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
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...

Prototyping

C

SoC/FPGA
Optimization at different phases in the development...
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
Example

(Profiling when prototyping in MATLAB)
Optimization at different phases in the development...
Now, lets take a closer look of the workflow in embedded design!
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)
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)
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 Design

System Object
(Reference Design)

Optimize

ARM
Cortex-A9

FPGA

ARMSimulink

RF

Data Type Conversion

Control Parameters

FPGA

ARM

Spectrum Analyzer

Scope
Example: Workflow for embedded design
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**

Constrained Design
(E.g. Fixed-Point)

Reference Design

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

Target Specific Design
C
(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)

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

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

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

System Design

System Object
(Reference Design)

Optimize

ARM
Cortex-A9

FPGA

ARM

RF

Optimize

System Design
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.
DEMO

(Embedded Coder Quick Start)
Another use case...
Example: Workflow for embedded design

Reference Design

Constrained Design
(E.g. Fixed-Point)

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

MATLAB

Target Specific Design
C
(E.g. Fixed-Point)
Example

(Generating Code ARM Cortex A9 from MATLAB)
Component Integration

System Object
(Reference Design)

System Design

Optimize

ARM
Cortex-A9

FPGA
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

Constrained Design (E.g. Fixed-Point)

Reference Design

Target Specific Design
(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

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

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

- **Reference Design**

- **Target Specific Design**
  - C
  - (E.g. Fixed-Point)
Questions?