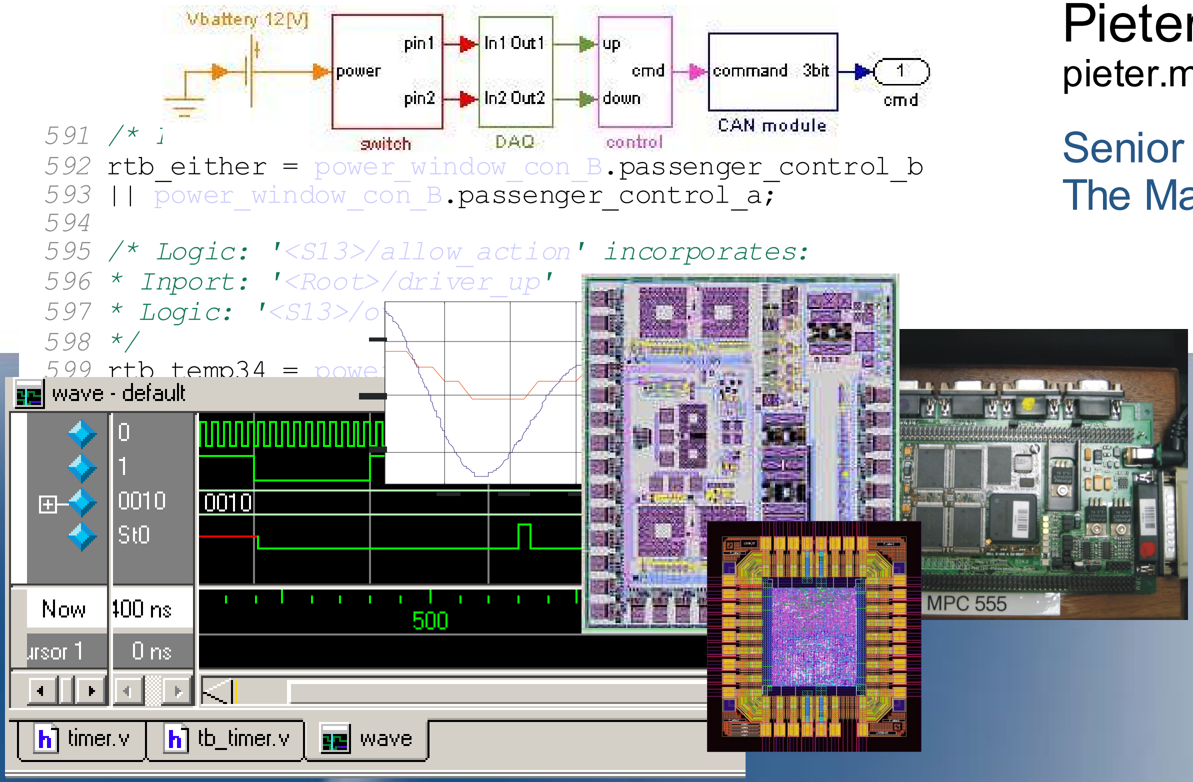
An Introduction to Model-Based Design of Embedded Systems

Pieter J. Mosterman
pieter.mosterman@mathworks.com

Senior Research Scientist
The MathWorks, Inc.



```
/* l
rtb_either = power_window_con_B.passenger_control_b
|| power_window_con_B.passenger_control_a;

/* Logic: '<S13>/allow_action' incorporates:
* Inport: '<Root>/driver_up'
* Logic: '<S13>/o
*/
rtb_temp34 = powe
```

Introduction

- Model-Based Design
  - Exploit computational models
  - Increasingly adopted in industry
- Modeling at an enterprise level
  - Many different modeling formalisms
  - Relate and combine models
    - Different parts of a system
    - Different design stages of a system
- Challenges
  - Efficiently manage models, formalisms and levels of abstraction
  - Efficiently evaluate dynamics of different computational semantics

Agenda

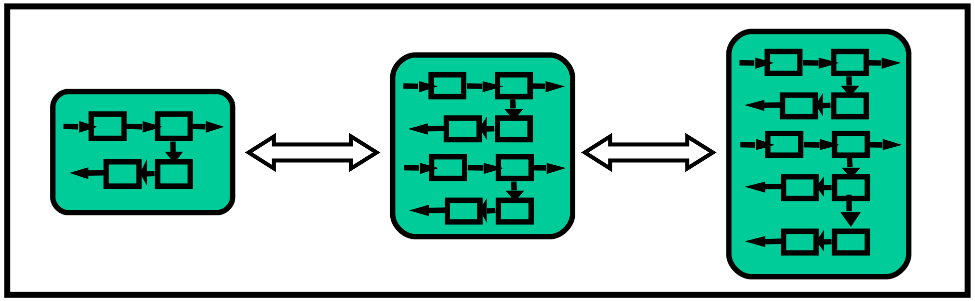
- Model-Based Design
- Model elaboration demonstration
- Implementing a functional specification
- Summary

Model-Based Design

Requirements and Specifications

Excel
Word
Scenario diagrams

Functionality

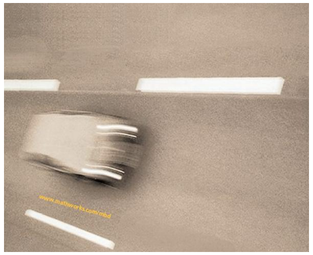


Block diagrams
Statecharts
Hybrid Automata
Bond graphs

Implementation

C/C++
Java
HDL
ASM

Test and Verification



DEVS
Automata
Numerical Values
Polynomials

Mercedes-Benz trucks cruise controller

- Created and implemented modular cruise control software on a target ECU
- Results
  - Compact, efficient code
  - High test efficiency
  - Fast development

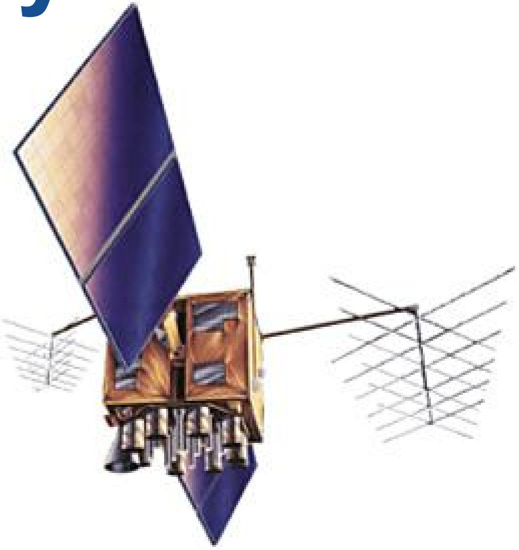
Mercedes-Benz truck.

“MathWorks tools for modeling and code generation enabled us to quickly and seamlessly perform design and test iterations, and release our product within a hard deadline of 18 months."

Mario Wunsche,
DaimlerChrysler

Satellite attitude control system

- Developed the attitude control system for Ministat-01
- Results
  - Accelerated simulation
  - Substantial time and cost savings
  - Problem-free performance, helping to extend the satellite mission's life



"We faced the challenge of not only developing the software for the attitude control system for Ministat-01 in less than one year, but also completing exhaustive tests before the integration of the software with the other satellite systems, all within 14 months. It would not have been possible to develop, produce, and test the software within that timeframe without MathWorks tools."

Jose Ramon Villa,
SENER Ingenieria y Sistemas, S.A.

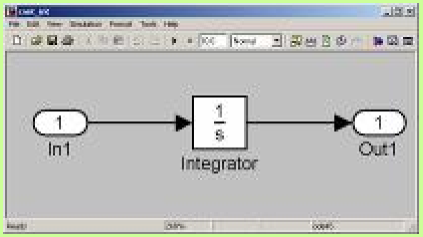
Enterprise wide Model-Based Design

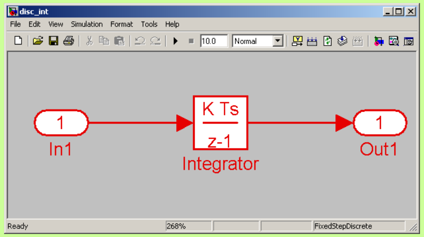
- Multiple teams of experts working on
  - Different parts of the system
    - Engine, transmission, suspension
    - Fuel injection, shift logic, ABS
  - One part in different phases of development
    - Requirements, systems, controls, and software engineers
    - Requiring different levels of abstraction
- Challenges arise due to the partition of this effort
  - How to minimize dependencies, when the parts are inherently coupled?
  - How to integrate the parts and optimize the complete system?

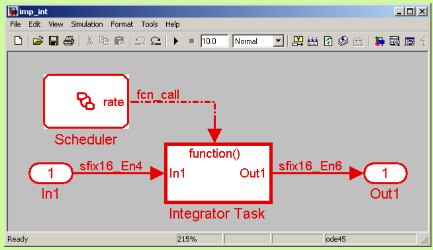
Integration is two-fold

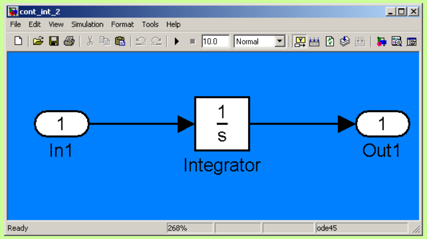
Same part needs to migrate through different phases

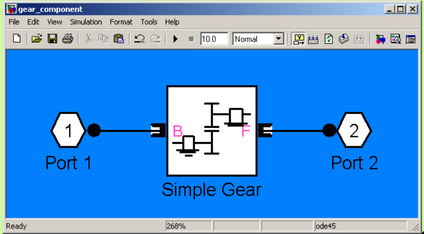
System parts

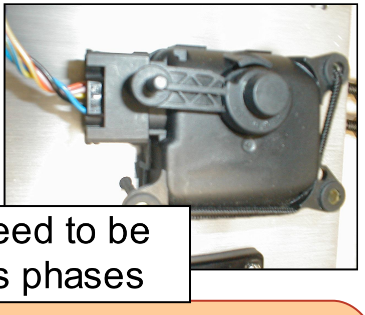




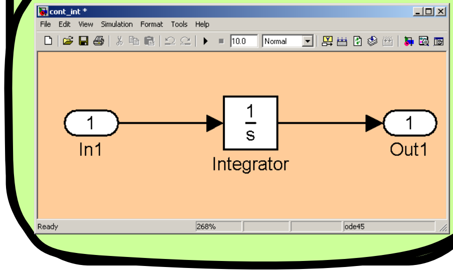








Different parts need to be integrated across phases



```
...
/* Model update function */
void disc_int_rp_update(int_T tid)
{
  /* Update for DiscreteIntegrator:
'<Root>/Integrator' */
  disc_int_rp_DWork.Integrator_DSTATE +=
   disc_int_rp_P.Integrator_gainval *
   disc_int_rp_U.In1;
...
```

```
void disc_int_pcg_step(void)
{

  b = acc;
  acc += 5243 * a >> 17;

}
```

Different parts need to be integrated within the same phase

Development phases

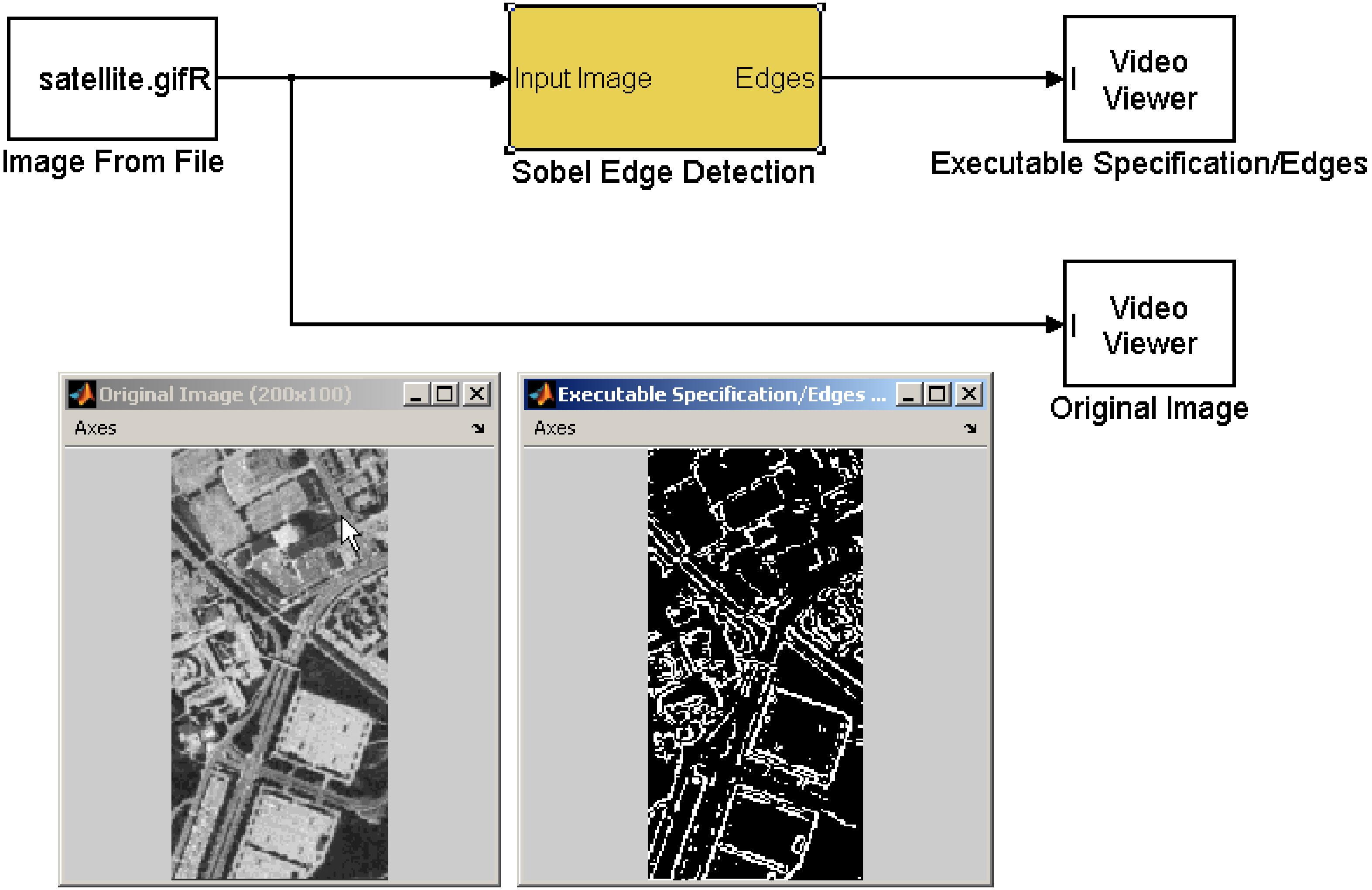
Ecosystem of computational models

- Make models electronically available
- Directly upload from within Simulink
- Proper modeling practice
  - Test harness with referenced core model
- Meta tags
  - Number of input and output ports
  - Continuous time integration
  - Cyclomatic complexity

Agenda

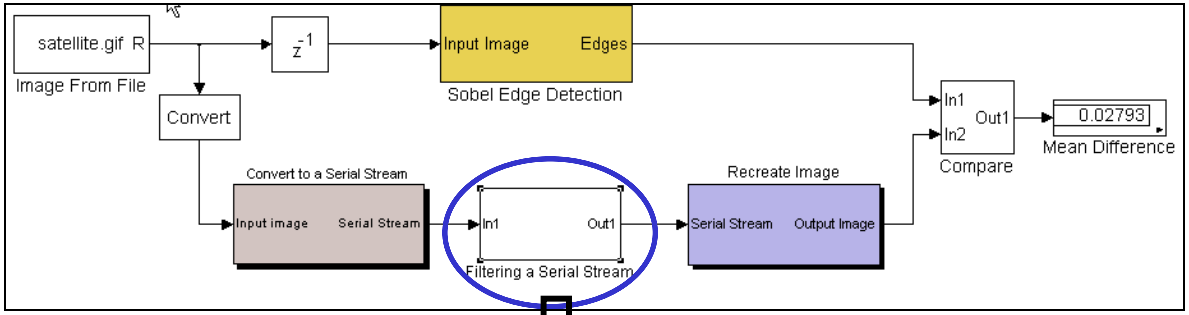
- Model-Based Design
- Model elaboration
- Implementing a functional specification
- Summary

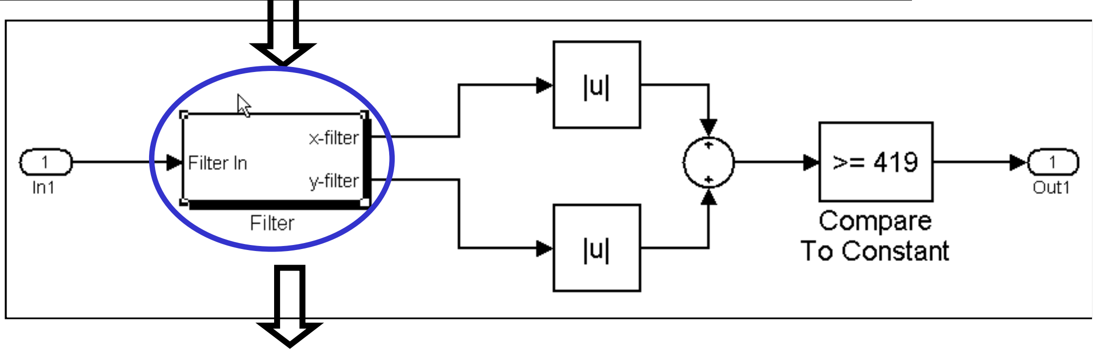
Executable specification

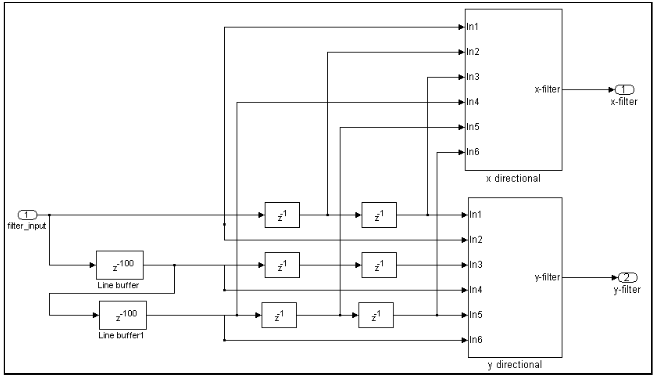


Design elaboration

Live
MATLAB
Demo







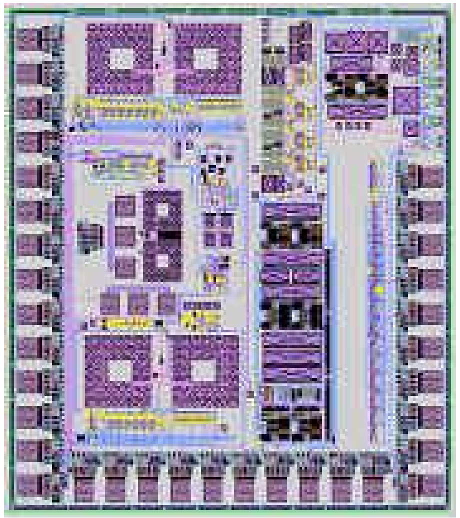
- Edge detection
  - Design space exploration using floating point
  - Conversion to fixed point
  - Elaboration to facilitate streaming data
  - Co-simulate with HDL

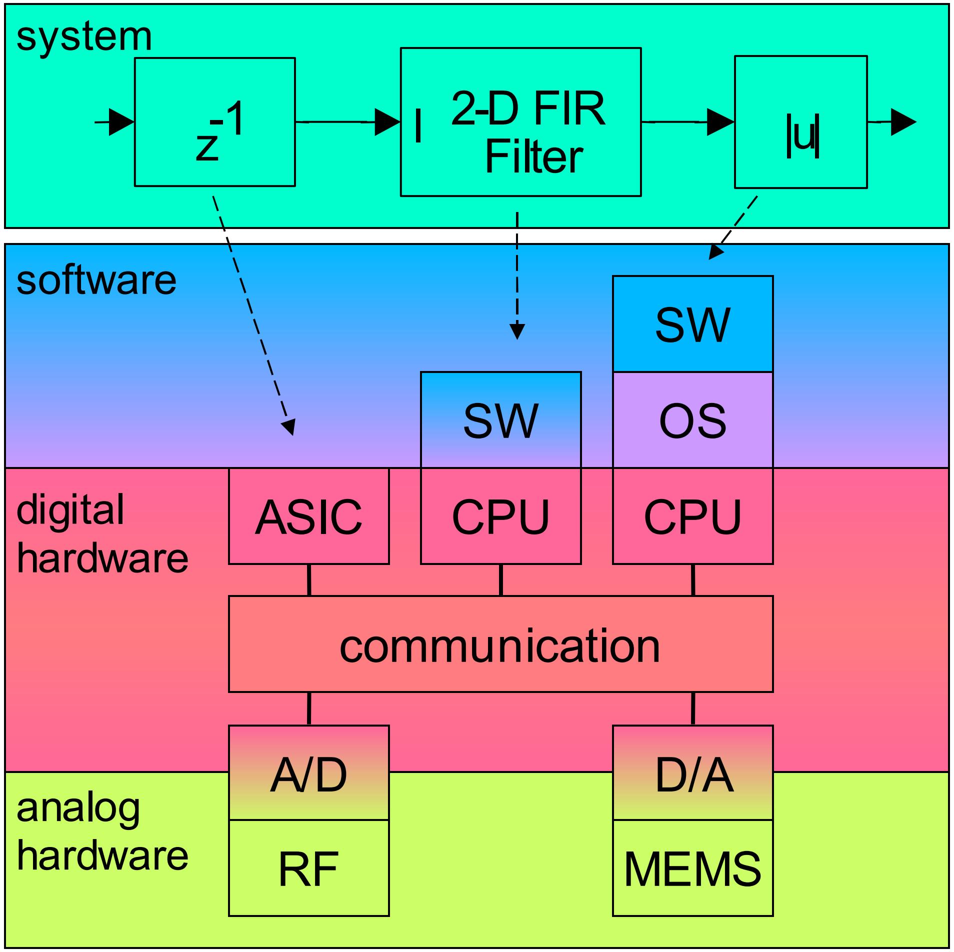
Agenda

- Model-Based Design
- Model elaboration demonstration
- Implementing a functional specification
- Summary

Towards an implementation

- SoC platform
  - Heterogeneous
  - Highlights more common design challenges

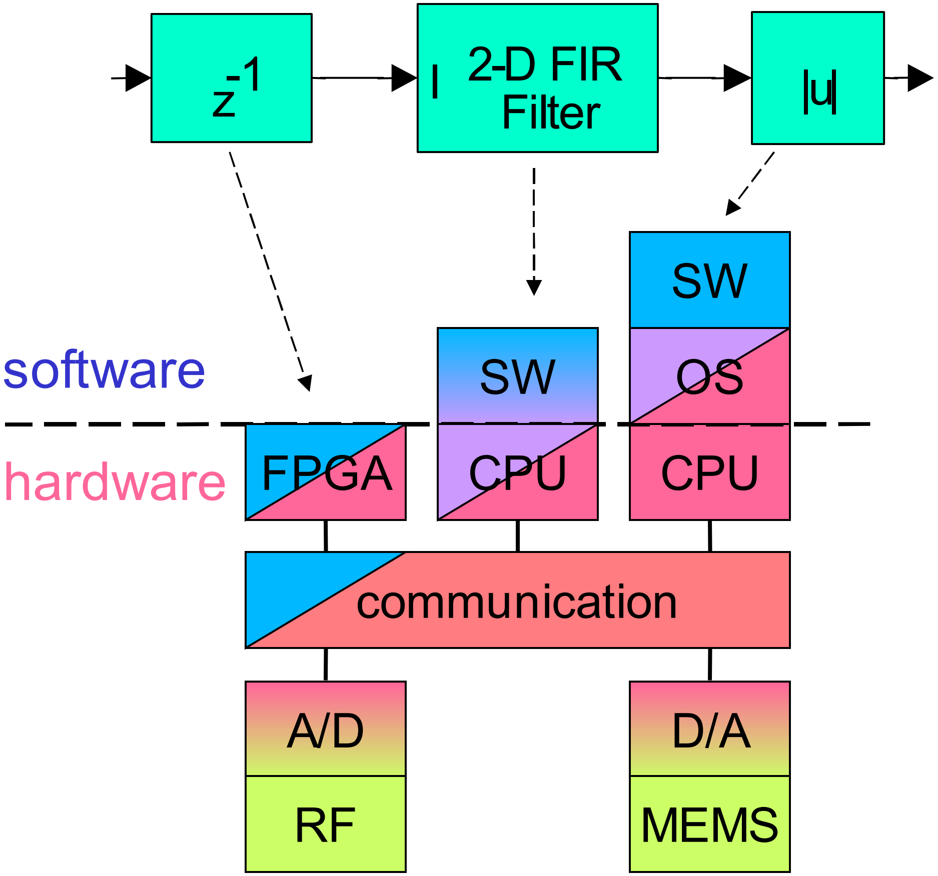




cf. Jan Madsen

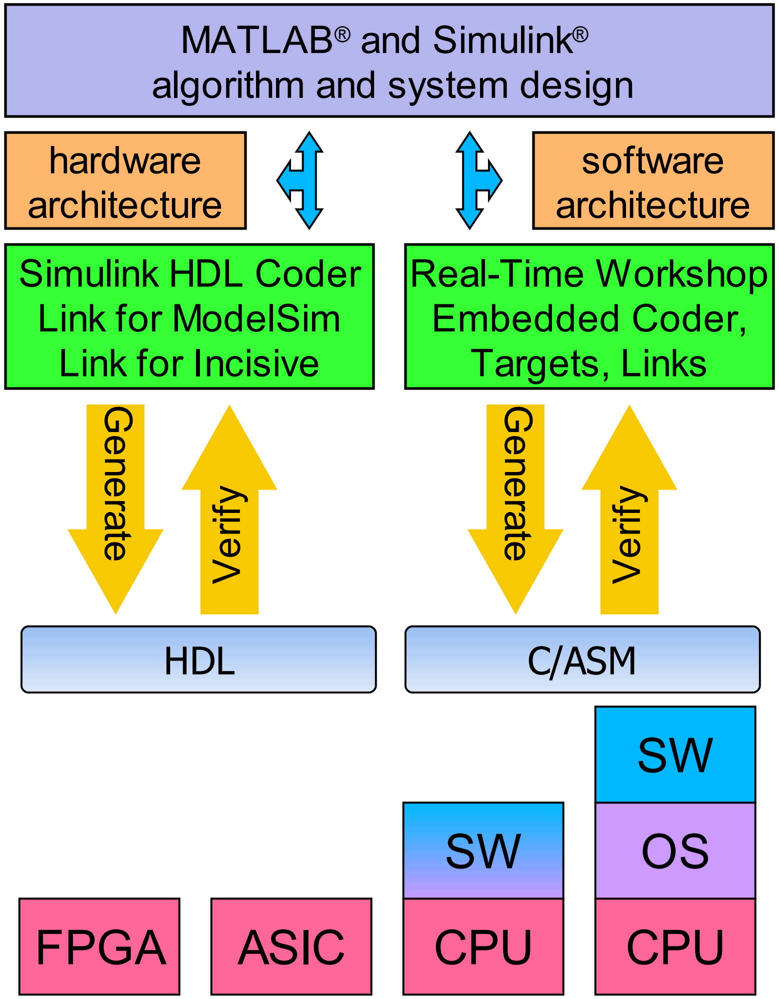
A new frontier …

- No clear HW/SW separation
- Traditionally different design paradigms
- Reconfigurable
  - Hardware and software
  - Adapt to environment
- Novel
  - design paradigms
  - applications



Piecing it all together …

- Compiler
  - Combines
    - Functional design
    - Architecture
  - Generates
    - Hardware (HDL)
    - Software (C/ASM)
- Simulator
  - Explore design space
  - Verify design choices
    - Hardware in the loop
    - Processor in the loop
    - Silicon in the loop
    - …



Agenda

- Model-Based Design
- Model elaboration demonstration
- Implementing a functional specification
- Summary

Summary

- Model-Based Design
  - Enterprise-wide usage
- Model transformation is everywhere
  - Design elaboration
  - Implementation design
- Strict hardware/software separation is fading
  - For example, cross discipline optimizations

Acknowledgments

- Jason Ghidella
- Mirko Conrad
- Hans Vangheluwe
- Ben Denckla
- Jan Madsen
- All participants of the annual Computer Automated Multiparadigm Modeling (CAMPaM) workshop at the McGill University Bellairs Campus, Barbados
- …