Fixed-Point Design
in
MATLAB and Simulink

Gaurav Dubey
Senior Team Lead - Pilot Engineering
Gaurav.Dubey@mathworks.in
What are you looking for?

- How can I convert an algorithm from floating-point to fixed-point representation?
- How can I comparing the floating-point and fixed-point simulation results?
- How can I generate fixed-point C and HDL code for embedded implementations?
- I don’t know how to convert a floating point model/code to a fixed-point model/code?
- I hope to maintain one algorithm file for all kind of data type, is this possible?
Fixed-Point Concepts

• What is fixed-point?
• Fixed-point targets
• Design Challenges

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 Conversion Tool in MATLAB

Code Generation from fixed-point design

• C and HDL
**What is fixed-point?**

**Floating-point:**
- Sign bit
- Exponent width (determines range)
- Mantissa / Fraction bits (determines 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 (determines range)
- Fractional part (determines precision)
- Fixed radix / decimal point
  - Fixed size, trade-off between precision and range

Fixed-Point Targets

- Fixed-Point DSPs (TI, Analog Devices, etc.)
  - Fixed word lengths
  - Less expensive
  - Use less power - good for battery powered applications
  - Higher clock speeds

- FPGAs (Xilinx, Altera, etc.)
  - Designer can select word lengths
  - Every bit increased word length - more silicon area and increased power usage
# Fixed Point Tradeoffs

<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 and ROM 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>
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;
    Accum = 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
- No saturation or rounding
Fixed-Point Concepts

- What is fixed-point?
- Fixed-point targets
- Design Challenges

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 Conversion Tool in MATLAB

Code Generation from fixed-point design

- C and HDL
Fixed Point ECU Development Process

with Model-Based Design

System Requirements

Simulation

System Design

Floating-Point

Vehicle Integration & Calibration

Hardware-in-the-Loop Testing

Fixed-Point

Hardware/Software Integration

Processor-in-the-Loop Testing

Convert

Software Design

On-Target Rapid Prototyping

Software Integration

Software-in-the-Loop Testing

Coding

Production Code Generation

Requirements Traceability
Configuration Management
Documentation

Rapid Prototyping
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 Conversion Tool for MATLAB
Requires: Fixed-Point Designer
MATLAB Coder
Works on: MATLAB Code

Instrumented Code Generation Report for MATLAB
Requires: Fixed-Point Designer
Works on: MATLAB Code
MATLAB Function Blocks
Fixed-Point Concepts

• What is fixed-point?
• Fixed-point targets
• Design Challenges

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 Conversion Tool in MATLAB

Code Generation from fixed-point design

• C and 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 Conversion Tool 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 targets
- Design Challenges

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 Conversion Tool in MATLAB

Code Generation from fixed-point design

- C and HDL
Code Generation
Automatically generate readable, traceable HDL and C code

Automatically generate C and HDL code from Simulink, MATLAB and Stateflow

Requirements
Full bi-directional traceability!!
Summary

- Fixed-Point Designer
- Conversion Workflow
  - MATLAB/Simulink
- Code generation
  - C and HDL
Fixed Point Conversion: Simulink

Fixed-Point Advisor

- The Fixed-Point Advisor facilitates converting a floating-point model or subsystem into a fixed-point representation.
The Fixed-Point Tool is a graphical user interface that automates specifying fixed-point data types in a model.
Fixed Point Conversion: Simulink

Float- to Fixed- Conversion in FPA and FPT

Integrated workflow to accomplish float- to fixed-point conversion of Simulink models in FPA and FPT

- Streamline workflow from modeling preparation in FPA and autoscaling in FPT
- Support floating-, fixed-point and mixed data type designs
- Emphasize design best-practice to achieve production code generation design goals
- Integrated with features of word length selection, model-wide range analysis and multiple run management support in both FPA and FPT
Fixed Point Conversion: Simulink Output Comparison
Fixed Point Conversion: Simulink

Code Generation
Fixed Point Conversion: MATLAB
Fixed-Point Converter

```
function [y1, y2] = mldlc_kalman_c(z)
2 %#codegen
4 % Copyright 2011 The MathWorks, Inc.
5 %
6 % Initialize state transition matrix
7 At=[1 0 dt 0 0;
8 0 1 0 dt 0;
9 0 0 1 0 dt;
10 0 0 0 1 0;
11 0 0 0 0 1];
14 % Measurement matrix
15 H=[1 0 0 0 0; 0 0 1 0 0];
17 Q=eye(4);
18 R=1000*eye(2);
19 % Initial conditions
20
Variable | Function Replacements
--- | ---
Input | |
| z | 2x1 double |
| y1 | double |
| y2 | double |
| Persistent | |
| p_est | 6x6 double |
```

To compute proposed fixed-point types for variables, use Run Simulation, Compute Derived Ranges, or both.
Fixed Point Conversion: MATLAB
Fixed-Point Converter

Trajectory of object [blue] its Kalman estimate [red]
Fixed Point Conversion: MATLAB
Output Comparison
Fixed Point Code Generation: MATLAB
MATLAB Coder