type Conversion2</td>
<td>-102.8</td>
<td>107.7</td>
<td>S16</td>
<td>V=Q*2^-8</td>
</tr>
<tr>
<td>Data Type Conversion3</td>
<td>-102.8</td>
<td>107.7</td>
<td>S16</td>
<td>V=Q*2^-8</td>
</tr>
</tbody>
</table>

**Logging type:** Overwrite log

**Autoscale current system**

- **Safety margin:** 0.0

The MathWorks Aerospace & Defense Conference | 2006
Fixed-Point Design Aids

- Model annotation
- Histogram techniques
Automatic Code Generation
Code Generation from Models

- Target generic C-programmable devices
- Production Code for embedded systems
  - ROM size, RAM size, Execution Speed: Comparable with optimized handwritten code
  - Compact ERT code format
  - ANSI and ISO floating-point libraries
  - Customized main program: deploy on target with or without OS
  - Supports user-defined data objects and S-functions
  - Detailed HTML Reports
Code Optimization Options
Code Customization Options
Example Design Flows for Target Platforms
DSP Design Flow

Link for Code Composer Studio™, Real-Time Workshop®, Embedded Target for TI C6000™ DSP

Target-Specific and Optimized Code

Code Composer Studio™ (Compiler, Linker, and Loader)

Specific Peripheral Software Drivers

Customer-Specific Board

The MathWorks Aerospace & Defense Conference 2006
Conclusion

Model-Based Design is a crucial methodology for successful design of Software Defined Radios

- Simulations
- Fixed-point design
- Code generation
- Deployment into diverse hardware and software platforms
- Integration with SCA tools