Addressing Fixed Point Design Challenges

Manohar Reddy M
Application Engineer
MathWorks India

Manohar.Reddy@mathworks.in
## Fixed Point Design Challenges

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Floating Point</th>
<th>Fixed Point</th>
<th>Fixed Point with MathWorks Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM Consumption</td>
<td>↑</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Execution Speed</td>
<td>↓</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>Hardware Power Consumption</td>
<td>↑</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Embedded Hardware Cost</td>
<td>↑</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Development Time</td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
</tr>
<tr>
<td>Implementation Complexity</td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
</tr>
<tr>
<td>Error Prone</td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
</tr>
</tbody>
</table>
Fixed-Point Concepts

- What is Fixed-Point?
- Fixed-Point Arithmetic
- Fixed-Point Targets

Introducing Fixed-Point Designer

- Fixed-Point Designer Tools

Floating-Point to Fixed-Point Conversion Workflows

- Using the Fixed-Point Tool in Simulink
- Using the Fixed-Point Converter App in MATLAB

Code Generation from Fixed-Point Design

- C/HDL
Fixed-Point Concepts

- What is Fixed-Point?
- Fixed-Point Arithmetic
- Fixed-Point Targets

Introducing Fixed-Point Designer

- Fixed-Point Designer Tools

Floating-Point to Fixed-Point Conversion Workflows

- Using the Fixed-Point Tool in Simulink
- Using the Fixed-Point Converter App in MATLAB

Code Generation from Fixed-Point Design

- C/HDL
What is Fixed Point?

**Floating-point:**
- Sign bit
- Exponent Width → Range
- Mantissa / Fraction bits → Precision
- Floating Radix / Decimal Point
  - Same word size, we can increase both precision and range

IEEE 754 single-precision binary floating-point format: binary32*

**Fixed-point:**
- Sign bit
- Whole Number Part → Range
- Fractional Part → Precision
- Fixed Radix / Decimal Point
  - Fixed size, **Trade-off between Precision and Range**

\[
\text{value} = (-1)^{\text{sign}} \times \left(1 + \sum_{i=1}^{23} b_i 2^{-i}\right) \times 2^{(e-127)}
\]

*Picture from http://en.wikipedia.org/wiki/Single_precision_floating-point_format*
Fixed Point Arithmetic

\[ \begin{align*}
6 \times 10 &= 60 \\
+ &
\begin{align*}
3.4 \times 10 &= 34 \\
+ &
\end{align*}
\end{align*} \]

\[ \begin{align*}
&+ 6.8 \\
\text{Fixed Point MCU} &+ \\
&34
\end{align*} \]

\[ 102 \div 10 = 10.2 \]
Example: Fixed-Point C hand implementation

```c
void differentialEq( void )
{
    /* Implements a fixed point first order difference equation */

    int Prod;
    long Accum;
    static short lastVal=0;
    short a=0x7eb8; // 0.99 in s16,15
    short oneminusa=0x0148; // .01 in s16,15
    short temp;

    Prod = gAlg_in1 * gAlg_in1;
    temp = Prod >> 15;
    Accord = a*lastVal + oneminusa*temp;
    gAlg_out1 = (short)(Accum >> 15);
    lastVal = gAlg_out1;
}
```

- Convert variables to integer types
- Need lots of comments to understand code
- Keep track of binary point location
Weinmann Develops Life-Saving Transport Ventilator Using Model-Based Design

Challenge
Develop embedded software for an advanced emergency and hospital transport ventilator

Solution
Use MATLAB and Simulink for Model-Based Design to model and simulate the controller, generate production code, and streamline compliance certification

Results
- Code development and reviews accelerated by 50%
- Dozens of design alternatives explored
- 60% of core design reused

“Modeling, simulating, and implementing the ventilator’s embedded software with Simulink greatly simplified compliance certification. The model helped ensure a structured development process and provided thorough documentation and a visual representation of the system for the certification review.”

Dr. Florian Dietz
Weinmann

Link to user story
Fixed-Point Concepts

• What is Fixed-Point?
• Fixed-Point Arithmetic
• Fixed-Point Targets

Introducing Fixed-Point Designer

• Fixed-Point Designer Tools

Floating-Point to Fixed-Point Conversion Workflows

• Using the Fixed-Point Tool in Simulink
• Using the Fixed-Point Converter App in MATLAB

Code Generation from Fixed-Point Design

• C/HDL
Fixed Point ECU Development Process
with Model-Based Design

System Requirements
Simulation
System Design
Floating-Point
Simulation
System Design
Fixed-Point
Rapid Prototyping
Software Design
On-Target Rapid Prototyping
Software Integration
Processor-in-the-Loop Testing
Coding
Production Code Generation
Hardware/Software Integration
Hardware-in-the-Loop Testing
Vehicle Integration & Calibration
Requirements Traceability
Configuration Management
Documentation
Fixed Point ECU Development

Using MathWorks Tools

- MATLAB / SIMULINK
  Algorithm Design

- FIXED-POINT DESIGNER
  Fixed-Point Conversion

- MATLAB CODER

- SIMULINK CODER

- HDL CODER

- EMBEDDED CODER

- C/C++

- VHDL/Verilog

- MCU

- DSP

- FPGA

- ASIC

Using MathWorks Tools
Tools for Converting a Design to Fixed Point

**Fixed-Point Tool for Simulink**
Requires:  
Fixed-Point Designer  
Simulink
Works on:  
Simulink Blocks  
Stateflow Charts

**Fixed-Point Converter App for MATLAB**
Requires:  
Fixed-Point Designer  
MATLAB
Works on:  
MATLAB Code
Fixed-Point Concepts

• What is Fixed-Point?
• Fixed-Point Arithmetic
• Fixed-Point Targets

Introducing Fixed-Point Designer

• Fixed-Point Designer Tools

Floating-Point to Fixed-Point Conversion Workflows

• Using the Fixed-Point Tool in Simulink
• Using the Fixed-Point Converter App in MATLAB

Code Generation from Fixed-Point Design

• C/HDL
Workflow using Fixed-Point Tool in Simulink:

1. Prepare for Fixed-Point Conversion
2. Collect Range Information
3. Propose Data types
4. Apply Data types
5. Compare Results
6. Generate Code
Workflow using Fixed-Point Converter App in MATLAB:

1. Prepare for Fixed-Point Conversion
2. Collect Range Information
3. Propose Data types
4. Generate Fixed-Point MATLAB Code
5. Test Numerics
6. Generate Code
Fixed-Point Concepts

• What is Fixed-Point?
• Fixed-Point Arithmetic
• Fixed-Point Targets

Introducing Fixed-Point Designer

• Fixed-Point Designer Tools

Floating-Point to Fixed-Point Conversion Workflows

• Using the Fixed-Point Tool in Simulink
• Using the Fixed-Point Converter App in MATLAB

Code Generation from Fixed-Point Design

• C/HDL
Code Generation
Automatically generate readable C/HDL code

Automatically generate C/HDL code from MATLAB/Simulink/Stateflow
Summary

- Fixed-Point Designer

- Conversion Workflow
  - MATLAB
  - Simulink

- Code Generation
  - C/HDL
# Key Takeaway

<table>
<thead>
<tr>
<th>Consideration</th>
<th>Floating Point</th>
<th>Fixed Point</th>
<th>Fixed Point with MathWorks Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM Consumption</td>
<td>↑</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Execution Speed</td>
<td>↓</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>Hardware Power Consumption</td>
<td>↑</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Embedded Hardware Cost</td>
<td>↑</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Development Time</td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
</tr>
<tr>
<td>Implementation Complexity</td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
</tr>
<tr>
<td>Error Prone</td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
</tr>
</tbody>
</table>
Training Services

Exploit the full potential of MathWorks products

Flexible delivery options:

- Public training available in several cities
- Onsite training with standard or customized courses
- Web-based training with live, interactive instructor-led courses

More than 30 course offerings:

- Introductory and intermediate training on MATLAB, Simulink, Stateflow, code generation, and Polyspace products
- Specialized courses in control design, signal processing, parallel computing, code generation, communications, financial analysis and other areas.

www.mathworks.in/training
# Public Trainings in the next Few Months

<table>
<thead>
<tr>
<th>Course</th>
<th>Dates</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB Fundamentals</td>
<td>8th – 10th June</td>
<td>Chennai</td>
</tr>
<tr>
<td>Simulink for System and Algorithm Modeling</td>
<td>11th – 12th June</td>
<td>Chennai</td>
</tr>
<tr>
<td>Signal Processing with MATLAB</td>
<td>16th – 17th June</td>
<td>Bangalore</td>
</tr>
<tr>
<td>Image Processing with MATLAB</td>
<td>18th – 19th June</td>
<td>Bangalore</td>
</tr>
<tr>
<td>MATLAB Fundamentals</td>
<td>29th Jun – 01st July</td>
<td>Bangalore</td>
</tr>
<tr>
<td></td>
<td>31st Aug – 2nd Sep</td>
<td>Pune</td>
</tr>
<tr>
<td>Simulink for System and Algorithm Modeling</td>
<td>2nd – 3rd July</td>
<td>Bangalore</td>
</tr>
<tr>
<td></td>
<td>3rd Sep – 4th Sep</td>
<td>Pune</td>
</tr>
<tr>
<td>Statistical Methods in MATLAB</td>
<td>20th – 21st July</td>
<td>Bangalore</td>
</tr>
<tr>
<td>MATLAB based Optimization Techniques</td>
<td>22nd Jul</td>
<td>Bangalore</td>
</tr>
<tr>
<td>Building Interactive Applications with MATLAB</td>
<td>23rd July</td>
<td>Bangalore</td>
</tr>
<tr>
<td>MATLAB to C with MATLAB Coder</td>
<td>3rd – 4th Aug</td>
<td>Bangalore</td>
</tr>
<tr>
<td>Embedded Coder for Production Code Generation</td>
<td>5th – 7th Aug</td>
<td>Bangalore</td>
</tr>
<tr>
<td>Generating HDL Code from Simulink</td>
<td>7th – 8th Sep</td>
<td>Bangalore</td>
</tr>
<tr>
<td>Programming Xilinx Zynq SoCs with MATLAB and Simulink</td>
<td>10th – 11th Sep</td>
<td>Bangalore</td>
</tr>
</tbody>
</table>

Email: training@mathworks.in    URL: [http://www.mathworks.in/services/training](http://www.mathworks.in/services/training)    Phone: 080-6632-6000
Training Certification

- Accelerate professional growth
- Validate proficiency with MATLAB
- Increase productivity and project success

MathWorks Certified MATLAB Associate Examination

Bangalore
29th July & 25th Nov

Pune
3rd June

Recommended Courses
MATLAB Fundamentals (MLBE)

Email: training@mathworks.in
URL: http://in.mathworks.com/#training
Phone: 080-6632-6000