Optimization and Implementation of Embedded Signal Processing Algorithms

Jonas Rutström
Senior Application Engineer
Two important questions in embedded design...

1. What’s your algorithm?
Two important questions in embedded design...

2. What’s your target?
Targets are very different...
The embedded hardware might be very different...

DSP

MCU

SoC/FPGA
Workflow | Idea to implementation
Workflow | Idea to implementation

Prototyping
Workflow | Idea to implementation

Prototyping
Workflow / Idea to implementation

Prototyping

Production
Workflow | Idea to implementation

Prototyping

Production
Workflow / Idea to implementation

Optimization

Prototyping

Production
Let’s take a step back...
What do we mean with optimization?
Speed | Memory | Readability | Target
Speed | Memory | Readability | Target
---|---|---|---

Fixed point
Optimization at different phases in the development...
Optimization at different phases in the development...

Prototyping

C

SoC/FPGA

Model-Based Design
Optimization at different phases in the development...
Optimization at different phases in the development...

Prototyping

How can I analyze performance and optimize code in MATLAB/Simulink?
Profiling!
Profiling in MATLAB

How long did it take?

- tic/toc

```matlab
% start timer
tic

% execute code
out = myFunction(in);

% stop timer (and store % elapsed time)
et = toc;
```

Where are the bottlenecks?

- profile

```matlab
% turn on profiler
profile on

% execute code
out = myFunction(in);

% turn off profiler
profile off
% open html report
profile report
```
Profiling in MATLAB
Profiling in Simulink

Simulink Profile Report: Summary

Report generated 05-Apr-2016 11:07:29

Total recorded time: 6.49 s
Number of Block Methods: 32
Number of Internal Methods: 5
Number of Model Methods: 11
Clock precision: 0.00000004 s
Clock Speed: 2601 MHz
To write this data as model1ProfileData in the base workspace click here

Function List

<table>
<thead>
<tr>
<th>Name</th>
<th>Time</th>
<th>Calls</th>
<th>Time/call</th>
<th>Self time</th>
<th>Location (must use MATLAB Web Browser to view)</th>
</tr>
</thead>
<tbody>
<tr>
<td>simulate(modell)</td>
<td>0.49964160</td>
<td>160%</td>
<td>1.4856415000</td>
<td>0.06090000</td>
<td>modell</td>
</tr>
<tr>
<td>initializationPhase</td>
<td>1.41801550</td>
<td>37.3%</td>
<td>1.4180155000</td>
<td>1.41801550</td>
<td>modell</td>
</tr>
<tr>
<td>simulationPhase</td>
<td>2.2933214700</td>
<td>52.5%</td>
<td>2.2933214700</td>
<td>0.46580000</td>
<td>modell</td>
</tr>
<tr>
<td>model1.Start</td>
<td>1.24800880</td>
<td>19.2%</td>
<td>1.2480088000</td>
<td>0.00000000</td>
<td>modell</td>
</tr>
<tr>
<td>compileAndLinkPhase</td>
<td>1.17000750</td>
<td>18.0%</td>
<td>1.1700075000</td>
<td>1.17000750</td>
<td>modell</td>
</tr>
<tr>
<td>model1.Outports_Motor</td>
<td>0.951566510</td>
<td>14.7%</td>
<td>0.951566510</td>
<td>0.00000000</td>
<td>modell</td>
</tr>
<tr>
<td>model1.Update</td>
<td>0.87356580</td>
<td>12.5%</td>
<td>0.8735658000</td>
<td>0.00000000</td>
<td>modell</td>
</tr>
<tr>
<td>model1/ScopeAnalyzer (ScopeAnalyzer.Start)</td>
<td>0.63986410</td>
<td>9.5%</td>
<td>0.6398641000</td>
<td>0.63986410</td>
<td>modell/ScopeAnalyzer</td>
</tr>
<tr>
<td>terminationPhase</td>
<td>0.60840390</td>
<td>9.4%</td>
<td>0.6084039000</td>
<td>0.51403390</td>
<td>modell</td>
</tr>
<tr>
<td>model1/ScopeAnalyzer (ScopeAnalyzer.Update)</td>
<td>0.56168360</td>
<td>8.5%</td>
<td>0.5616836000</td>
<td>0.56168360</td>
<td>modell/ScopeAnalyzer</td>
</tr>
<tr>
<td>model1/ScopeAnalyzer (ScopeAnalyzer.Start)</td>
<td>0.42120270</td>
<td>6.5%</td>
<td>0.4212027000</td>
<td>0.42120270</td>
<td>modell/ScopeAnalyzer</td>
</tr>
<tr>
<td>model1/ScopeAnalyzer (Scope.Outports_Motor)</td>
<td>0.42120270</td>
<td>6.5%</td>
<td>0.4212027000</td>
<td>0.42120270</td>
<td>modell/ScopeAnalyzer</td>
</tr>
<tr>
<td>model1/ScopeAnalyzer (Scope.time)</td>
<td>0.39000250</td>
<td>6.0%</td>
<td>0.3900025000</td>
<td>0.39000250</td>
<td>modell/ScopeAnalyzer</td>
</tr>
</tbody>
</table>
Example

(Profiling when prototyping in MATLAB)
Optimization at different phases in the development...
Now, let's take a closer look of the workflow in embedded design!
Example: Workflow for embedded design

- Constrained Design (E.g. Fixed-Point)
- Reference Design
- Target Specific Design
- Target Specific Design SoC/FPGA (E.g. Fixed-Point)
Example: Workflow for embedded design

1. **Reference Design**
2. **Constrained Design**
   - (E.g. Fixed-Point)
3. **Target Specific Design**
   - SoC/FPGA
     - (E.g. Fixed-Point)
4. **Target Specific Design**
   - C
     - (E.g. Fixed-Point)
Key Detection Algorithm

Accelerate the Design and Prototyping of Signal Processing Algorithms
15:00–15:30

In this session we show how you can use MATLAB® for designing and prototyping an algorithm that operates on streaming data. This way of working makes simulations fast and memory efficient. We also show how to test algorithms against data coming from low-cost hardware such as a smartphone or Raspberry Pi™. In the session Optimization and Implementation of Embedded Signal Processing Algorithms, we integrate the algorithm in a larger system that is targeted for a specific hardware platform.

You will see how to:

• Work with streaming data
• Accelerate simulations by keeping the memory footprint low
• Test algorithms against low-cost hardware
Component Integration

System Object
(Reference Design)

System Design

Optimize
Example: Workflow for embedded design

- Reference Design
- Constrained Design (E.g. Fixed-Point)
- Target Specific Design C (E.g. Fixed-Point)
- Target Specific Design SoC/FPGA (E.g. Fixed-Point)
Example: Workflow for embedded design

1. **Constrained Design**
   - (E.g. Fixed-Point)

2. **Reference Design**

3. **Target Specific Design**
   - SoC/FPGA
   - (E.g. Fixed-Point)

4. **Target Specific Design**
   - C
   - (E.g. Fixed-Point)
Example: Workflow for embedded design
Example: Workflow for embedded design

- Reference Design
- Constrained Design (E.g. Fixed-Point)
- Target Specific Design (E.g. SoC/FPGA)
- Target Specific Design (E.g. Fixed-Point)
Example: Workflow for embedded design

Reference Design

Constrained Design
(E.g. Fixed-Point)

Target Specific Design
SoC/FPGA
(E.g. Fixed-Point)

Simulink

C
(E.g. Fixed-Point)

Target Specific Design
DEMO
(System Integration and Generating Code for ARM Cortex A9 from Simulink)
(Including verification with PIL)
Component Integration

System Object
(Reference Design)

System Design

Optimize

ARM Cortex-A9

ARM

FPGA

RF

Data Type Conversion

Ready

95%
A few words about Embedded Coder...
Embedded Coder® Quick Start helps you generate production code for a Simulink model.

The Quick Start tool:
- Asks a few questions about your code generation goals and your target hardware.
- Validates your model against your selections.
- Shows you the recommended configuration changes.
- Applies the configuration changes and generates code.

No changes are made to your configuration until you choose to generate code. After successful code generation, the Quick Start tool presents possible next steps.
Another use case...
Example: Workflow for embedded design

1. Reference Design
2. Constrained Design (E.g. Fixed-Point)
3. Target Specific Design (E.g. SoC/FPGA, Fixed-Point)
4. Target Specific Design (C)

MATLAB
Example

(Generating Code ARM Cortex A9 from MATLAB)
Component Integration

System Design

System Object
(Reference Design)

Optimize
Example: Workflow for embedded design

Reference Design

Constrained Design
(E.g. Fixed-Point)

Target Specific Design
SoC/FPGA
(E.g. Fixed-Point)

Target Specific Design
C
(E.g. Fixed-Point)
Example: Workflow for embedded design

- Reference Design
- ARM Cortex-A8
- Constrained Design (E.g. Fixed-Point)
- Target Specific Design (C, E.g. Fixed-Point)
- Speed / Area Optimizations
- Target Specific Design SoC/FPGA (E.g. Fixed-Point)
Area Optimization for FPGA implementation
Example

(Area Optimization for FPGA Implementation)
Component Integration

System Design

System Object
(Reference Design)

Optimize

ARM

Cortex-A9

FPGA
Summary
Optimization and Implementation of Embedded Signal Processing Algorithms

- Reference Design
- Constrained Design (E.g. Fixed-Point)
- Target Specific Design C (E.g. Fixed-Point)
- Target Specific Design SoC/FPGA (E.g. Fixed-Point)
Questions?