Rapid Firmware Prototyping with Matlab/Simulink for MicroTCA.4

Introducing the Rapid-X toolset for fast prototyping for the MTCA.4 platform

Paweł Prędki

3rd MTCA Workshop for Industry and Research
10.12.2014 DESY, Hamburg

Outline

- Motivation
- Overview
- Example
- Current applications
- Conclusions

Motivation

> MTCA.4 – („MORE BANDWIDTH!!!" – Vollrath Dirksen)

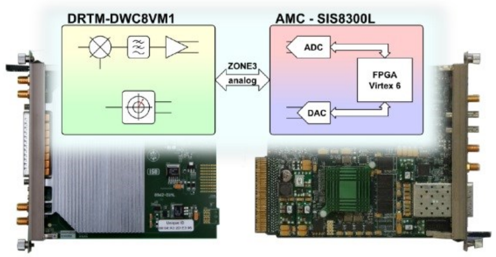
> Allows faster and more robust application algorithms

but...

> Requires new firmware components
  - Board Support Packages
  - Interface support (e.g. PCIe)
  - Modules for application algorithms

> Requires HDL knowledge for prototyping

> Problem: **Significant workload for FPGA Developers**



Idea

> Matlab/Simulink used by application engineers for modelling and simulation

> Generate the application code directly from this model

> **Xilinx SysGen allows netlist generation**

Main Advantages:

> Shift workload from FPGA developers to application engineers

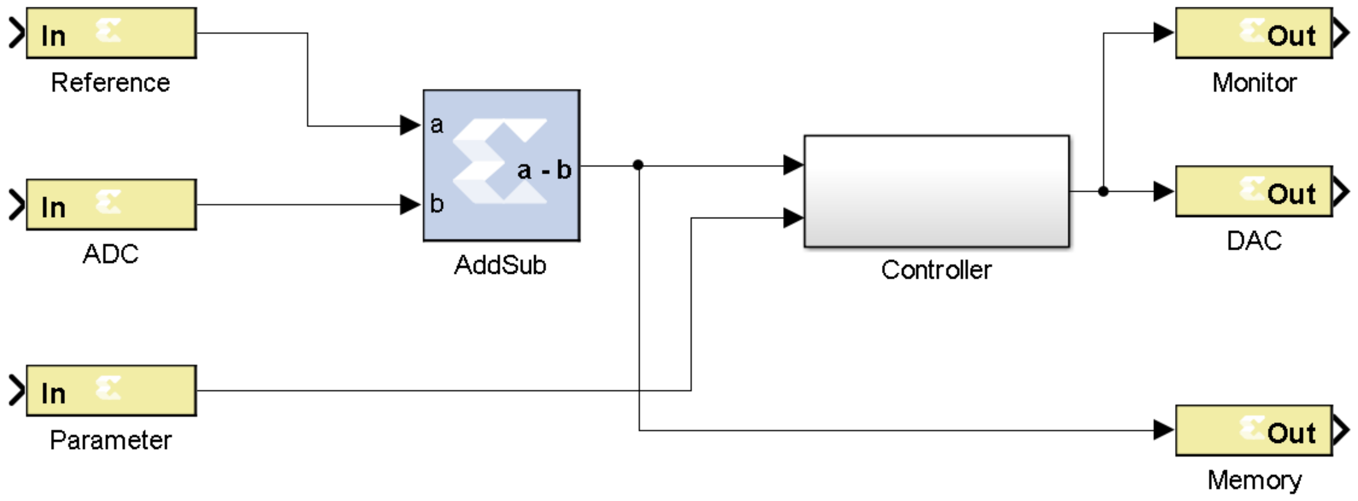
> Very fast from idea to prototype

> Simulation will show real behavior (finite word length, delays, etc…)

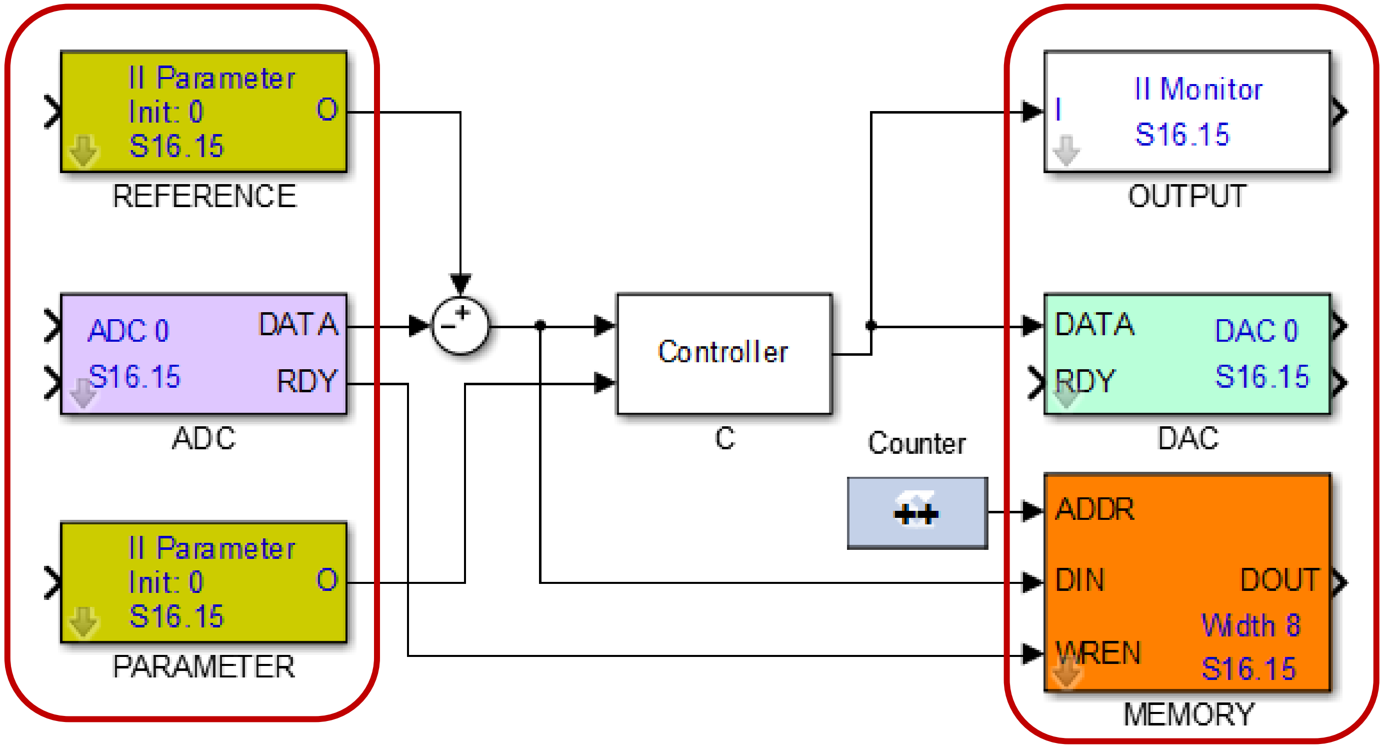
Idea – cont.

> Disadvantages

- No interfaces to BSP
- No distinction of signal types (ADC input, parameter, etc.)
- No possibility to easily re-use previously designed HDL modules

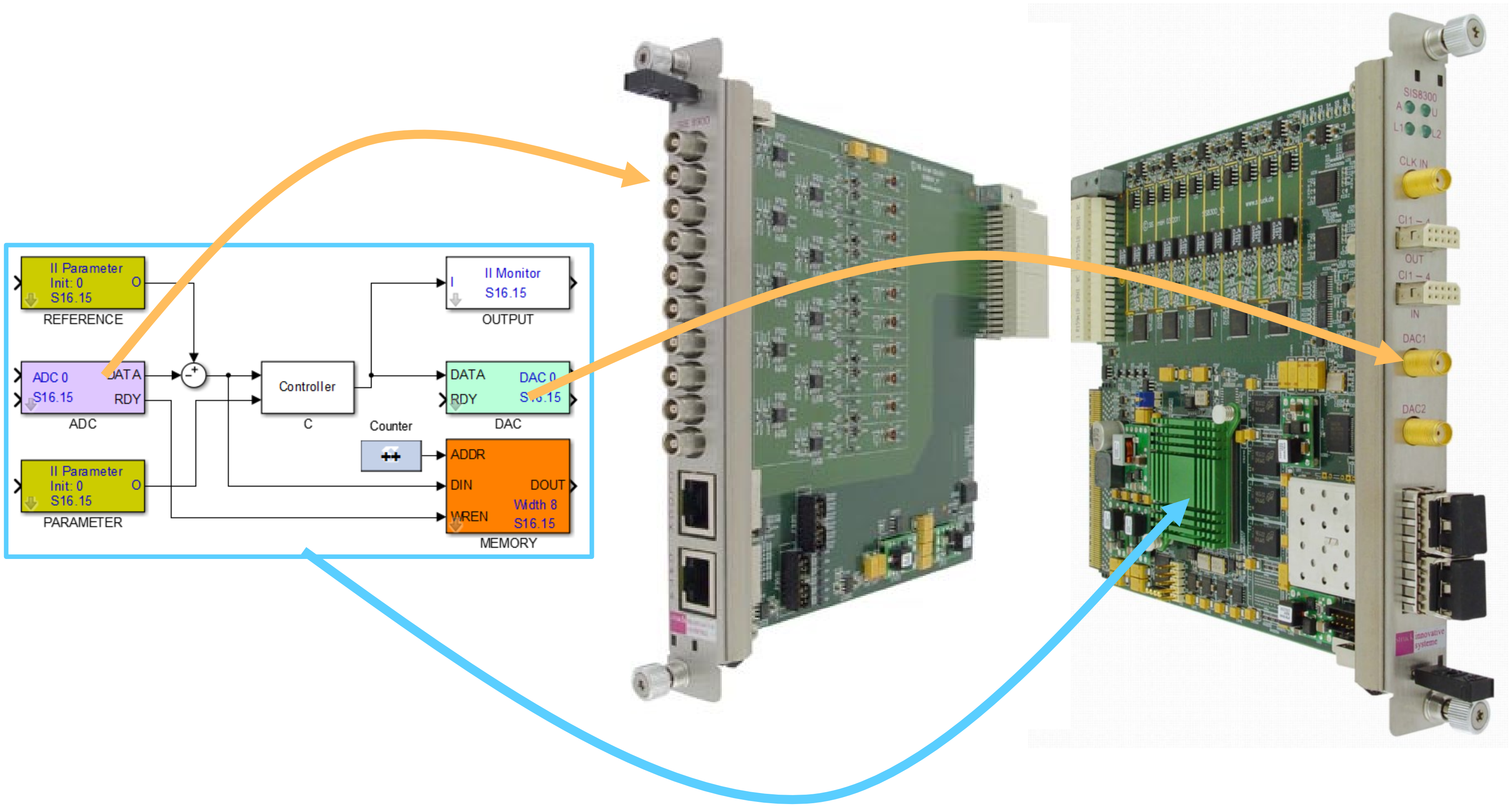


Solution – Rapid-X toolset with custom library components

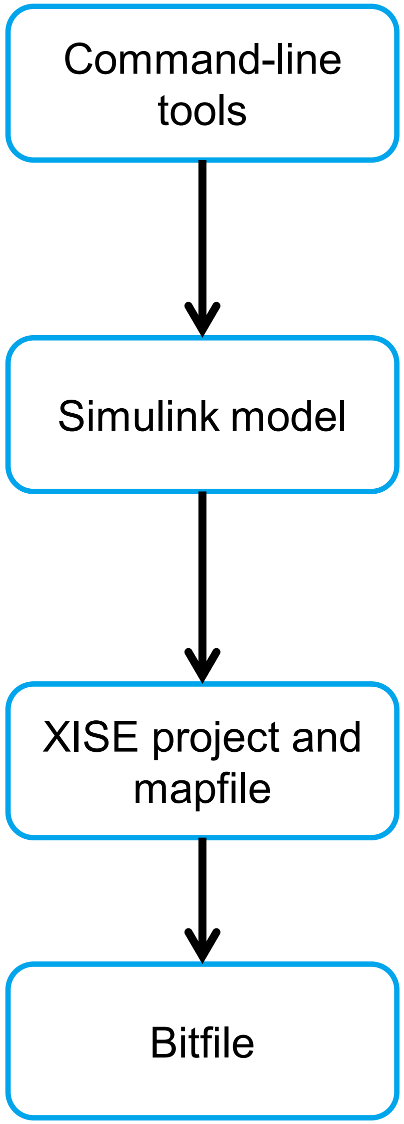


Example Application: SISO-Controller

Block Representation of Hardware Components



Project flow

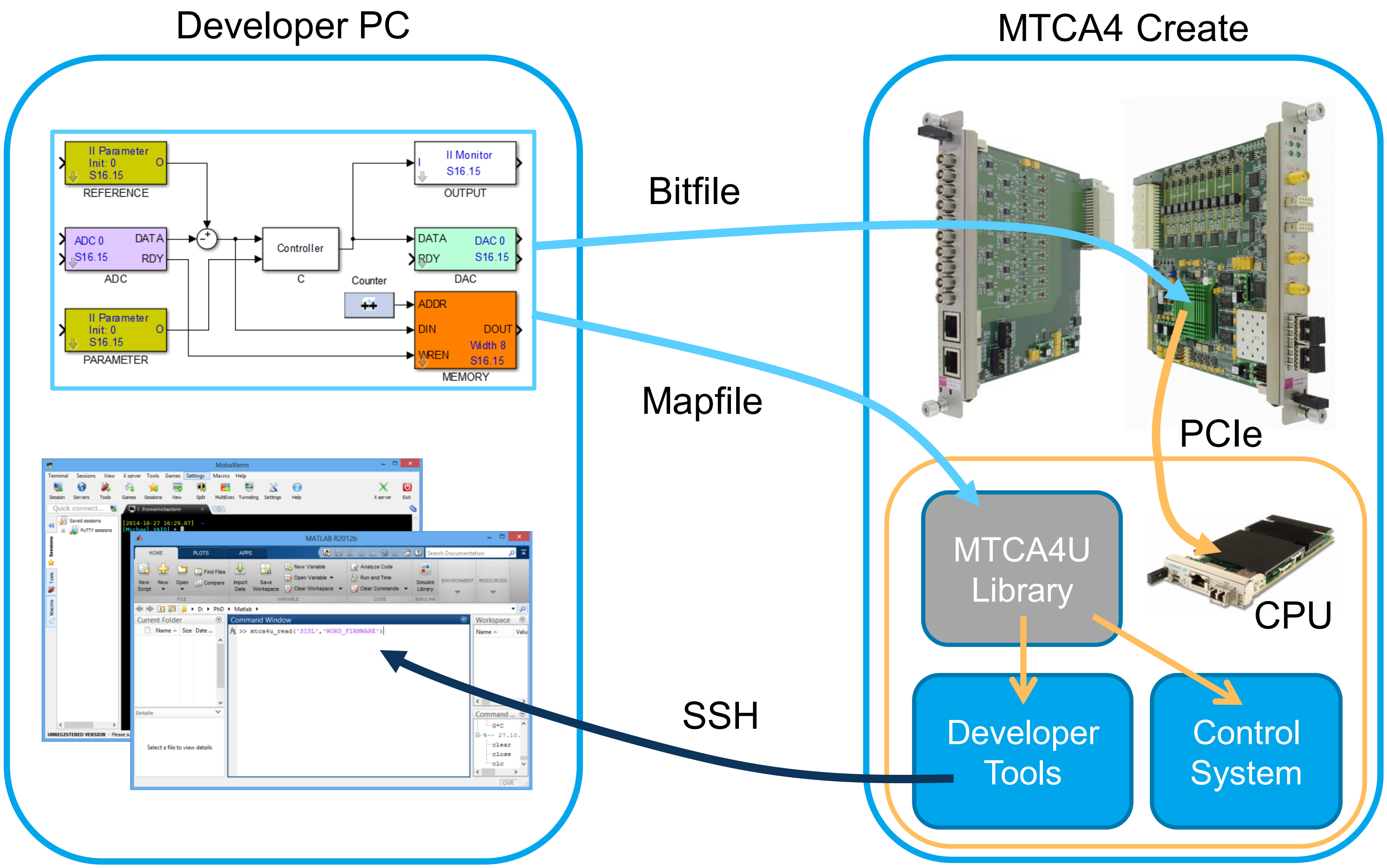


1. Create empty application project from template
2. Include BSP

1. Generate netlists based on the model (Xilinx SysGen)
2. Populate interfaces between BSP and application (Rapid-X)
3. Generate mapfile (Rapid-X)

1. Synthesis, translation, mapping, PAR, bitfile generation (Xilinx ISE)

Application Engineers Point of View



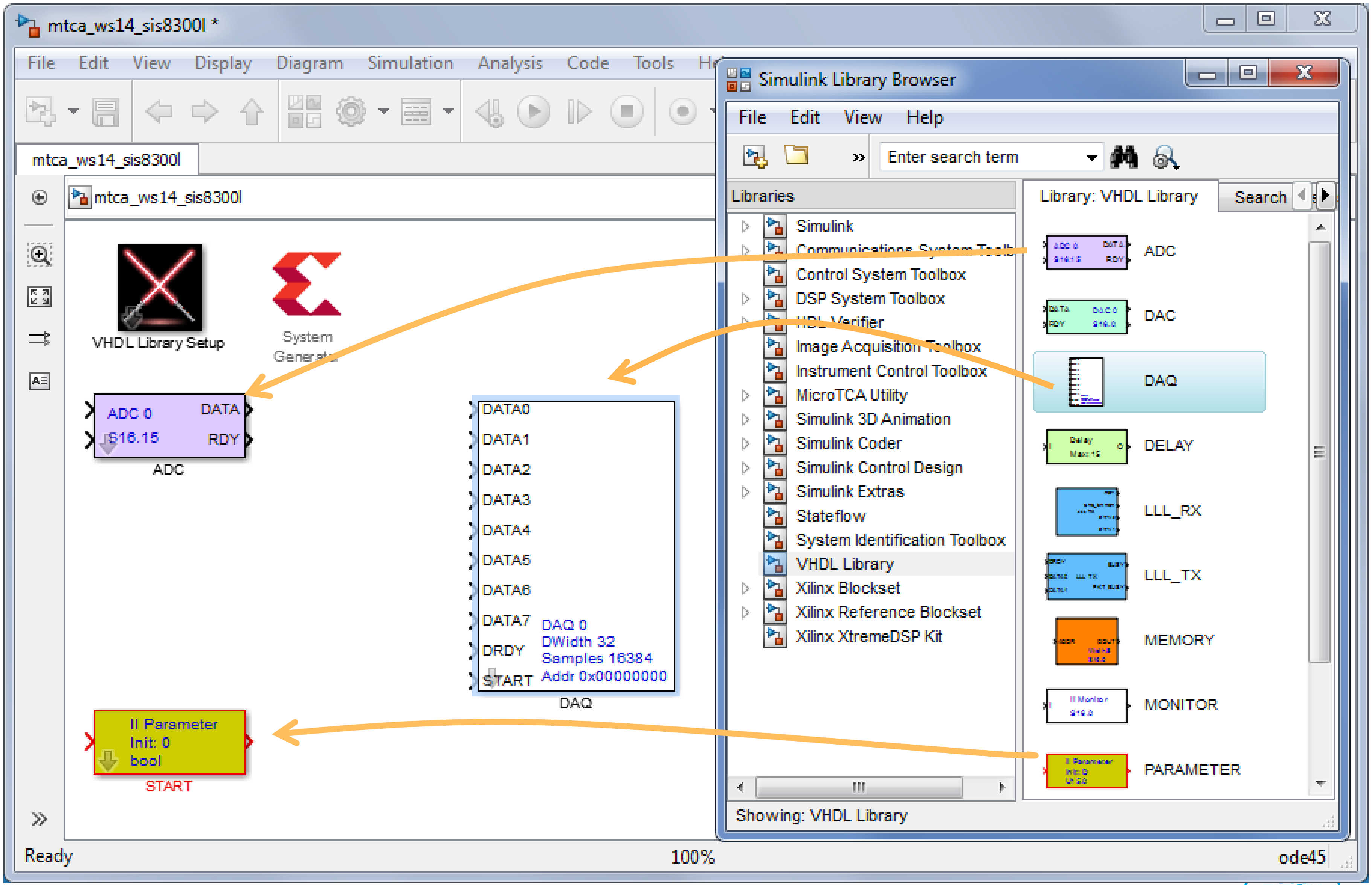
```
>> mtca4u_read('SISL','WORD_FIRMWARE')
```

Example: Data Aquision and Analysis

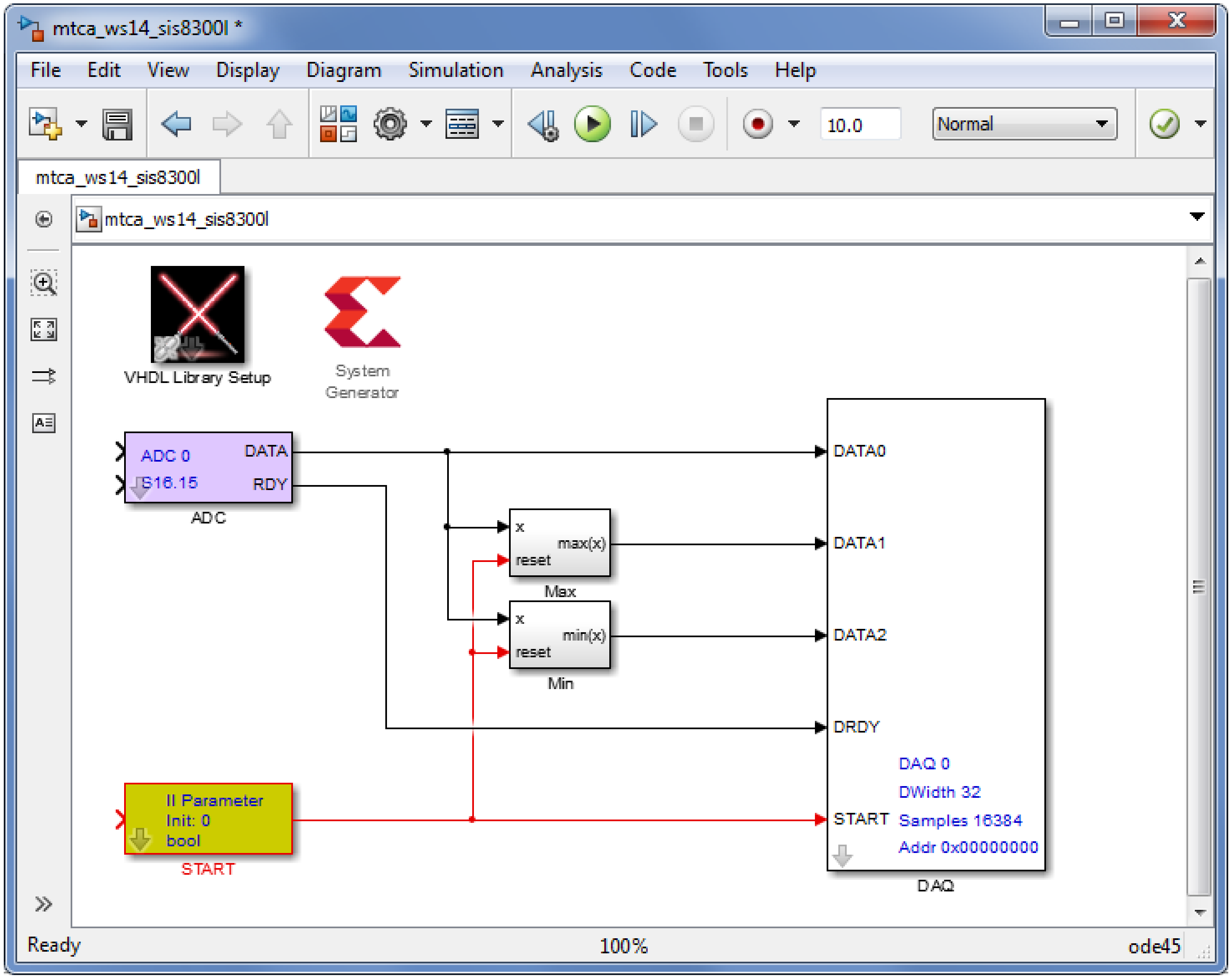
> Requirements:

- Sample raw ADC data with 81MHz to DDR-RAM
- Store the highest value in one channel of the RAM
- Store the smallest value in another channel of the RAM
- Start acquisition on command from the CPU

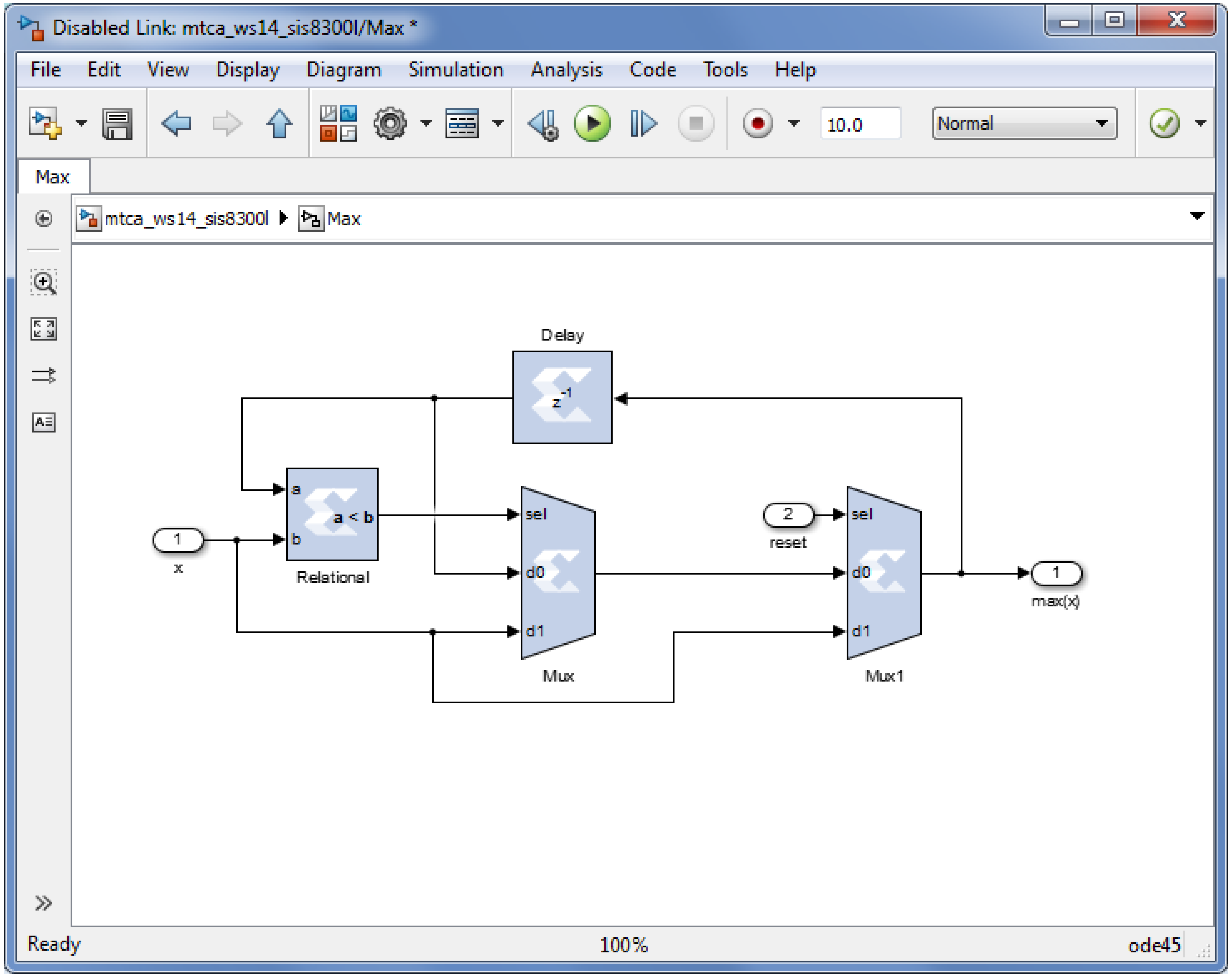
Example: Data Aquision and Analysis



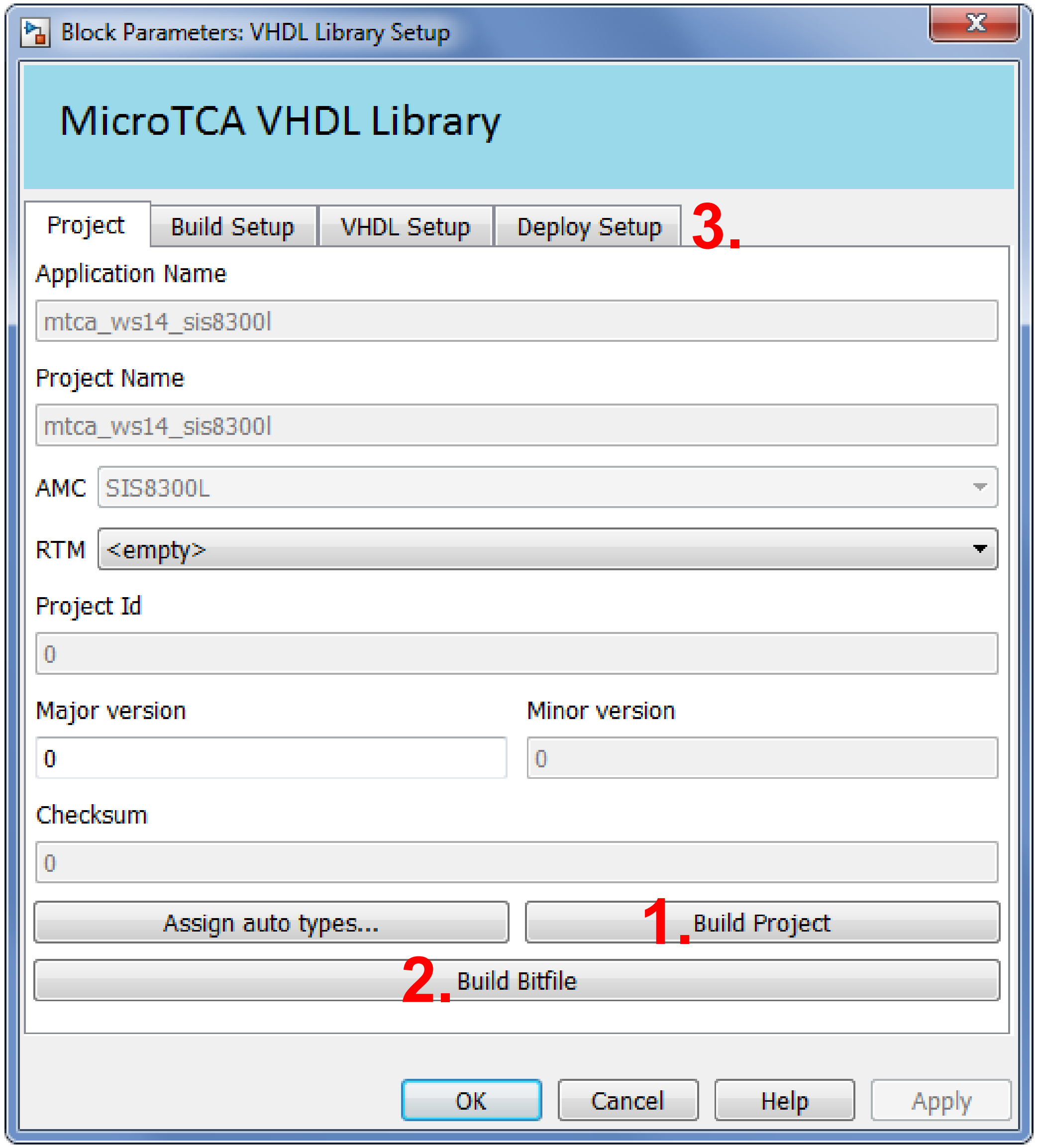
Example: Data Aquision and Analysis



Example: Data Aquision and Analysis



Example: Data Aquision and Analysis



Block Parameters: VHDL Library Setup

MicroTCA VHDL Library

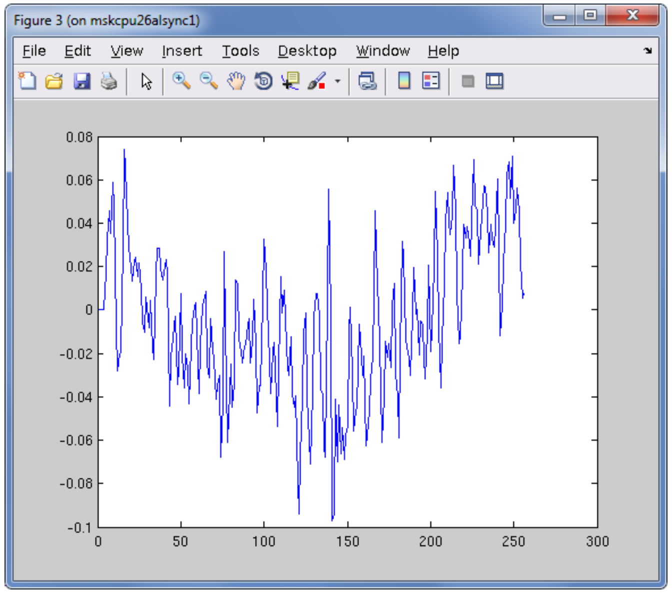
Application Name	mtca_ws14_sis8300l
Project Name	mtca_ws14_sis8300l
AMC	SIS8300L
RTM	<empty>
Project Id	0
Major version	0
Minor version	0
Checksum	0

1. Build the XISE Project
2. Generate the bitfile
3. Upload the bitfile to the FPGA and the mapfile to the crate CPU (automatic upload not supported currently)

Example: Data Aquision and Analysis

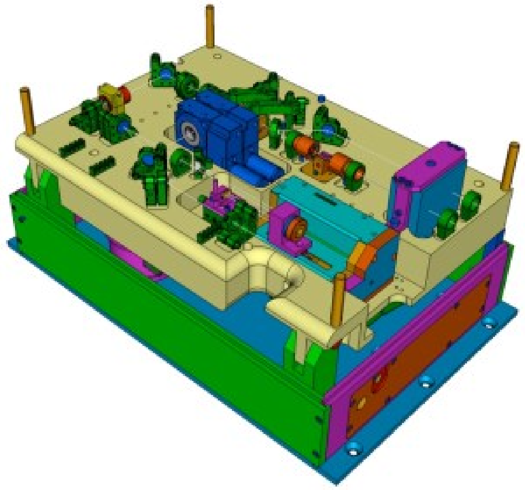
```
%% Data Aqusition Test
%

m = mtca4u(); % initalize MicroTCA4 Matlab Tools

SISL8300_init(m, 'SISL'); % initalize SISL board
m.write('SISL', 'WORD_RESET_N', 1); % activate board
```

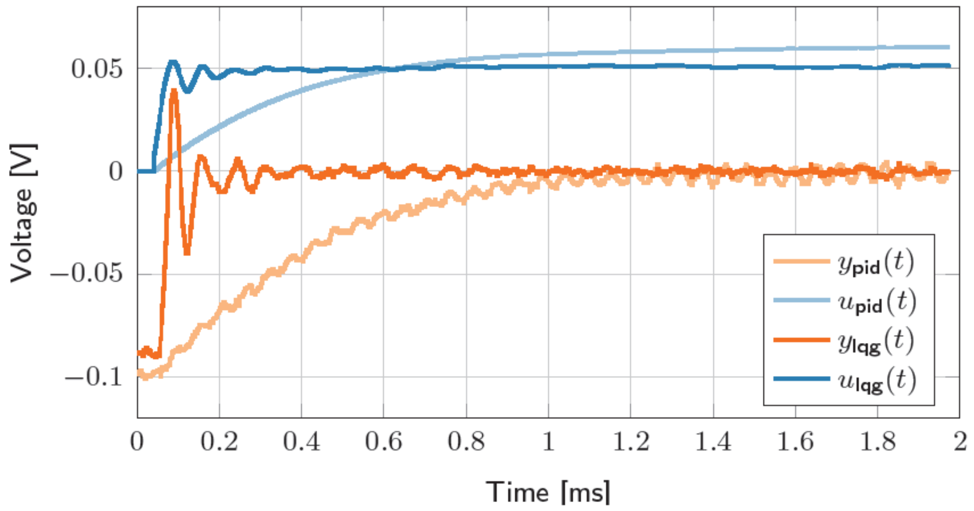
```
%%

m.write('SISL','WORD_START', 1, 'WORD_START', '0'); % Trigger the sampling
pause(1);

plot(m.read_dma('SISL',1,256)); % Plot the raw data
```



Current Application @ DESY

- Link Stabilizing Unit in Laboratory Setup





- Controller for the Master Laser Oscillator in development
- Beam charge stabilization with Pockels Cell

Conclusion

- Rapid prototyping **with virtually no** knowledge of VHDL
- **Workload shift** : VHDL developer → application engineer
- **Real behavior modelled** using tools known to application engineers
- Easily extendable with **new modules** optimized by VHDL developers
- **Separation** between hardware (board) and application
- Rapid-X models **migratable** to different boards
- **No external tools** - everything is coded in Matlab and tcl scripts

The End

Thank you for your attention

Comments?

Questions?