Model Driven Approaches to Firmware Development in Selex ES

21 Jan. 2015
Presentation Overview

• Selex ES Company Overview
  – Firmware Engineering at Selex ES Edinburgh

• History of Model Driven Engineering within Selex ES Edinburgh

• Cross-Functional MDE Strategy

• Firmware-Specific MDE Workflow for DSP Design

• Future MDE Workflow for Joint Firmware/Software Design

• Do’s and Don’ts for Establishing a Model Driven Workflow
• Mission Critical Systems and Defensive Aids Systems

Integrated Networking Solutions for Netcentric Capabilities

Sensors & Systems for Homeland Protection, Homeland Defence, ATC/ATM, VTMS
The Firmware Engineering discipline are responsible for providing FPGA expertise to a wide range of projects and products within the Radar and Advanced Targeting business based in Edinburgh.

Designs include:
  • Radar antenna control
  • Control of aircraft self-protection systems
  • Implementation of radar and image processing algorithms
  • Control of electro-optic turrets
A Simplified ‘V’ Diagram

Modelling & Algorithm Development
- Analysis & Requirements Capture
- Simulation Analysis & Design
- Sub-system Model Design

Test & Qual
- System Execution and Evaluation
- System Integration and Test
- Sub-system Model Integration and Test

Modelling Repository

Sub-system Model Implementation & Unit Test Framework

Firmware, Software, And Hardware Implementation

© 2014 Selex ES Ltd – All rights reserved
Model Driven Engineering at Selex ES Edinburgh

Motor control  Comms link  UAV Imaging Suite  Fast Jet Imaging Suite  High perf. mirror servo  Fire Control radar  Surveillance radar  Adv. IR Counter-Measure

2003  2004  2005  2006  2007  2008  2009  2010  2011  2012  2013/4
Defining an MDE workflow:

- A workflow must be designed, just like any other part of the system.
- Workflows will usually contain elements of simulation and of implementation.
- Workflows which use models as their basis have been consistently proven within our organisation to be more efficient, cheaper to implement and less prone to error.
- Workflows should be flexible enough to respond to both innovation and unexpected events.
- There is no one ‘right’ MDE workflow, there is however a mindset that will ensure success!

“If you’re failing to plan, you’re planning to fail”
Product Design – the bad old days

- Static requirements
- “Big Bang” (high risk!) integration
- Cost of fixing errors increases to right
- Incompatible with previously presented ‘V’ diagram
A new, more integrated strategy

- Teams are constantly interacting using the model as a reference
- Interactions are inherently bi-directional
- Errors and mistakes much more likely to be discovered early in the lifecycle
Golden Reference Model

Simulink Floating-Point Model

Simulink Fixed-Point Model

Implementation Model

FPGA-In-the-Loop Testing

• Initial, high-level capture of the system performance using Matlab scripts and functions.
• Functionality of subsequent stages is compared against this reference by means of automated testbenches
• Use of correct code structuring at this stage is essential.

• Functionality of the Golden Reference Model transferred to Simulink environment using idealised all-double-precision representation.

• Convert the Simulink DSP functionality to a fixed-point implementation

• Implement the DSP elements using the target architecture (available BRAMs, etc.)
• Generate HDL or netlist for use with down-stream tools

• DSP functionality verified on target FPGA using custom Ethernet protocol (more later)
Workflow Background

- Primarily used to target DSP designs to Xilinx FPGAs by firmware engineering discipline
- Has recently been extended to including targeting capability to the new generation Xilinx Zynq architecture with both Firmware and Software elements (more later)

- Tools used:
  - Matlab / Simulink
  - Xilinx System Generator
    - (more recently, Simulink HDL coder)
  - Xilinx ISE

- Examples of applications:
  - Airborne imaging for UAV and Fast Jet applications
  - Aircraft self-protection suite
  - Fast Jet and Surveillance radar
Application Example – Stabilised Mirror Controller

Captured data presented to model

Matlab used for frequency domain analysis and Test vector abstraction

• All design / analysis and (component) testing done in Matlab / Simulink
Combined Software/Firmware MDE Workflow

1. Golden Reference Model (Matlab scripts/functions)
   - Initial, high-level capture of the system performance using Matlab scripts and functions.
   - Functionality of subsequent stages is compared against this reference by means of automated testbenches.
   - Use of correct code structuring at this stage is essential.

2. Simulink Model (floating-point Embedded Matlab)

3. (Optional) Autocode ANSI C (PIL)
   - Optionally auto-code and test a Processor-In-the-Loop implementation of Simulink model.
   - Allows an upper bound on execution times (as little to no optimisation will have been carried out on the generated code) and could inform the software/firmware partitioning decision.

4. Software Design (C/C++)
   - DSP elements will be auto-coded from Simulink HDL Coder or a vendor-specific tool such as Xilinx System Generator.
   - Autocode (tailoring currently required) for software elements

5. Firmware Design (Fixed-Point Simulink Model)
   - Firmware and software designs are tested in isolation of each other.
   - Traceability against requirements demonstrated via PIL and FIL using the Golden Reference model
   - This stage currently subject of ongoing work within SELEX

6. Firmware verification & validation (FIL)
   - Conduct a full-system hardware-in-the-loop test involving both processors and FPGAs.
   - Compare to Golden Reference Model to ensure that the required functionality has been achieved
The Zynq 7000 is a complete Embedded Processing solution offering software and programmable logic processing in a single device

- Embedded ARM Cortex A9 dual core Processor
- Firmware Programmable Logic
- High performance processing with low power consumption footprint
Application Example – Zynq-based Embedded Tracker

- Requirements Captured in ‘Golden Reference’ Matlab Simulink Model
- Implementation Code Auto generated and verified against ‘Golden Reference’
- Code implemented on Target Hardware and Verified against ‘Golden Reference’

Complex Algorithms Implemented and Verified Directly on Hardware from System Model
MDE Workflow Developments

• Original Firmware MDE process is well established and is the “go to” process for developing new DSP functionality
  • Toolset allows for very rapid design iterations
  • Verification against a “Golden Reference” significantly improved traceability.

• New generation of combined processor/FPGA devices (i.e. Zynq) motivating changes to the workflow
  • Verification now covers FPGA-In-the-Loop and Processor-In-the-Loop, as well as a full System-In-the-Loop
  • Advances in the toolset allows for increased use of autocoding for software DSP as well as firmware
An ideal MDE workflow is:

- Not the preserve of a select few engineers
- Not focused only on modelling, or only on implementation
- Never more complex than it needs to be to achieve the workflow objective
- Capable of providing Intellectual Property protection where appropriate
- Easy to use: it’s surprising how many people are looking for an excuse to switch off

Most importantly:

- A workflow is no substitute for experience and (current) domain knowledge – MDE is NOT push button in all but the most trivial cases, it is a way of thinking
Do’s and Don’ts for MDE, Part 1 – Top 3 Do’s

Do:

- Tailor an approach to MDE which applies to your particular project
- Keep the models as simple as possible – complexity for its own sake causes mistakes
- Implement continuous quality management features in your models
Don’t:

• Model for the sake of it – some things are, literally, not worth the effort
• Assume that a MDE workflow will simply give you answers, or executable code. Domain knowledge is still required – anyone who says otherwise is naïve at best
• Believe tool vendors claims without verifying everything via pilot projects, offline evaluations or independent corroboration.