Lars Risbo, Purifi, Denmark

Rapid Prototyping of unknown solutions to only partially known problems using Simulink and the SoC design flow

Overview

- Challenges
  - » Need platform for fast idea/concept exploration, fast failure of unknown algorithms in a system with hi-speed, lo-latency feedback through the real world

- Solution
  - » Matlab SoC design flow, Zedboard, custom board, auto-generation of Simulink, real-time tuning, Built-in Self Test (BiST)

- Results
  - » Proof of concept, closed feedback loop, MLS analyzer as test case, ARM-FPGA comm is a bottleneck

Lars Risbo, Purifi

- Lars Risbo

  » Ph.D 1994: High-order sigma-delta modulation

  » Founder, Toccata 1996, world's first all digital power amplifier in 1998.

  » Toccata acquired by Texas Instruments in 2000

  » TI-Fellow (2012), TI audio CTO (2013-2014)

- Purifi

  » Founded in 2015 by Bruno Putzeys, Peter Lyngdorf & Lars Risbo

  » Stealth mode, but focused on signal processing, system-level modeling/optimization/control

# Matlab + Simulink: great RPT tool

- Instant success/failure! Size/cost not important

- Captures understanding of the problem, e.g. through models and data mining.

- Quick mock-up of solution ideas, Simulink mixes cont. /discrete time blocks

- Great for visualization, team sharing

- Test-beds and analysis integrated

- Powerful automation of tasks, scripts, objects etc.

- High level of abstraction: scripts, vectorization, objects, operator overloading, hierarchical models etc.
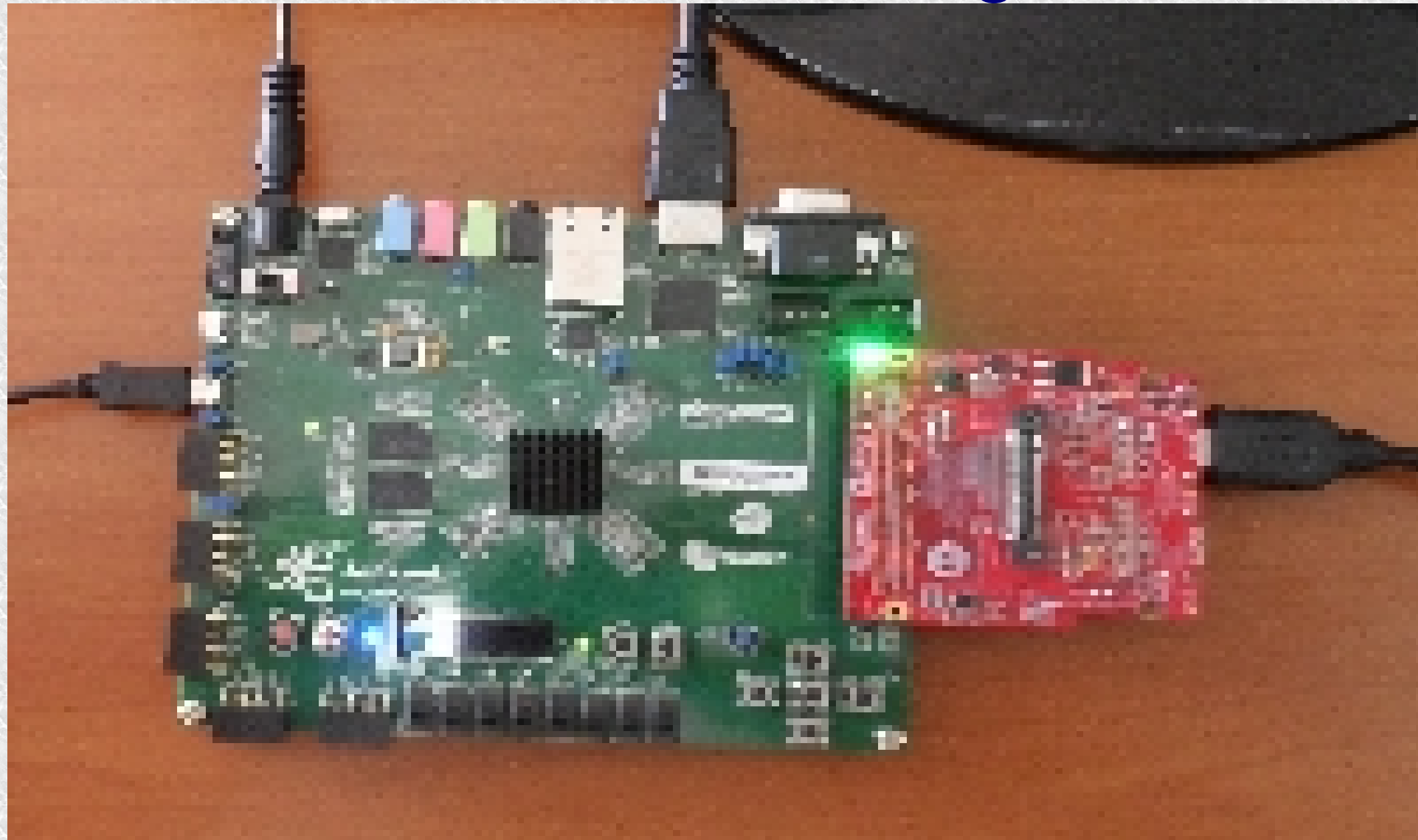
H/W support/acceleration when

- The real-world is involved in low latency feedback loops at high processing speed

- Custom high-speed interfaces

    » ADCs /DACs, mixers etc.

- Acceleration using FPGA+Embedded processing

- Solution:

    » Matlab SoC flow

    » Automated code generation from Simulink

    » HDL coder for fast H/W

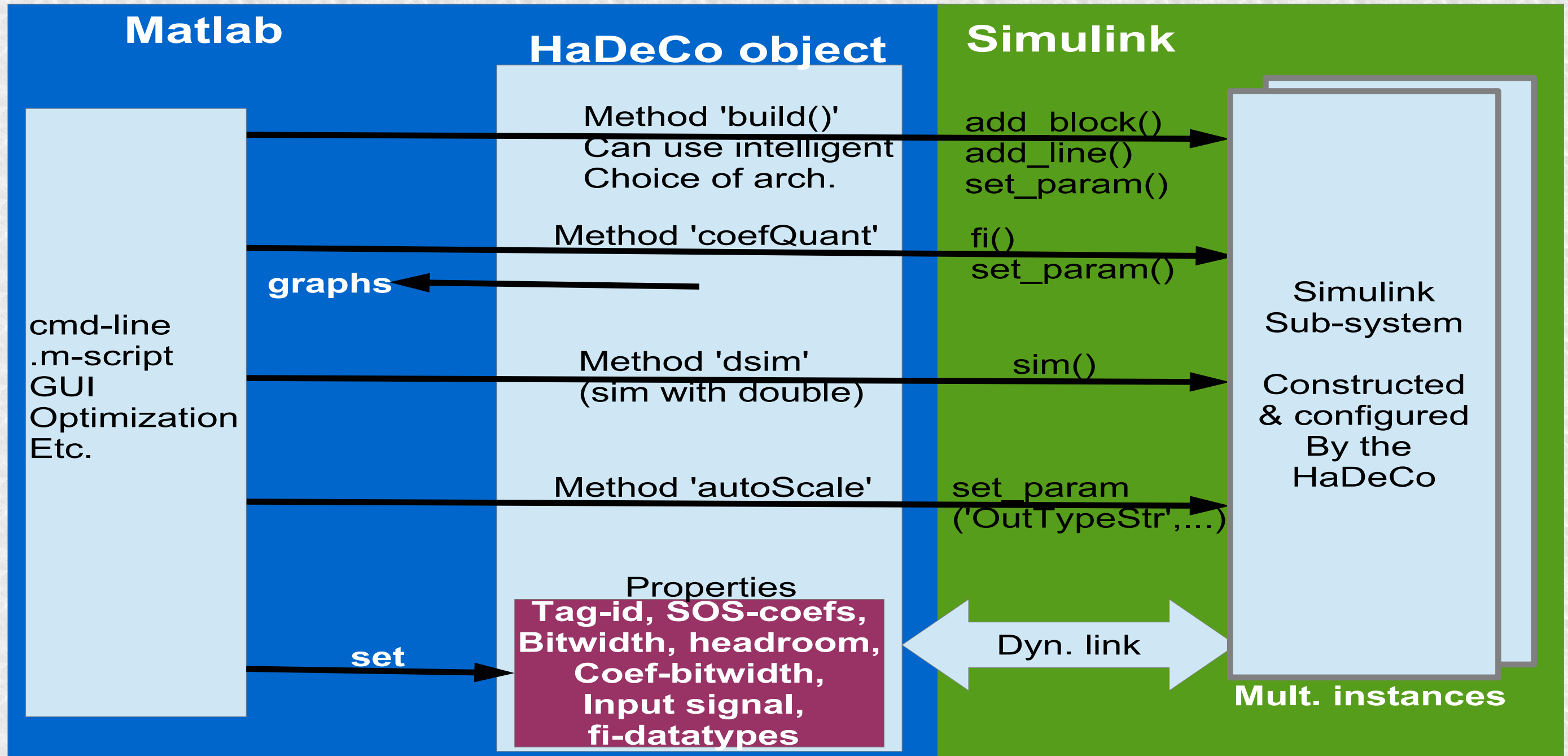# Purifi RPT Platform
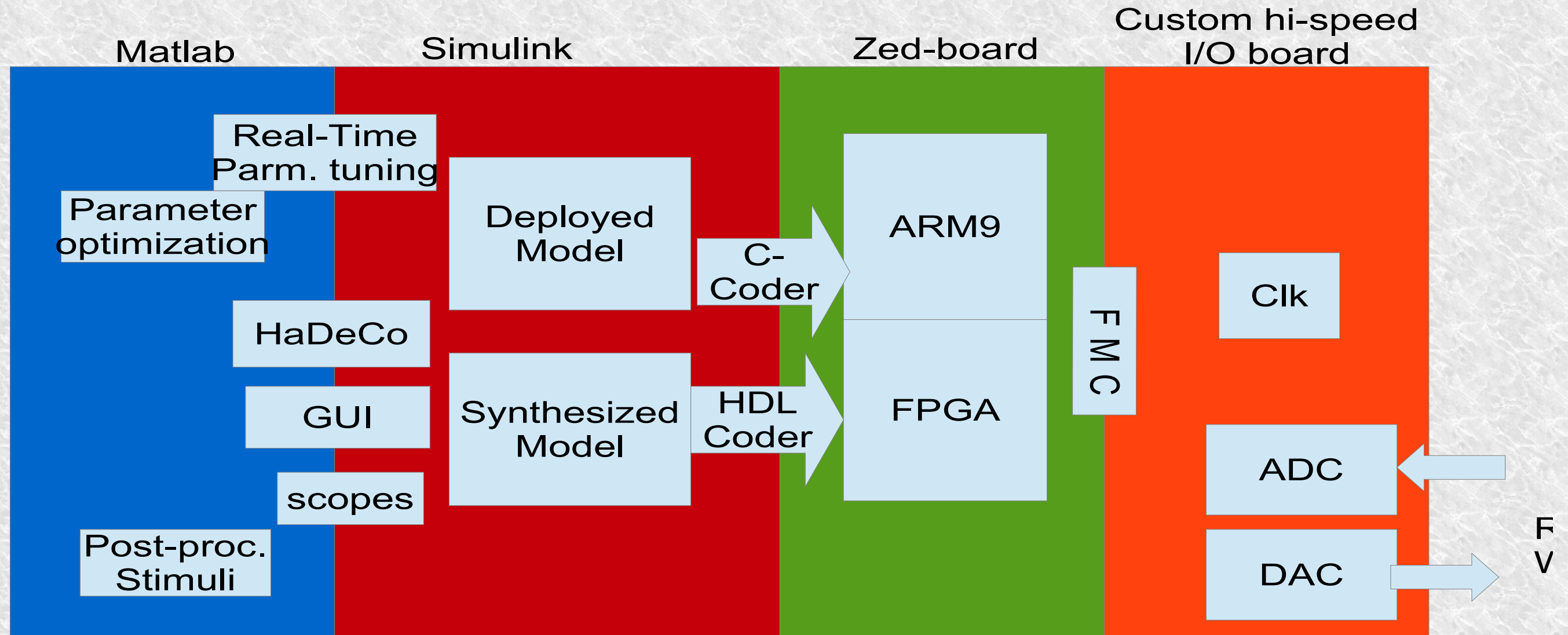
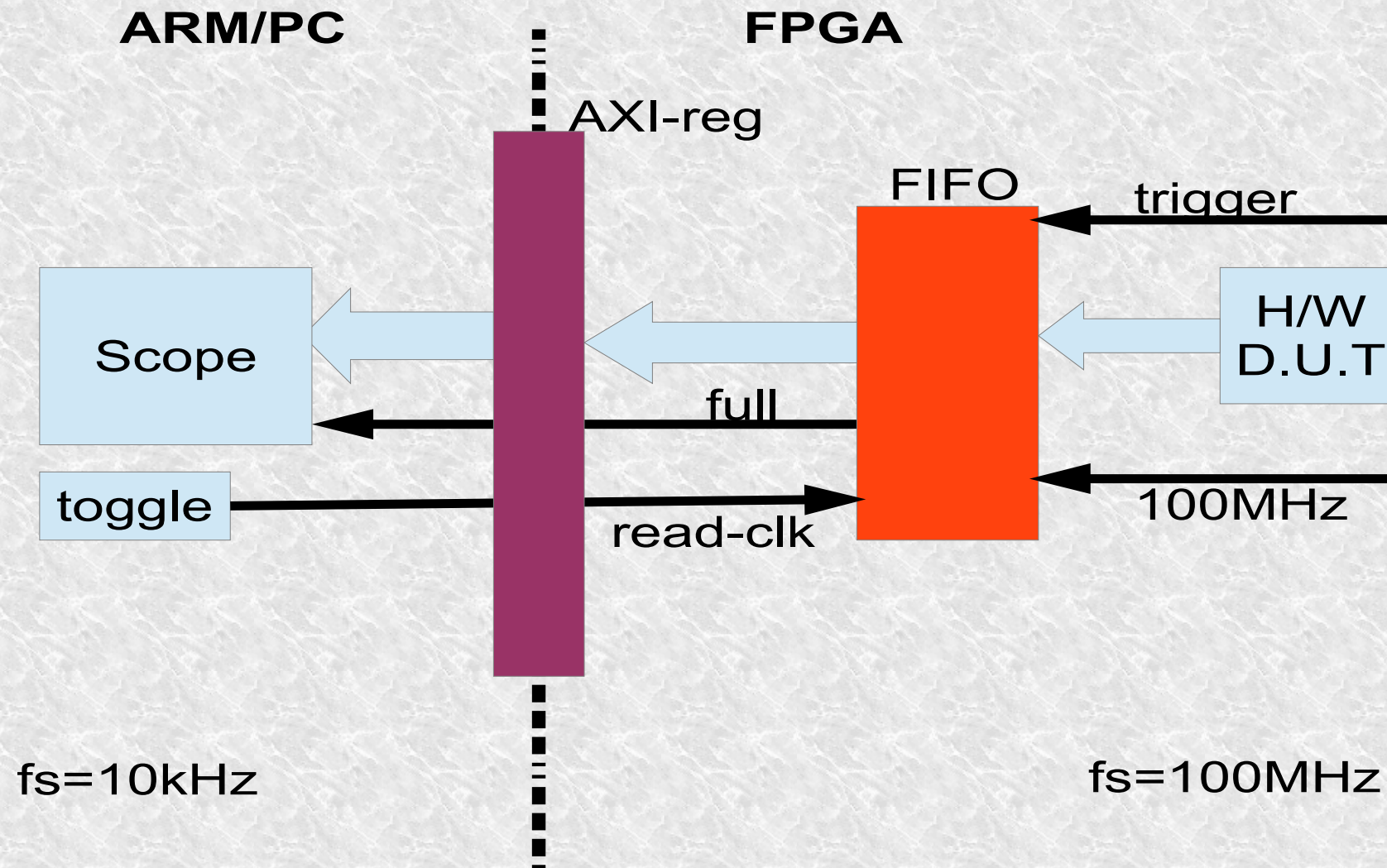# Zedboard with an FMC daughter card

Even higher levels of abstraction

- Example: linear filter block in pure Simulink:

    ! Just define the Transfer Function and go and tweak endlessly...

    ! Change the TF from workspace directly

- Using HDL Coder:

    ! Needs to be fixed-point, pick proper architecture and scaling, quite a barrier to creativity and speed

- Should be just as easy as in pure Simulink!

- More automation/abstraction:

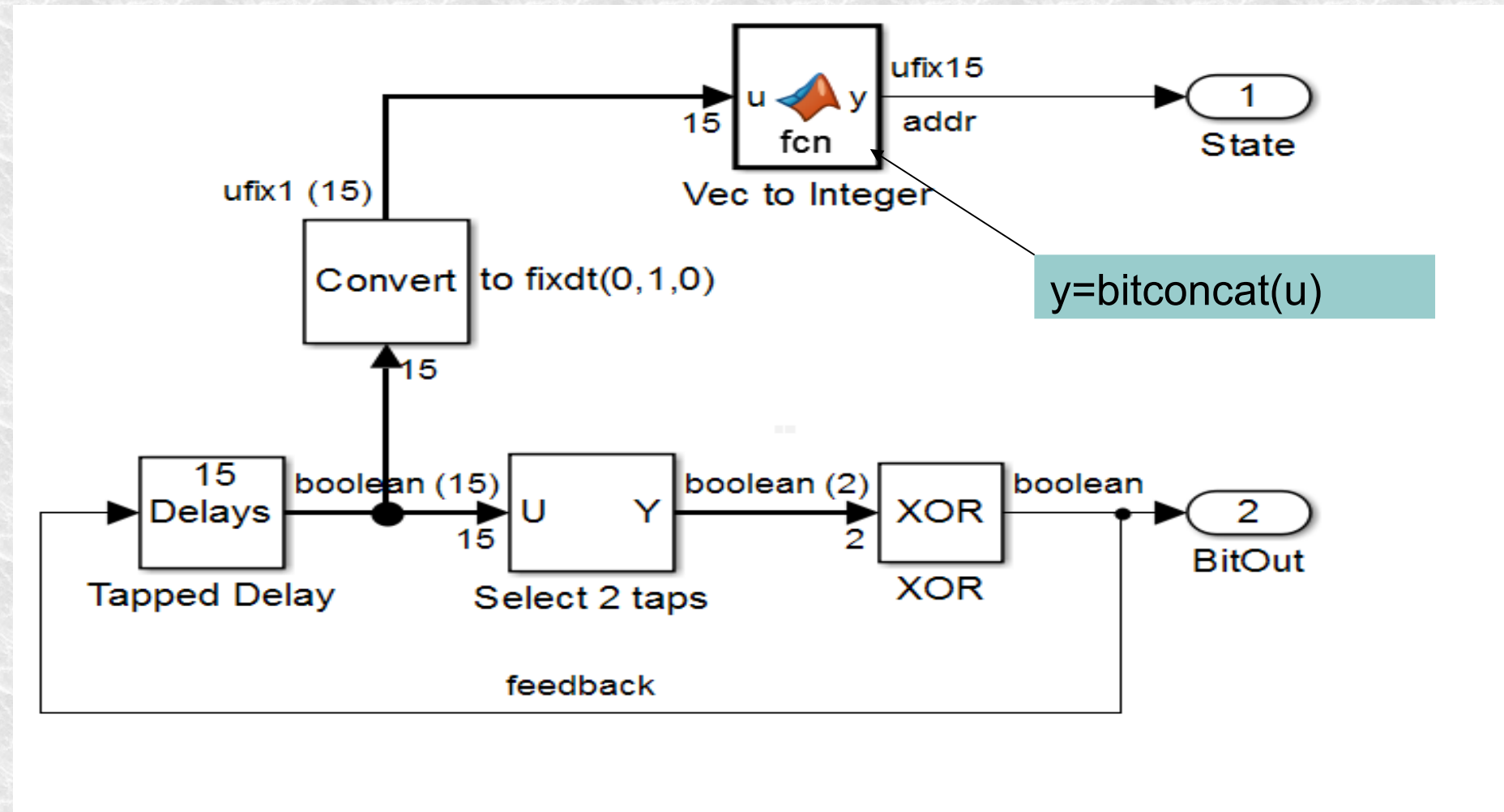   ! Introducing the Hardware-Design & Control Object HaDeCo

HaDeCo

Matlab

HaDeCo object

Simulink

cmd-line
.m-script
GUI
Optimization
Etc.

Method 'build()'
Can use intelligent
Choice of arch.

add_block()
add_line()
set_param()

Method 'coefQuant'

fi()
set_param()

graphs

Method 'dsim'
(sim with double)

sim()

Method 'autoScale'

set_param
('OutTypeStr',...)

Simulink
Sub-system

Constructed
& configured
By the
HaDeCo

Properties
**Tag-id, SOS-coefs,
Bitwidth, headroom,
Coef-bitwidth,
Input signal,
fi-datatypes**

set

Dyn. link

Mult. instances

Maximum Length Sequence Generator

1-st pass success in HDL Coder, 100% reconfigurable using workspace/scripts
Used in an MLS network analyzer for system identification of the ADCs and H/W

Path to ASIC/IC

- After prototyping → fast to real product
- HDL Code migrates to ASIC/IC
- Re-use of test-benches from prototyping
- Verification using HDL Verifier, co-simulations
- Automatic test vector generation
- Let the machine do the hard & repeated work

Conclusion

- The SoC flow expands Matlab/simulink into RPTing of fast real-time systems

- Fast idea exploration and fail/learn-cycles

- More abstraction/automation desirable for repeated tasks

  » Auto-generated/configured Simulink

- Real-time tuning, data analysis, self-test

- Fast-forward to real product (e.g. ASIC)

  » Re-use of test-beds, co-sim, vectors etc