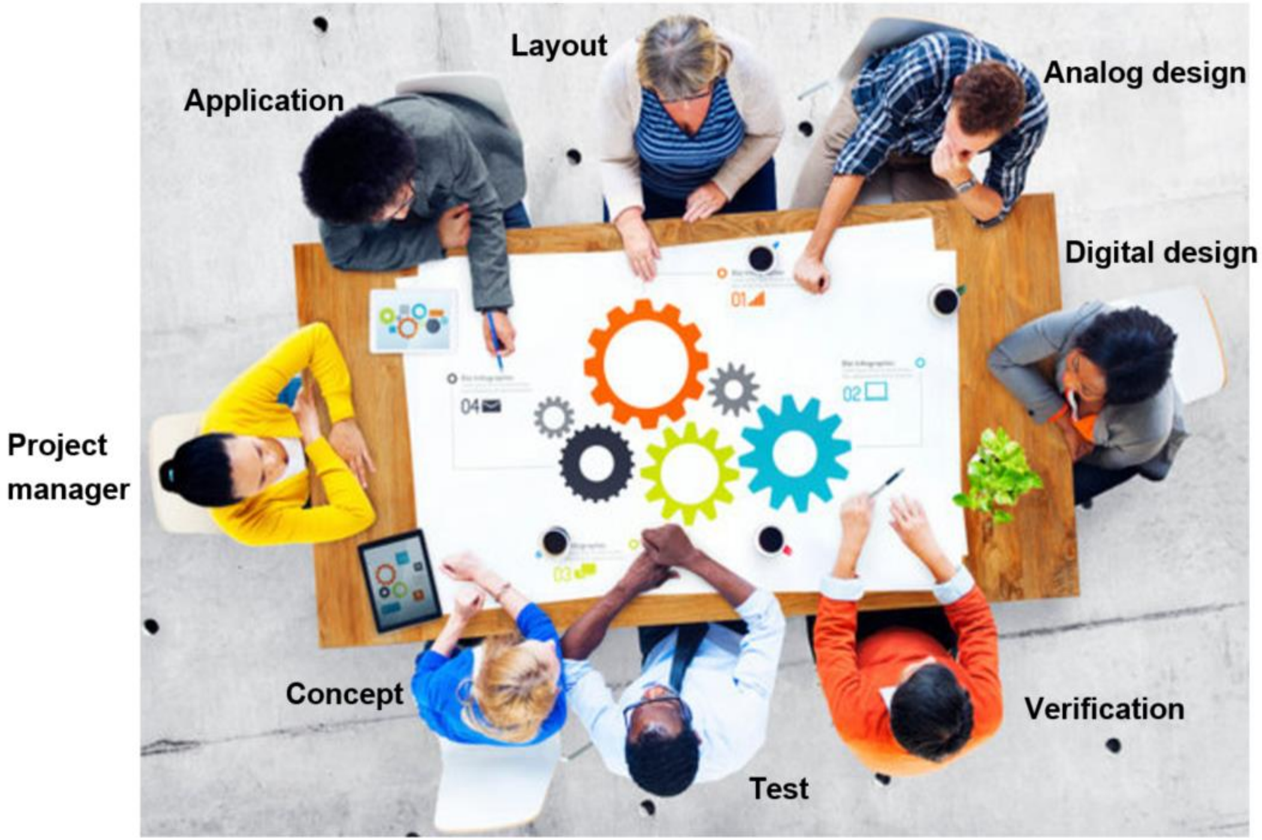
Fast and Furious
Quick Innovation from Idea to Real Prototype

Simone Fontanesi, Infineon Technologies Austria AG, Villach, Austria
Gaetano Formato, Infineon Technologies Austria AG, Villach, Austria
Thomas Arndt, COSEDA Technologies GmbH, Dresden, Germany
Andrea Monterastelli, Infineon Technologies Austria AG, Villach, Austria

Agenda

Selling an idea with a prototype

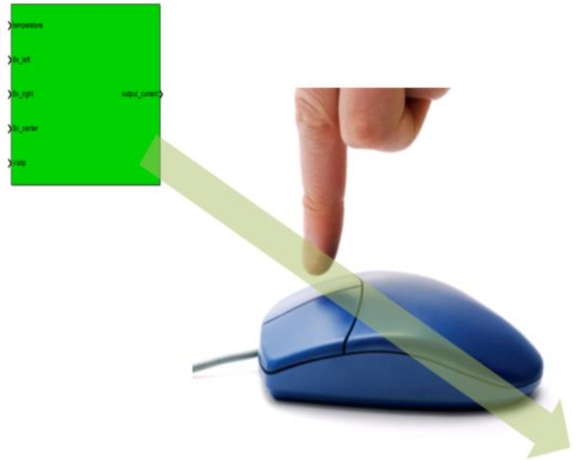
A possible approach…



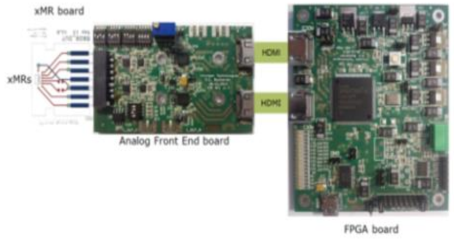
…Our vision!

Concept/Application engineer with a smart product concept idea

From model…

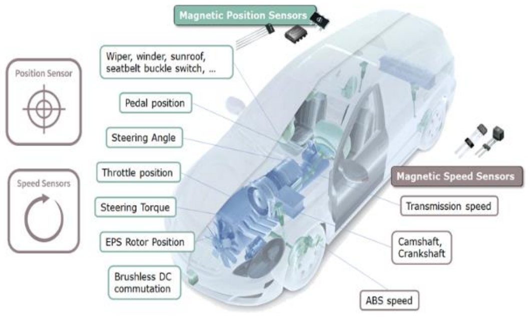


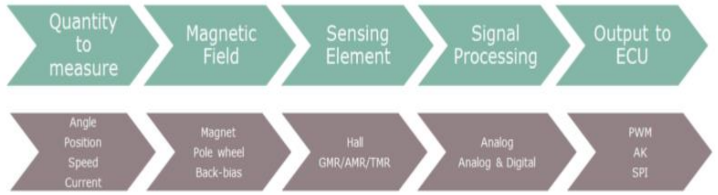
…to prototype!

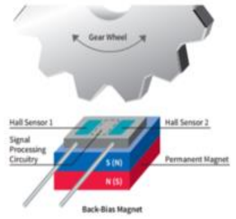


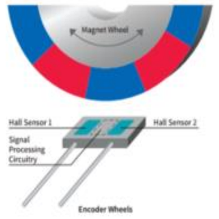
Agenda

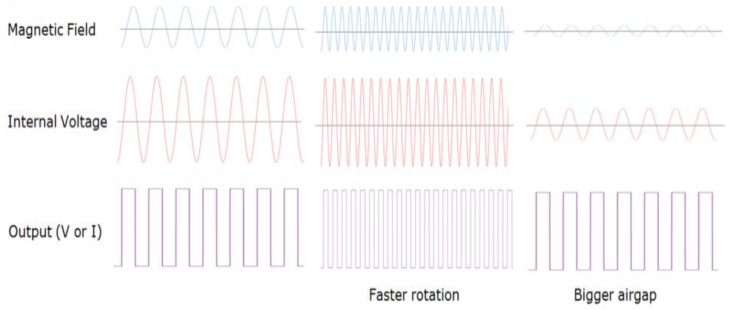
Magnetic sensors for automotive applications



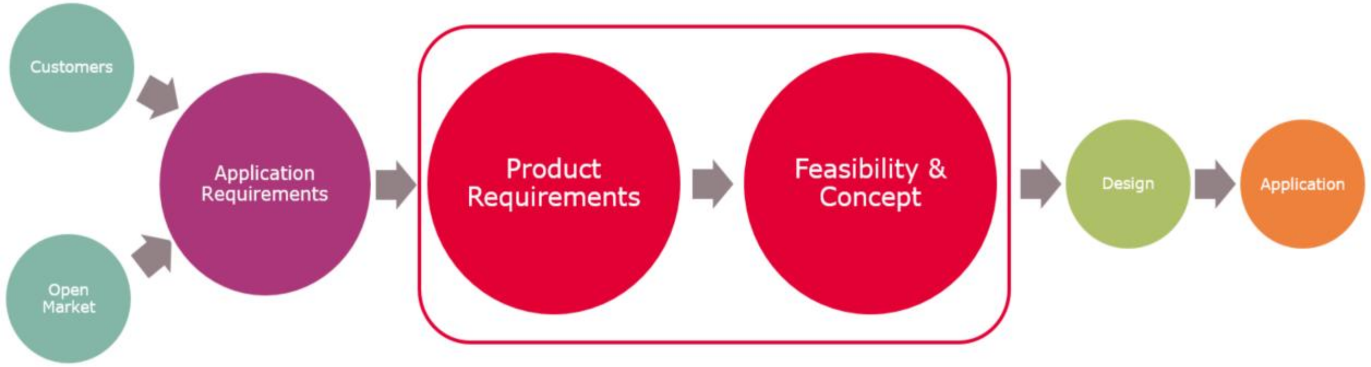


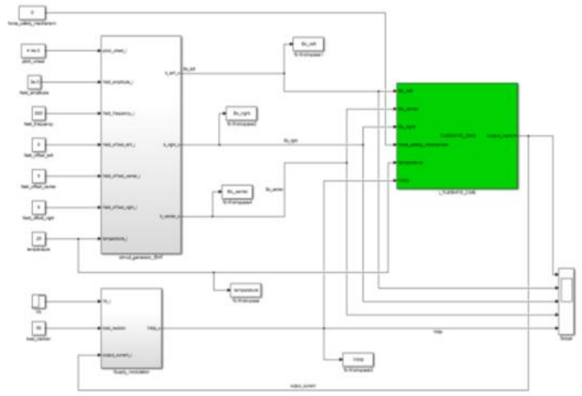


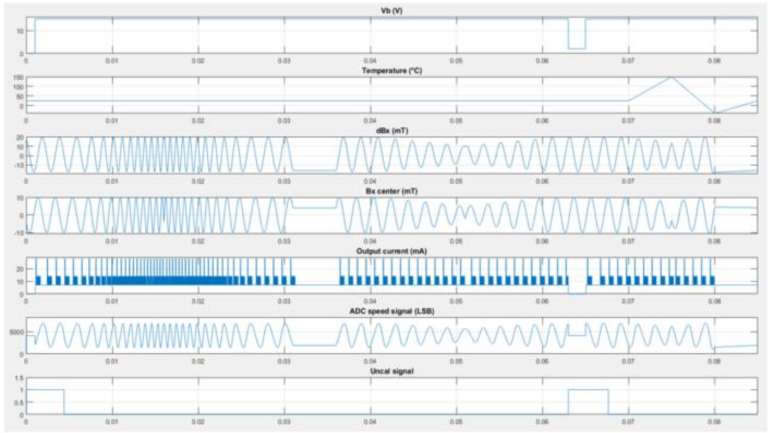




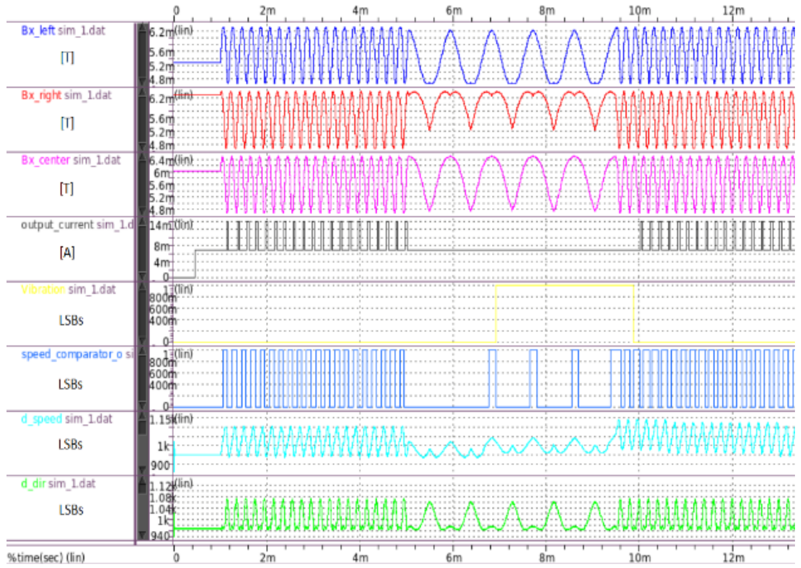
Virtual prototyping for concept definition



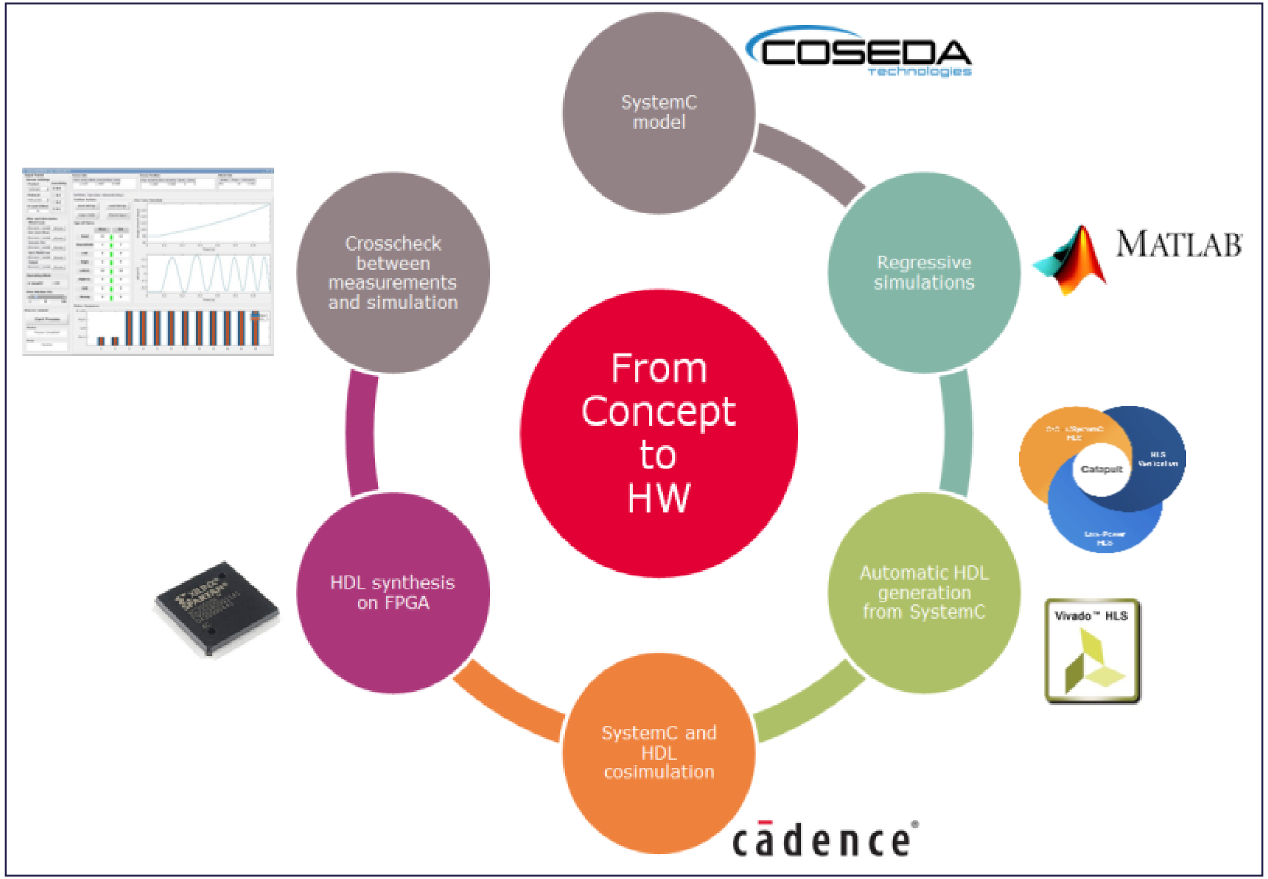




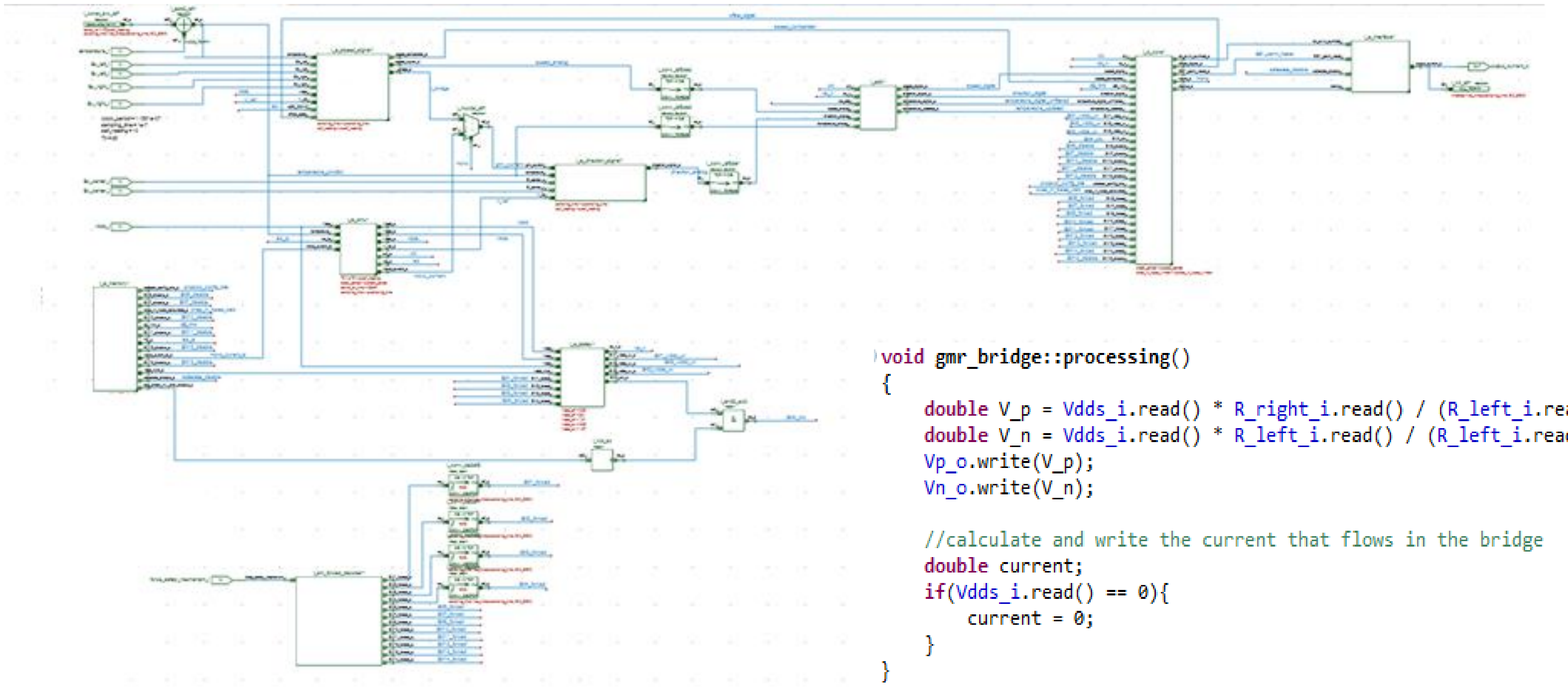
From simulation to real HW

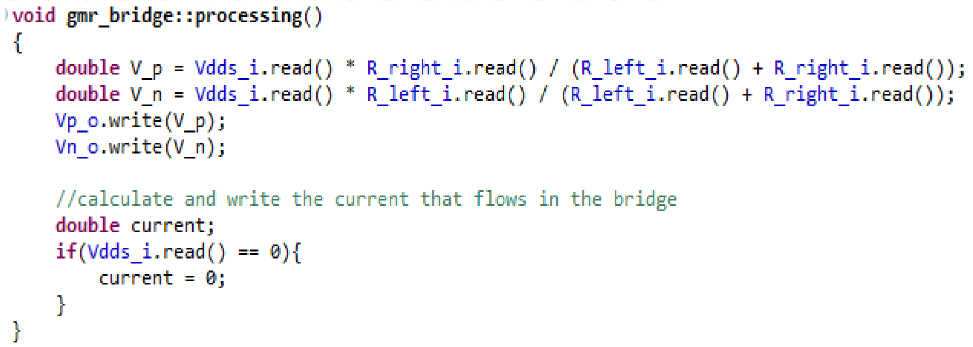


Fast & Furious: the methodology in a nutshell



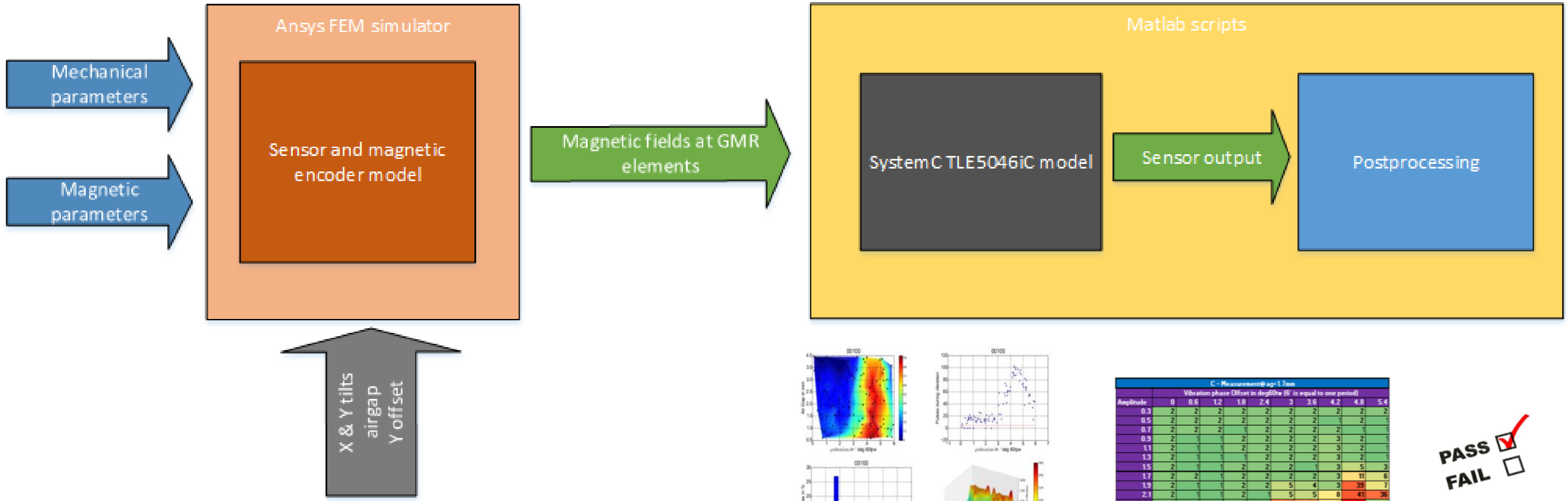
SystemC Modeling

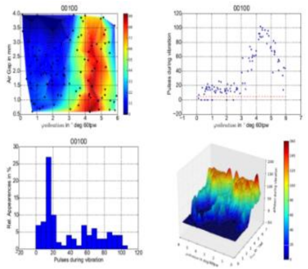


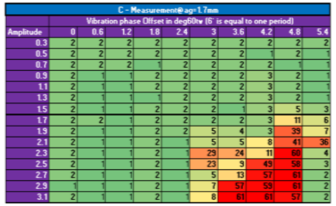


```
void gmr_bridge::processing()
{
    double V_p = Vdds_i.read() * R_right_i.read() / (R_left_i.read() + R_right_i.read());
    double V_n = Vdds_i.read() * R_left_i.read() / (R_left_i.read() + R_right_i.read());
    Vp_o.write(V_p);
    Vn_o.write(V_n);

    //calculate and write the current that flows in the bridge
    double current;
    if(Vdds_i.read() == 0){
        current = 0;
    }
}
```

Regressive simulations



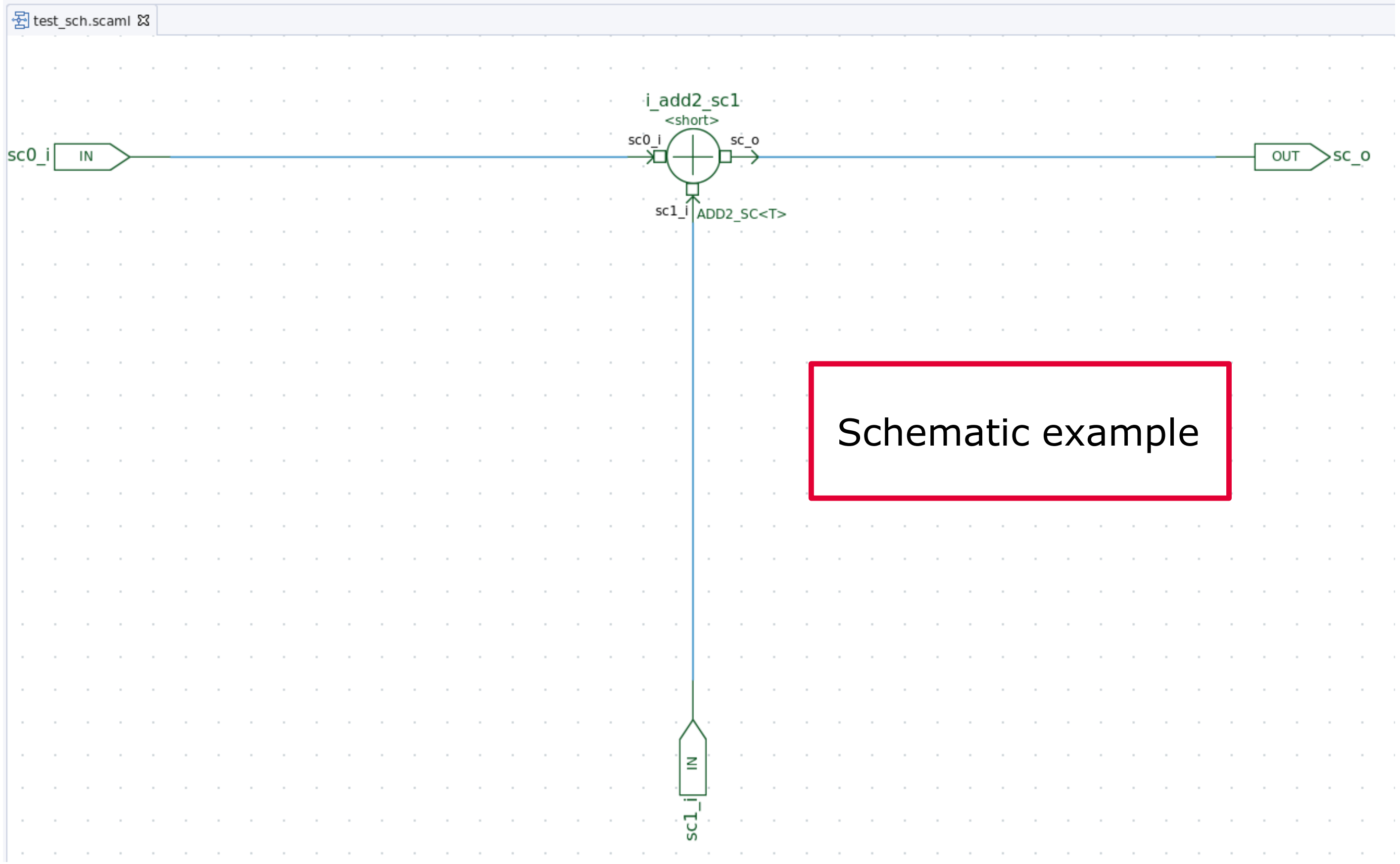




C - Measurement@ag=1.7mm										
	Vibration phase Offset in deg60tw (6° is equal to one period)									
Amplitude	0	0.6	1.2	1.8	2.4	3	3.6	4.2	4.8	5.4
0.3	2	2	2	2	2	2	2	2	2	2
0.5	2	2	2	2	2	2	2	1	2	1
0.7	2	2	2	1	2	2	2	2	1	1
0.9	2	1	1	2	2	2	2	3	2	1
1.1	2	1	1	2	2	2	2	3	2	1
1.3	2	1	1	1	2	2	2	3	2	1
1.5	2	1	1	2	2	2	1	3	5	3
1.7	2	2	1	2	2	2	2	3	11	6
1.9	2	1	1	2	2	5	4	3	39	7
2.1	2	1	1	2	1	5	5	8	41	36
2.3	2	1	1	2	1	29	24	11	60	4
2.5	2	1	1	2	1	23	9	49	58	3
2.7	2	1	1	2	1	5	13	57	61	2
2.9	1	1	1	2	1	7	57	59	61	2
3.1	2	1	1	2	1	8	61	61	57	2

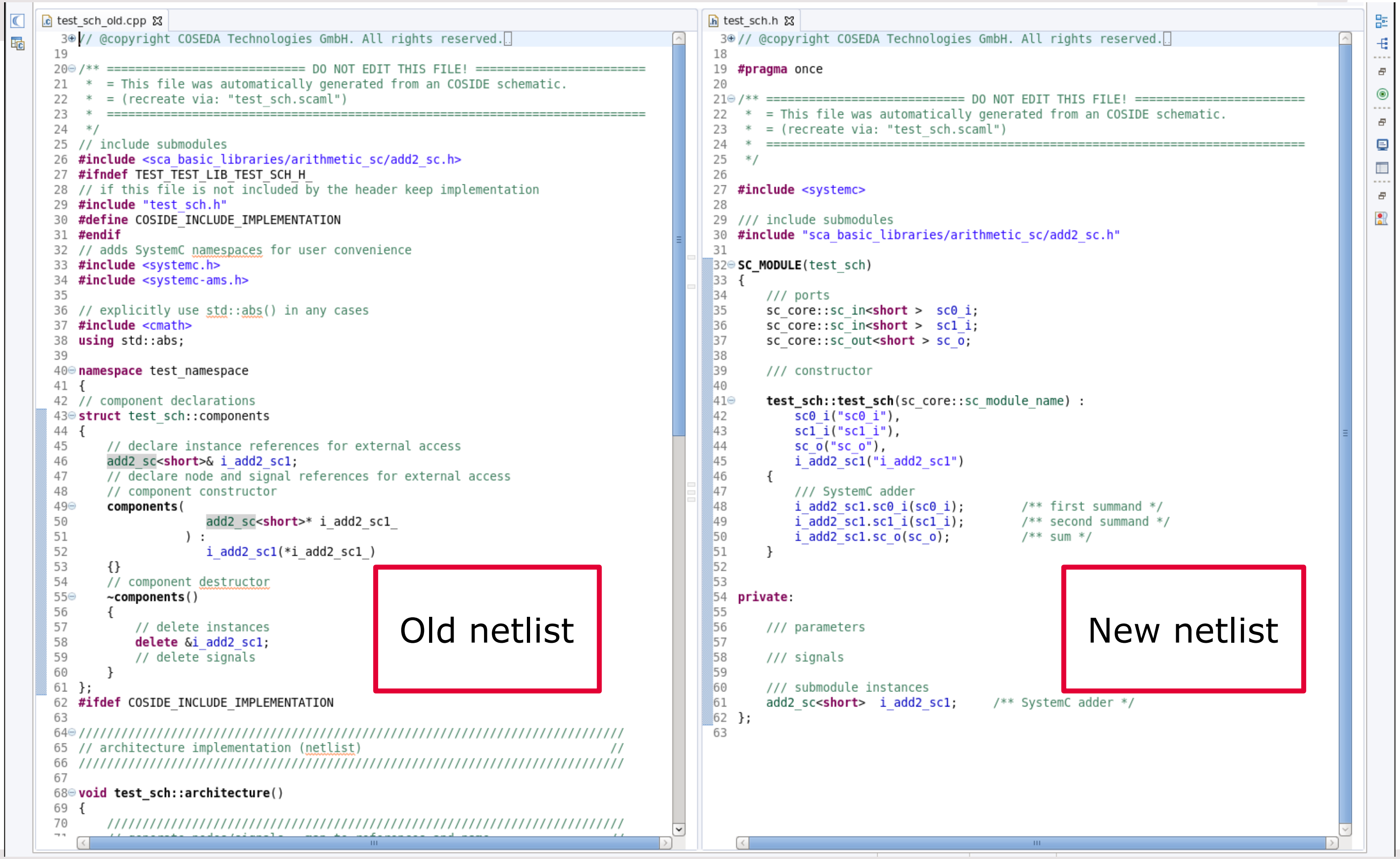
PASS [✓]
FAIL ☐

Coside clean netlist generation 1/3



Schematic example

Coside clean netlist generation 2/3



test_sch_old.cpp

```
// @copyright COSEDA Technologies GmbH. All rights reserved.

/** ============================ DO NOT EDIT THIS FILE! ========================
 *  = This file was automatically generated from an COSIDE schematic.
 *  = (recreate via: "test_sch.scaml")
 *  ============================================================================
 */
// include submodules
#include <sca_basic_libraries/arithmetic_sc/add2_sc.h>
#ifndef TEST_TEST_LIB_TEST_SCH_H_
// if this file is not included by the header keep implementation
#include "test_sch.h"
#define COSIDE_INCLUDE_IMPLEMENTATION
#endif
// adds SystemC namespaces for user convenience
#include <systemc.h>
#include <systemc-ams.h>

// explicitly use std::abs() in any cases
#include <cmath>
using std::abs;

namespace test_namespace
{
// component declarations
struct test_sch::components
{
    // declare instance references for external access
    add2_sc<short>& i_add2_sc1;
    // declare node and signal references for external access
    // component constructor
    components(
                  add2_sc<short>* i_add2_sc1_
              ) :
                  i_add2_sc1(*i_add2_sc1_)
    {}
    // component destructor
    ~components()
    {
        // delete instances
        delete &i_add2_sc1;
        // delete signals
    }
};
#ifdef COSIDE_INCLUDE_IMPLEMENTATION

//////////////////////////////////////////////////////////////////////////////
// architecture implementation (netlist)                                    //
//////////////////////////////////////////////////////////////////////////////

void test_sch::architecture()
{
    //////////////////////////////////////////////////////////////////////////
```

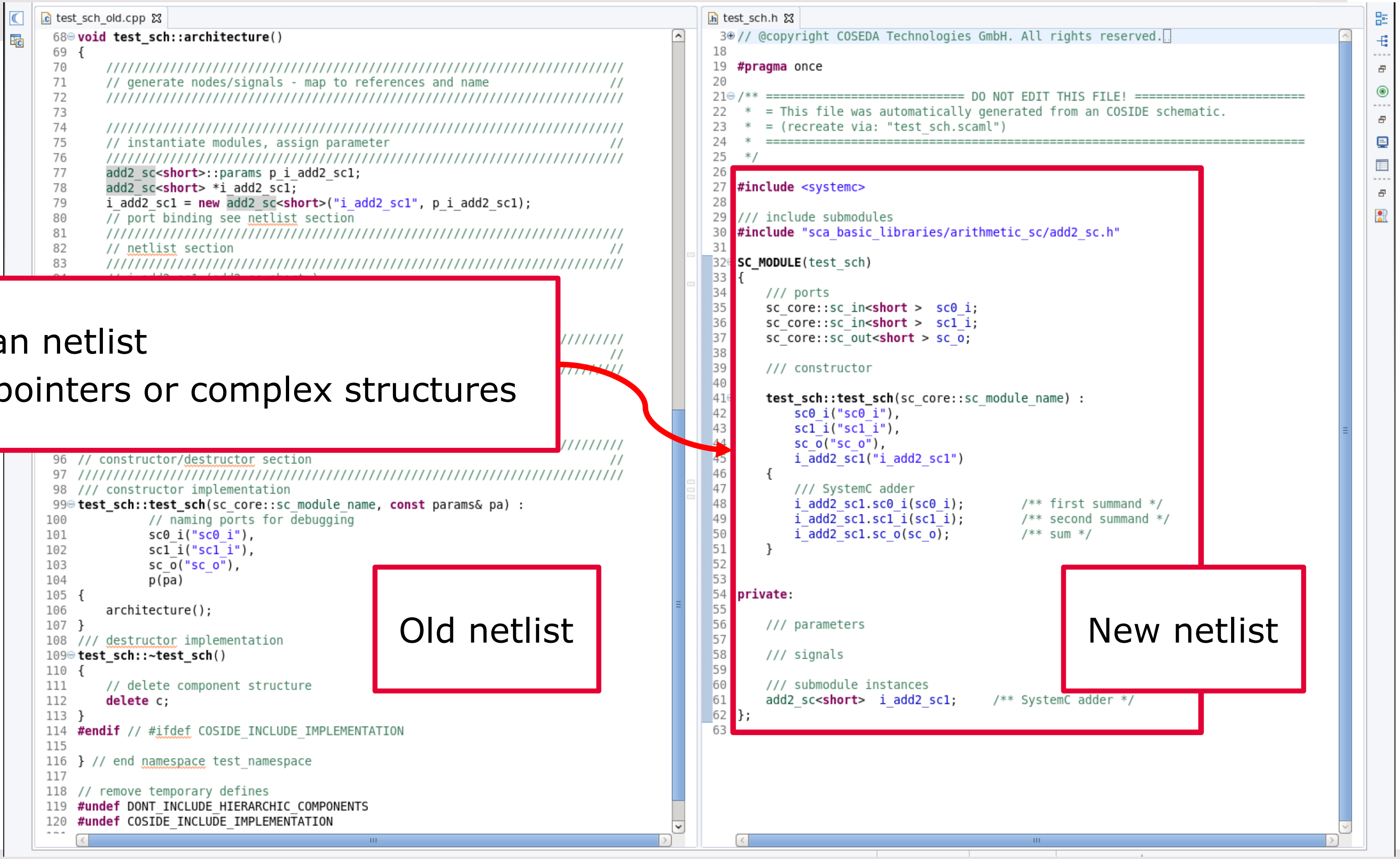
test_sch.h

```
// @copyright COSEDA Technologies GmbH. All rights reserved.

#pragma once

/** ============================ DO NOT EDIT THIS FILE! ========================
 *  = This file was automatically generated from an COSIDE schematic.
 *  = (recreate via: "test_sch.scaml")
 *  ============================================================================
 */

#include <systemc>

/// include submodules
#include "sca_basic_libraries/arithmetic_sc/add2_sc.h"

SC_MODULE(test_sch)
{
    /// ports
    sc_core::sc_in<short >  sc0_i;
    sc_core::sc_in<short >  sc1_i;
    sc_core::sc_out<short > sc_o;

    /// constructor

    test_sch::test_sch(sc_core::sc_module_name) :
        sc0_i("sc0_i"),
        sc1_i("sc1_i"),
        sc_o("sc_o"),
        i_add2_sc1("i_add2_sc1")
    {
        /// SystemC adder
        i_add2_sc1.sc0_i(sc0_i);        /** first summand */
        i_add2_sc1.sc1_i(sc1_i);        /** second summand */
        i_add2_sc1.sc_o(sc_o);          /** sum */
    }


private:

    /// parameters

    /// signals

    /// submodule instances
    add2_sc<short>  i_add2_sc1;     /** SystemC adder */
};

```

Old netlist

New netlist

Coside clean netlist generation 3/3



test_sch_old.cpp

```cpp
void test_sch::architecture()
{
    /////////////////////////////////////////////////////////////////////////
    // generate nodes/signals - map to references and name                //
    /////////////////////////////////////////////////////////////////////////

    /////////////////////////////////////////////////////////////////////////
    // instantiate modules, assign parameter                              //
    /////////////////////////////////////////////////////////////////////////
    add2_sc<short>::params p_i_add2_sc1;
    add2_sc<short> *i_add2_sc1;
    i_add2_sc1 = new add2_sc<short>("i_add2_sc1", p_i_add2_sc1);
    // port binding see netlist section
    /////////////////////////////////////////////////////////////////////////
    // netlist section                                                    //
    /////////////////////////////////////////////////////////////////////////
```

```cpp
/////////
       //
/////////

/////////
// constructor/destructor section                                         //
/////////////////////////////////////////////////////////////////////////////
/// constructor implementation
test_sch::test_sch(sc_core::sc_module_name, const params& pa) :
          // naming ports for debugging
          sc0_i("sc0_i"),
          sc1_i("sc1_i"),
          sc_o("sc_o"),
          p(pa)
{
    architecture();
}
/// destructor implementation
test_sch::~test_sch()
{
    // delete component structure
    delete c;
}
#endif // #ifdef COSIDE_INCLUDE_IMPLEMENTATION

} // end namespace test_namespace

// remove temporary defines
#undef DONT_INCLUDE_HIERARCHIC_COMPONENTS
#undef COSIDE_INCLUDE_IMPLEMENTATION
```

test_sch.h

```cpp
// @copyright COSEDA Technologies GmbH. All rights reserved.

#pragma once

/** =========================== DO NOT EDIT THIS FILE! ========================
 *  = This file was automatically generated from an COSIDE schematic.
 *  = (recreate via: "test_sch.scaml")
 *  ===========================================================================
 */

#include <systemc>

/// include submodules
#include "sca_basic_libraries/arithmetic_sc/add2_sc.h"

SC_MODULE(test_sch)
{
    /// ports
    sc_core::sc_in<short >  sc0_i;
    sc_core::sc_in<short >  sc1_i;
    sc_core::sc_out<short > sc_o;

    /// constructor

    test_sch::test_sch(sc_core::sc_module_name) :
        sc0_i("sc0_i"),
        sc1_i("sc1_i"),
        sc_o("sc_o"),
        i_add2_sc1("i_add2_sc1")
    {
        /// SystemC adder
        i_add2_sc1.sc0_i(sc0_i);        /** first summand */
        i_add2_sc1.sc1_i(sc1_i);        /** second summand */
        i_add2_sc1.sc_o(sc_o);          /** sum */
    }


private:

    /// parameters

    /// signals

    /// submodule instances
    add2_sc<short>  i_add2_sc1;     /** SystemC adder */
};
```

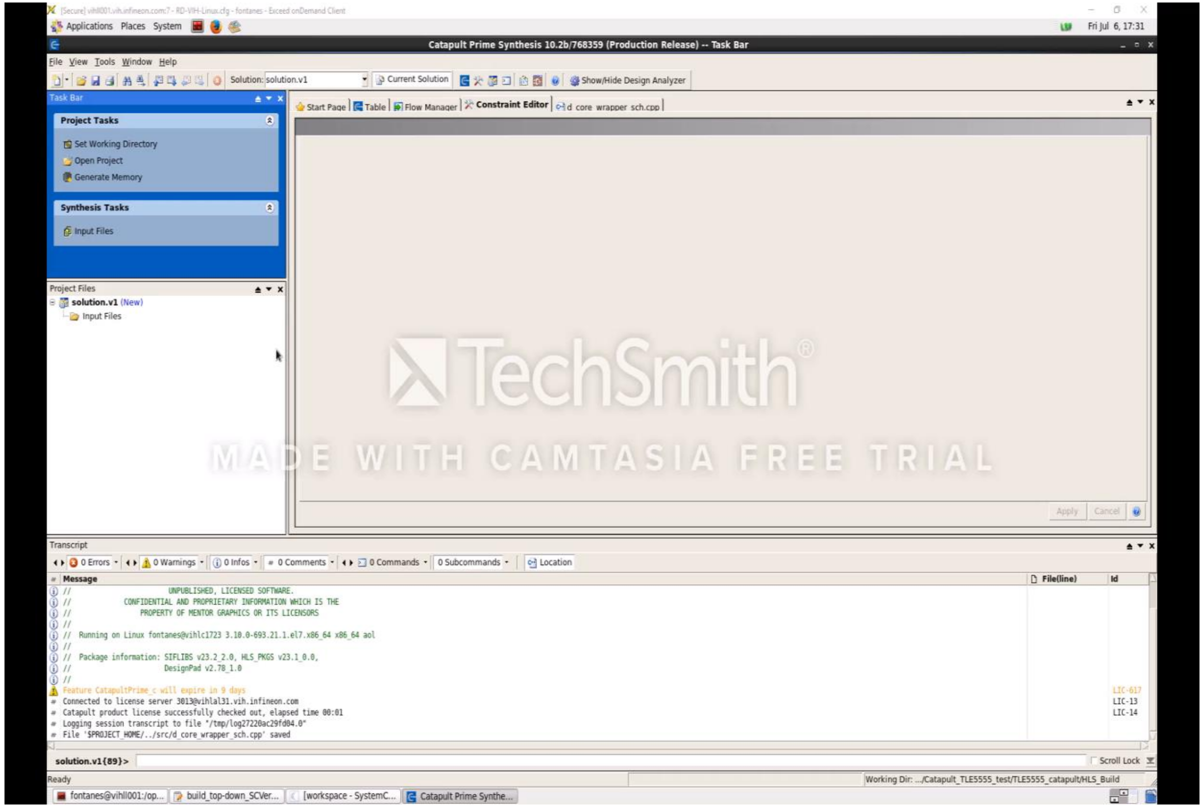
Clean netlist
No pointers or complex structures

Old netlist

New netlist

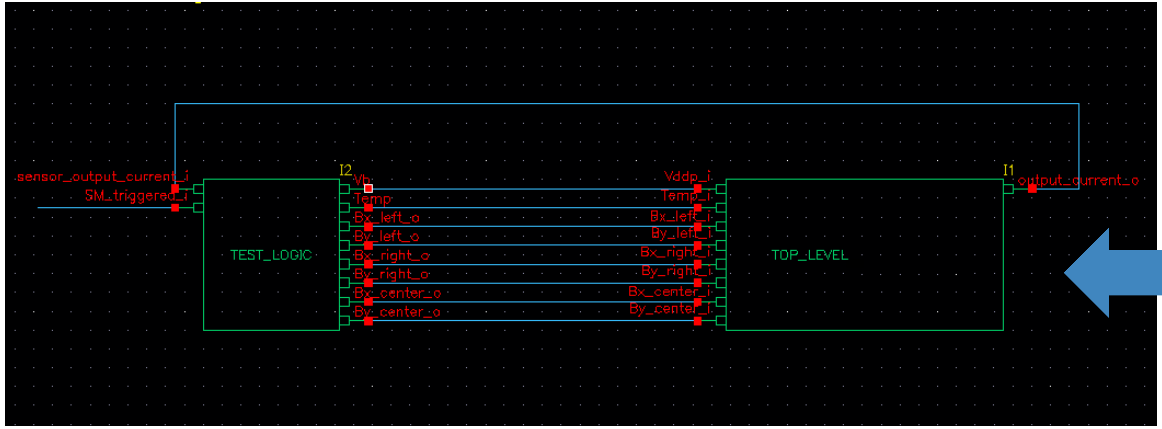
From SystemC to HDL in a few clicks

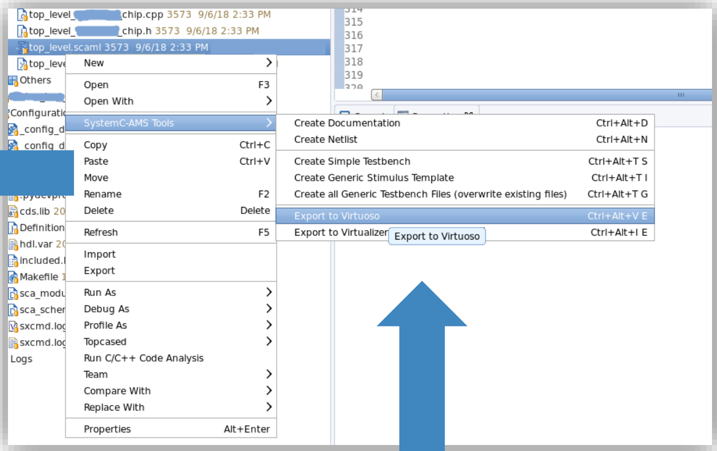
- SystemC „clean“ netlist from COSIDE®
- Conversion of each SystemC module
- Conversion of top-level
- High level synthesis
  - Vivado HLS
  - Mentor Catapult



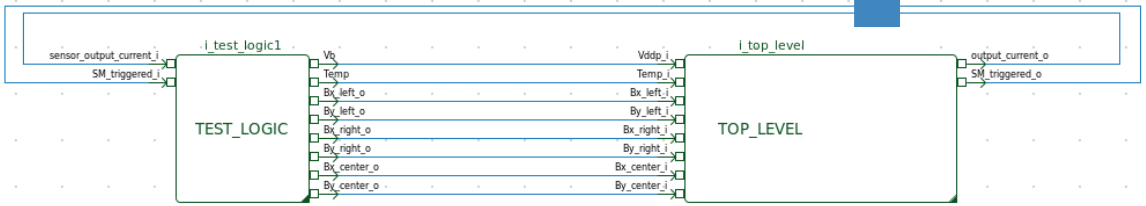
```
//                     UNPUBLISHED, LICENSED SOFTWARE.
//            CONFIDENTIAL AND PROPRIETARY INFORMATION WHICH IS THE
//                PROPERTY OF MENTOR GRAPHICS OR ITS LICENSORS
//
//  Running on Linux fontanes@vihlc1723 3.10.0-693.21.1.el7.x86_64 x86_64 aol
//
//  Package information: SIFLIBS v23.2_2.0, HLS_PKGS v23.1_0.0,
//                       DesignPad v2.78_1.0
//
Feature CatapultPrime_c will expire in 9 days
Connected to license server 3013@vihlal31.vih.infineon.com
Catapult product license successfully checked out, elapsed time 00:01
Logging session transcript to file "/tmp/log27220ac29fd04.0"
File '$PROJECT_HOME/../src/d_core_wrapper_sch.cpp' saved
```

Coside CCB – SystemC to Virtuoso Export 1/3



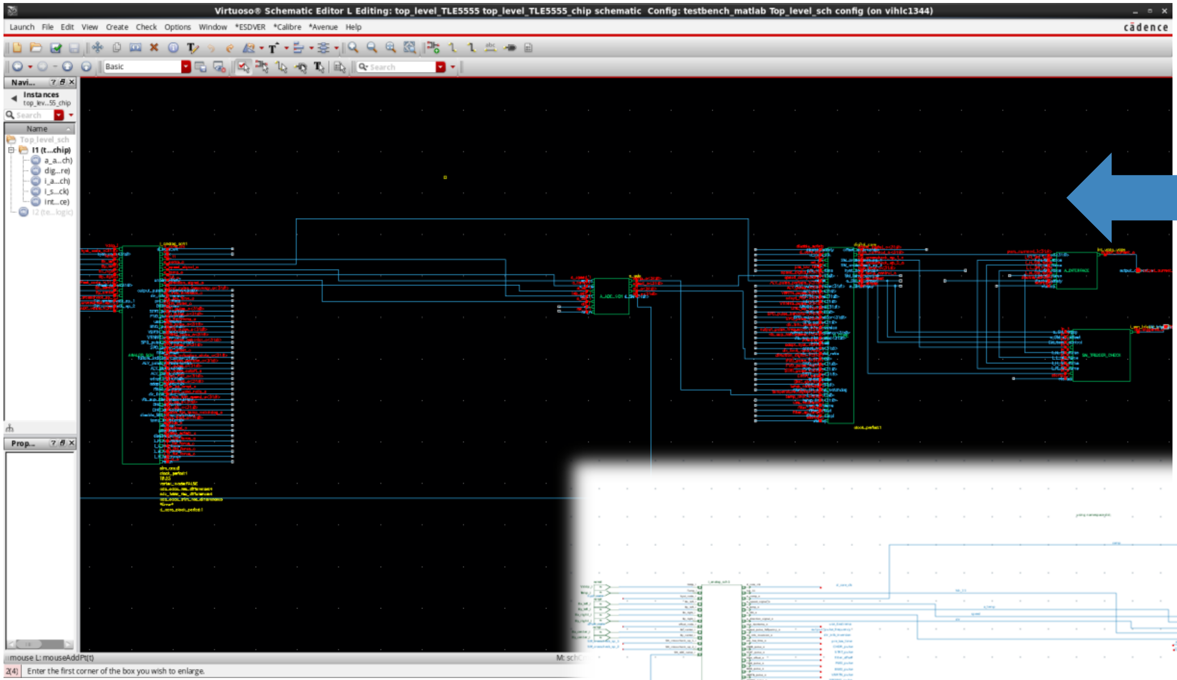


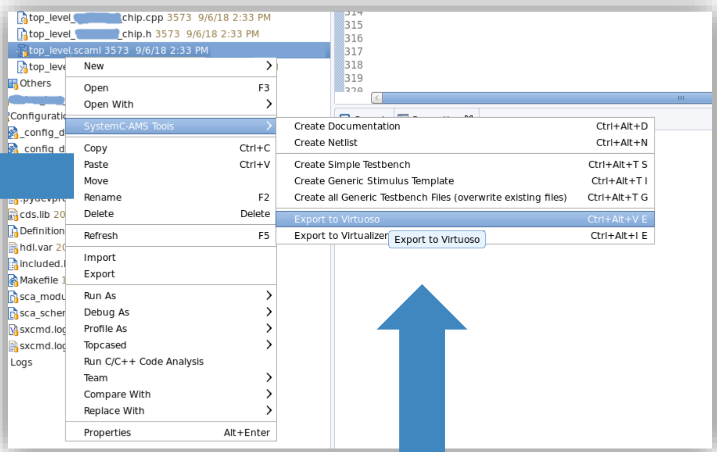
top_level_ chip.cpp 3573 9/6/18 2:33 PM
top_level_ chip.h 3573 9/6/18 2:33 PM
top_level.scaml 3573 9/6/18 2:33 PM
Others
Configuratio
_config_d
cds.lib
Definition
hdl.var
included.
Makefile
sca_modu
sca_scher
sxcmd.log
Logs
New
Open F3
Open With
SystemC-AMS Tools
Copy Ctrl+C
Paste Ctrl+V
Move
Rename F2
Delete Delete
Refresh F5
Import
Export
Run As
Debug As
Profile As
Topcased
Run C/C++ Code Analysis
Team
Compare With
Replace With
Properties Alt+Enter
Create Documentation Ctrl+Alt+D
Create Netlist Ctrl+Alt+N
Create Simple Testbench Ctrl+Alt+T S
Create Generic Stimulus Template Ctrl+Alt+T I
Create all Generic Testbench Files (overwrite existing files) Ctrl+Alt+T G
Export to Virtuoso Ctrl+Alt+V E
Export to Virtualizer Ctrl+Alt+I E
Export to Virtuoso

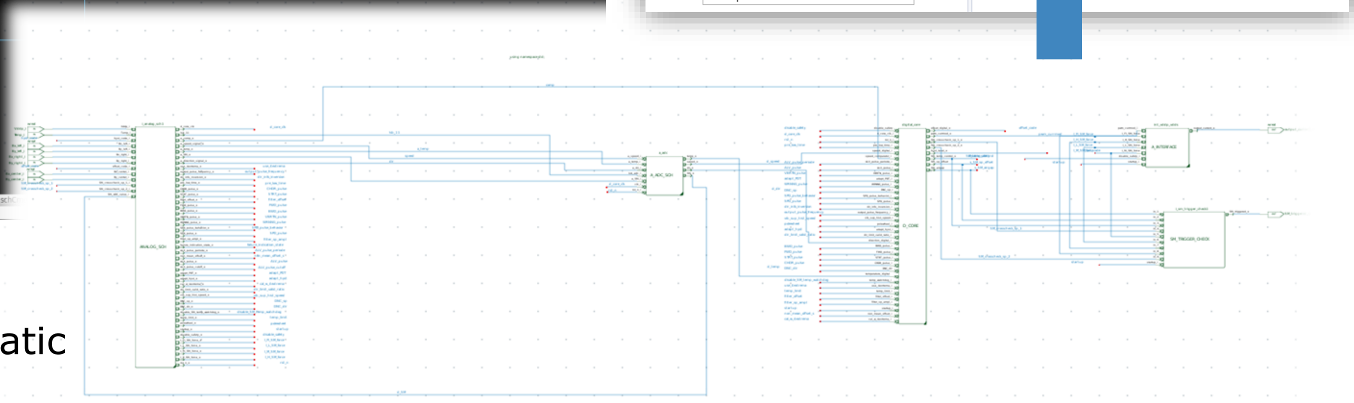


› Export of the top level testbench

Coside CCB – SystemC to Virtuoso Export 2/3

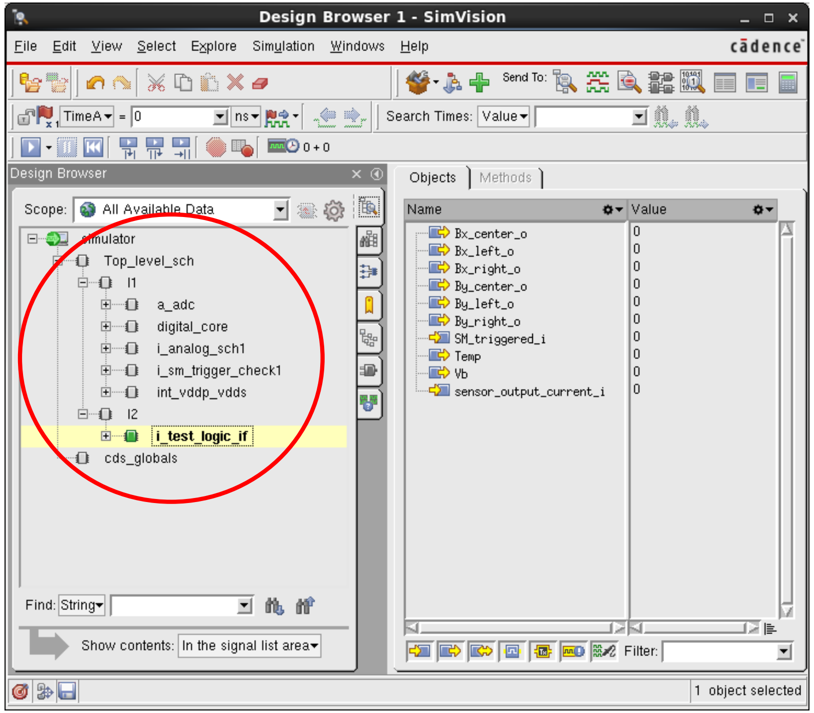

Virtuoso® Schematic Editor L Editing: top_level_TLE5555 top_level_TLE5555_chip schematic Config: testbench_matlab Top_level_sch config (on vihlc1344)
Launch File Edit View Create Check Options Window *ESDVER *Calibre *Avenue Help
cādence
mouse L: mouseAddPt(t)
Enter the first corner of the box you wish to enlarge.


top_level_[redacted]chip.cpp 3573 9/6/18 2:33 PM
top_level_[redacted]chip.h 3573 9/6/18 2:33 PM
top_level.scaml 3573 9/6/18 2:33 PM
New
Open F3
Open With
SystemC-AMS Tools
Copy Ctrl+C
Paste Ctrl+V
Move
Rename F2
Delete Delete
Refresh F5
Import
Export
Run As
Debug As
Profile As
Topcased
Run C/C++ Code Analysis
Team
Compare With
Replace With
Properties Alt+Enter
Create Documentation Ctrl+Alt+D
Create Netlist Ctrl+Alt+N
Create Simple Testbench Ctrl+Alt+T S
Create Generic Stimulus Template Ctrl+Alt+T I
Create all Generic Testbench Files (overwrite existing files) Ctrl+Alt+T G
Export to Virtuoso Ctrl+Alt+V E
Export to Virtualizer Ctrl+Alt+I E
Export to Virtuoso



› Export of the DUT schematic

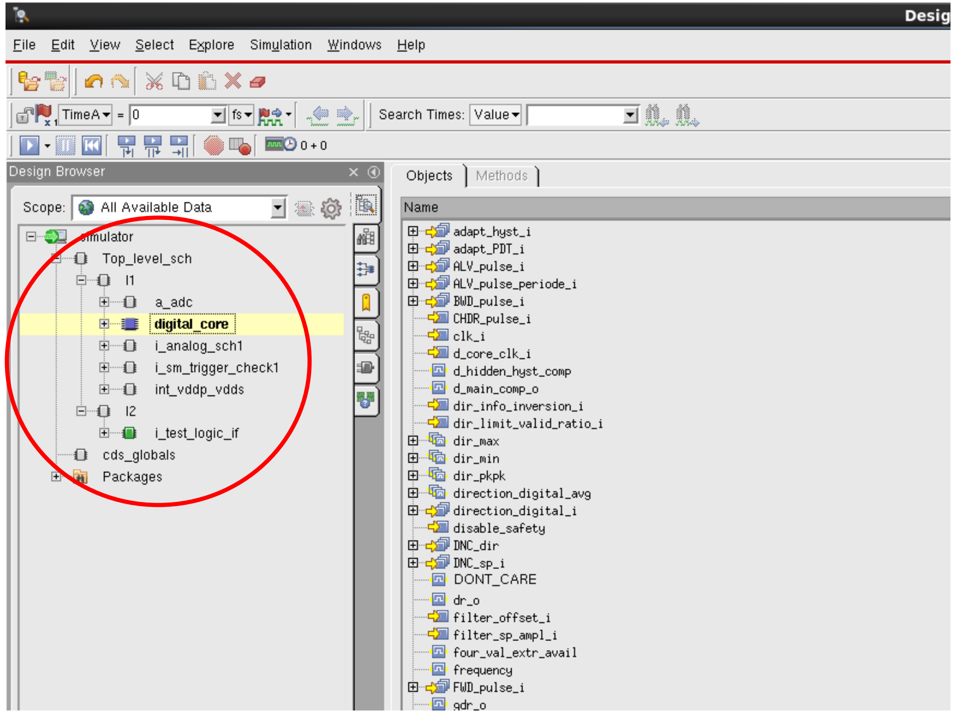
Coside CCB – SystemC to Virtuoso Export 3/3


Design Browser 1 - SimVision
File Edit View Select Explore Simulation Windows Help
cādence
Send To:
TimeA = 0 ns
Search Times: Value
0 + 0
Design Browser
Scope: All Available Data
Objects Methods
Find: String
Show contents: In the signal list area
Filter:

```
simulator
  Top_level_sch
    I1
      a_adc
      digital_core
      i_analog_sch1
      i_sm_trigger_check1
      int_vddp_vdds
    I2
      i_test_logic_if
  cds_globals
```

Name	Value
Bx_center_o	0
Bx_left_o	0
Bx_right_o	0
By_center_o	0
By_left_o	0
By_right_o	0
SM_triggered_i	0
Temp	0
Vb	0
sensor_output_current_i	0

1 object selected

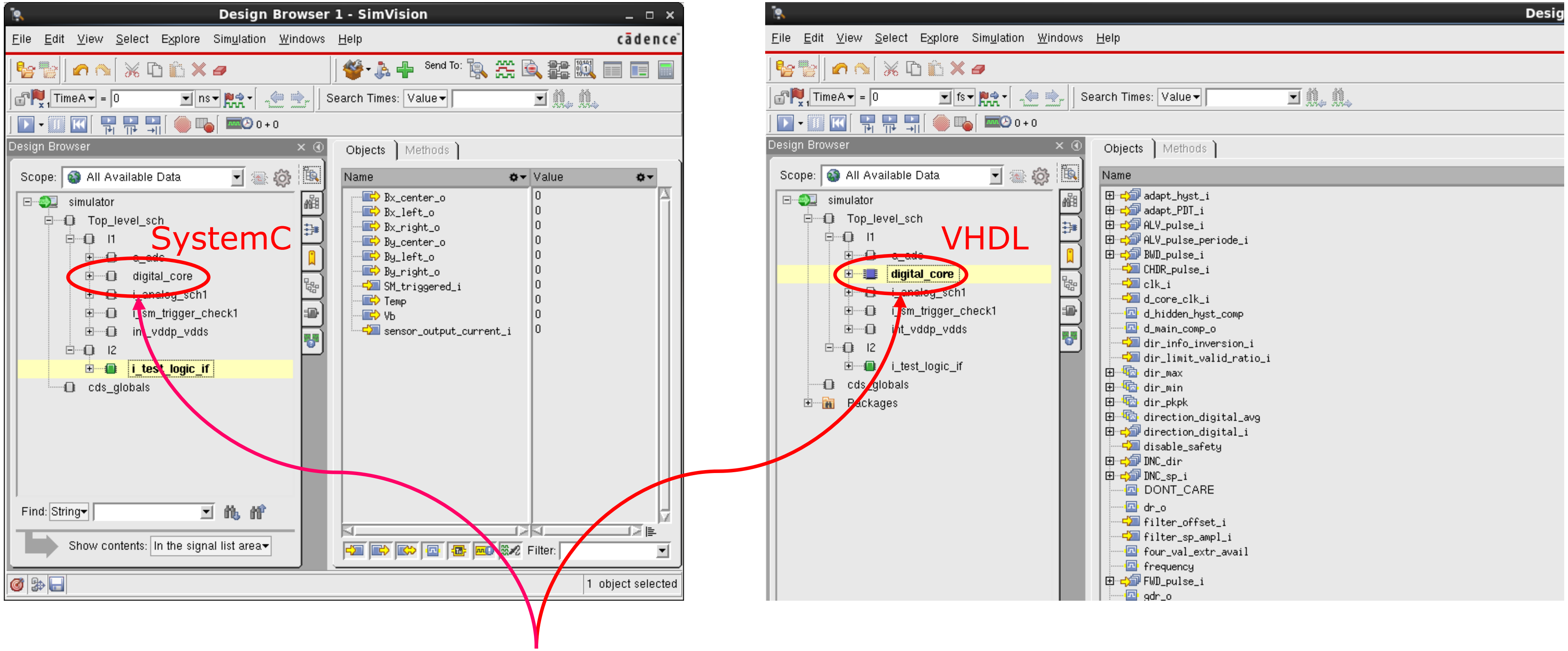

Desig
File Edit View Select Explore Simulation Windows Help
TimeA = 0 fs
Search Times: Value
0 + 0
Design Browser
Scope: All Available Data
Objects Methods

```
simulator
  Top_level_sch
    I1
      a_adc
      digital_core
      i_analog_sch1
      i_sm_trigger_check1
      int_vddp_vdds
    I2
      i_test_logic_if
  cds_globals
  Packages
```

Name
adapt_hyst_i
adapt_PDT_i
ALV_pulse_i
ALV_pulse_periode_i
BWD_pulse_i
CHDR_pulse_i
clk_i
d_core_clk_i
d_hidden_hyst_comp
d_main_comp_o
dir_info_inversion_i
dir_limit_valid_ratio_i
dir_max
dir_min
dir_pkpk
direction_digital_avg
direction_digital_i
disable_safety
DNC_dir
DNC_sp_i
DONT_CARE
dr_o
filter_offset_i
filter_sp_ampl_i
four_val_extr_avail
frequency
FWD_pulse_i
gdr_o

› Same netlists …

Coside CCB – SystemC to Virtuoso Export 3/3


SystemC
VHDL
Design Browser 1 - SimVision
File Edit View Select Explore Simulation Windows Help
cādence
Send To:
TimeA = 0 ns
Search Times: Value
0 + 0
Design Browser
Objects Methods
Scope: All Available Data
Find: String
Show contents: In the signal list area
Filter:
1 object selected

```
simulator
  Top_level_sch
    I1
      a_adc
      digital_core
      i_analog_sch1
      i_sm_trigger_check1
      int_vddp_vdds
    I2
      i_test_logic_if
  cds_globals
```

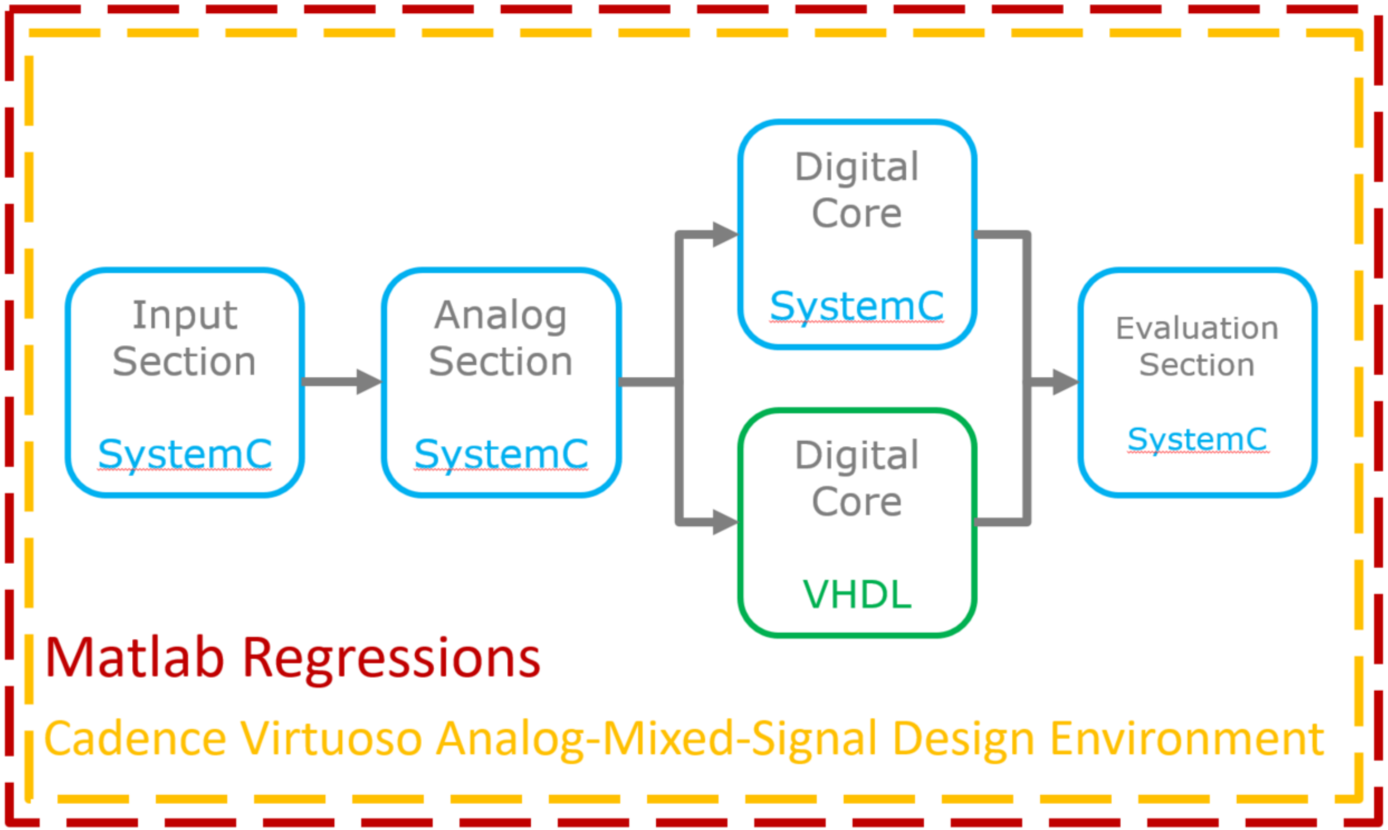
Name	Value
Bx_center_o	0
Bx_left_o	0
Bx_right_o	0
By_center_o	0
By_left_o	0
By_right_o	0
SM_triggered_i	0
Temp	0
Vb	0
sensor_output_current_i	0

```
simulator
  Top_level_sch
    I1
      a_adc
      digital_core
      i_analog_sch1
      i_sm_trigger_check1
      int_vddp_vdds
    I2
      i_test_logic_if
  cds_globals
  Packages
```

Name
adapt_hyst_i
adapt_PDT_i
ALV_pulse_i
ALV_pulse_periode_i
BWD_pulse_i
CHDR_pulse_i
clk_i
d_core_clk_i
d_hidden_hyst_comp
d_main_comp_o
dir_info_inversion_i
dir_limit_valid_ratio_i
dir_max
dir_min
dir_pkpk
direction_digital_avg
direction_digital_i
disable_safety
DNC_dir
DNC_sp_i
DONT_CARE
dr_o
filter_offset_i
filter_sp_ampl_i
four_val_extr_avail
frequency
FWD_pulse_i
gdr_o

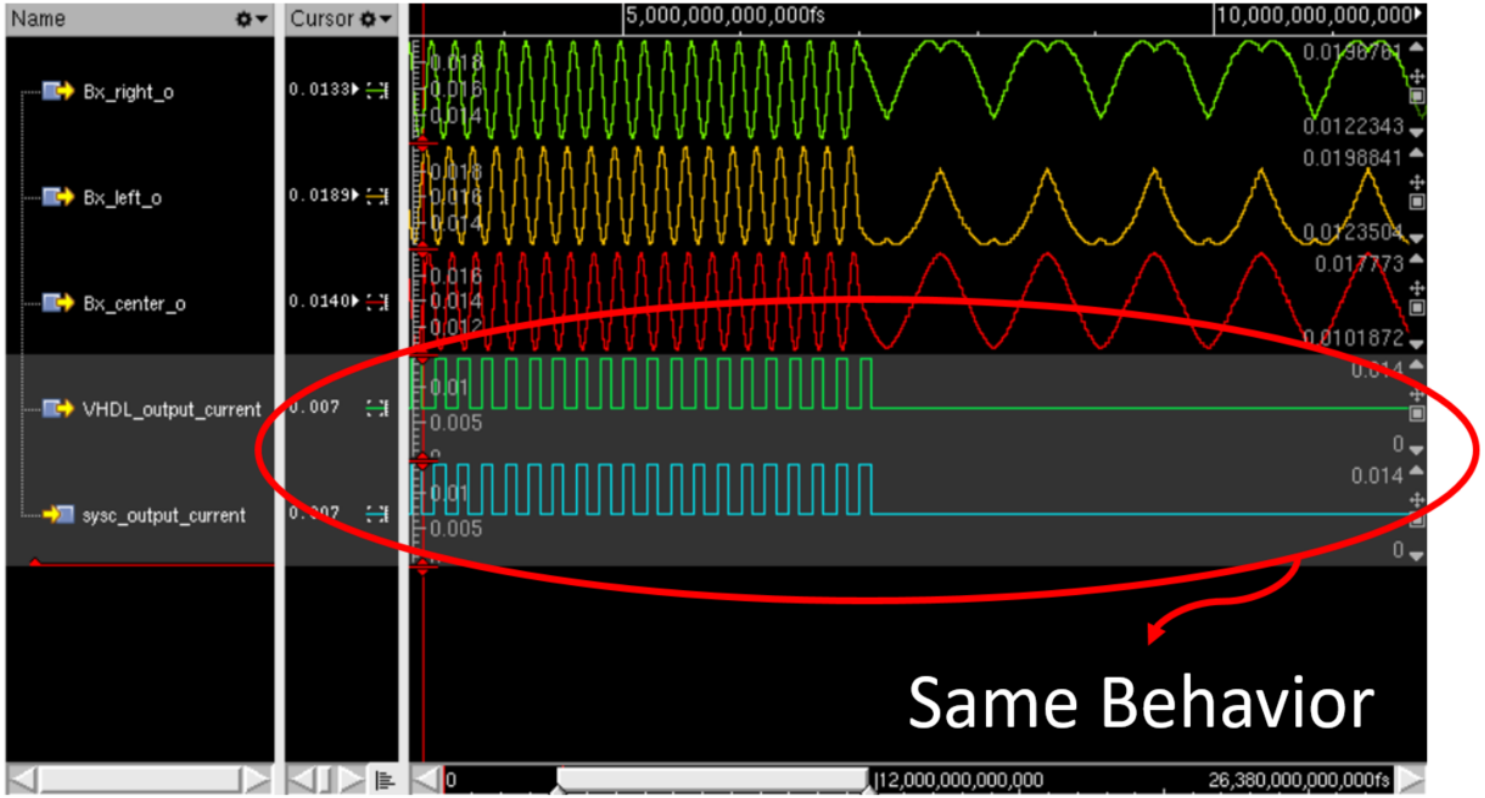
› … But different digital cores instances

SystemC & VHDL cosimulation (1/2)

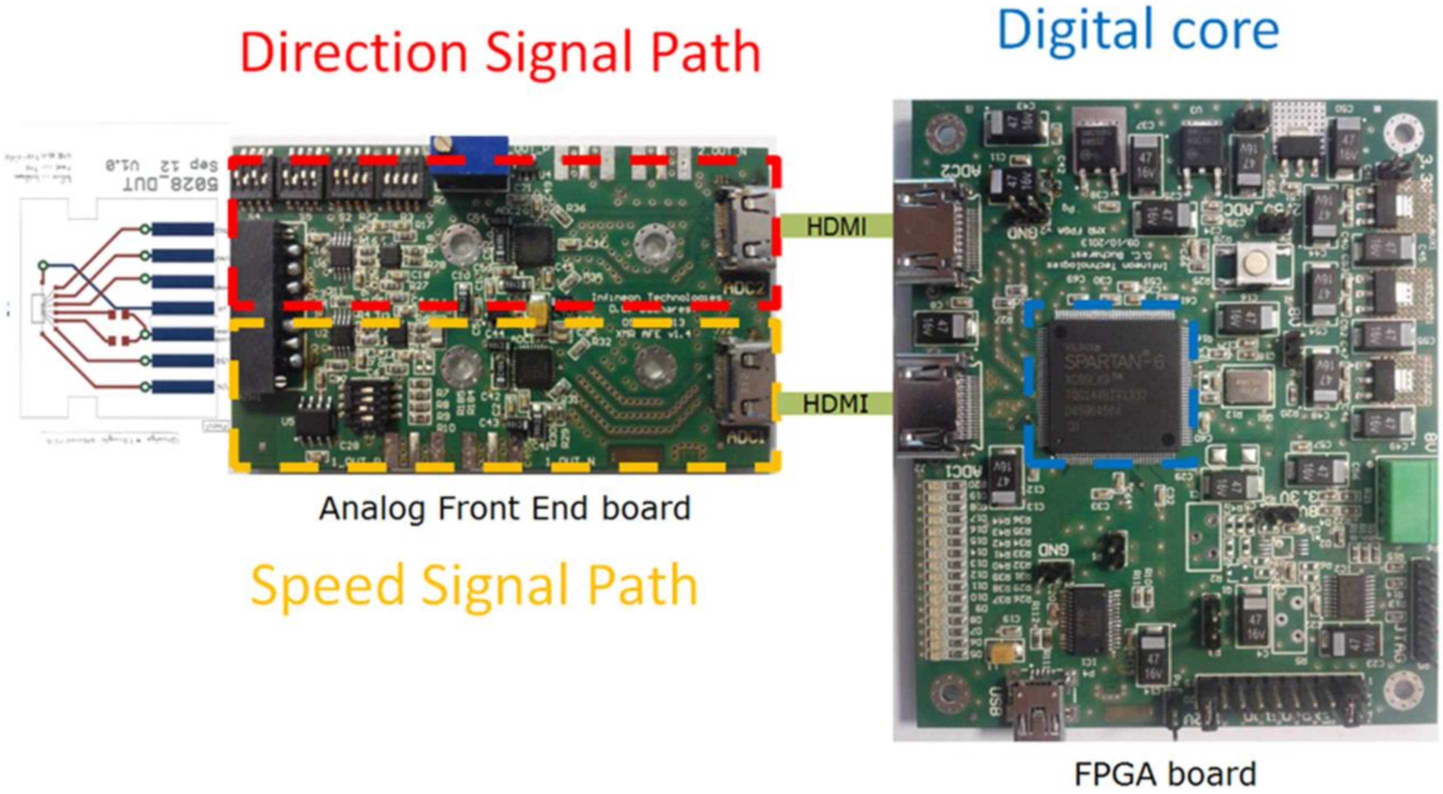


*Note: Mentor Catapult would also allow cosimulation in an integrated environment

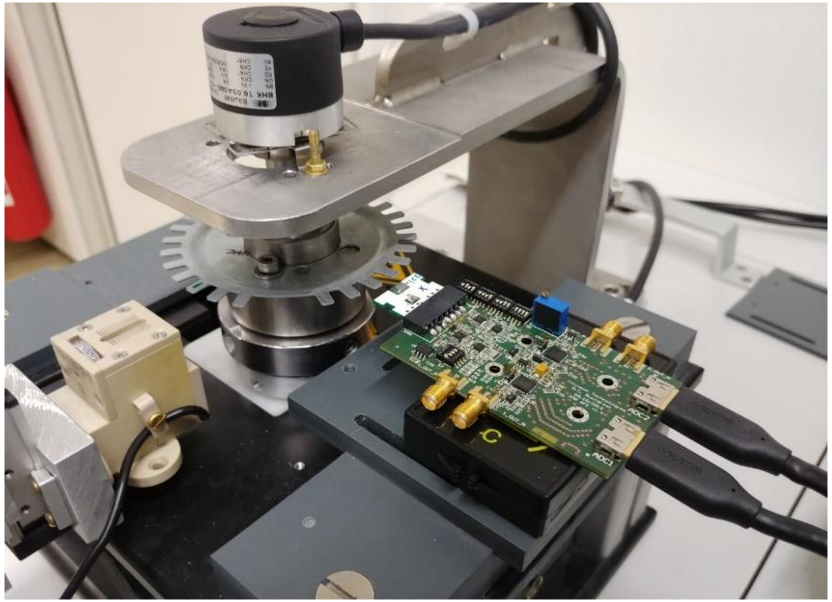
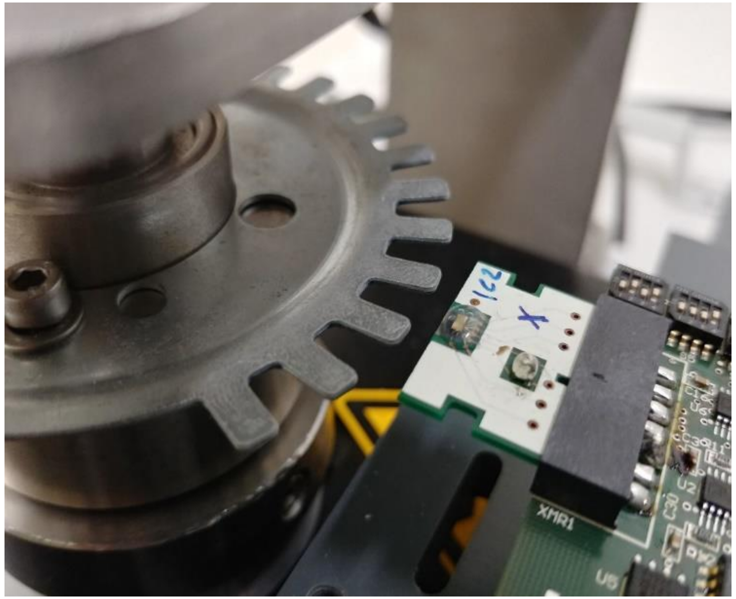
SystemC & VHDL cosimulation (2/2)



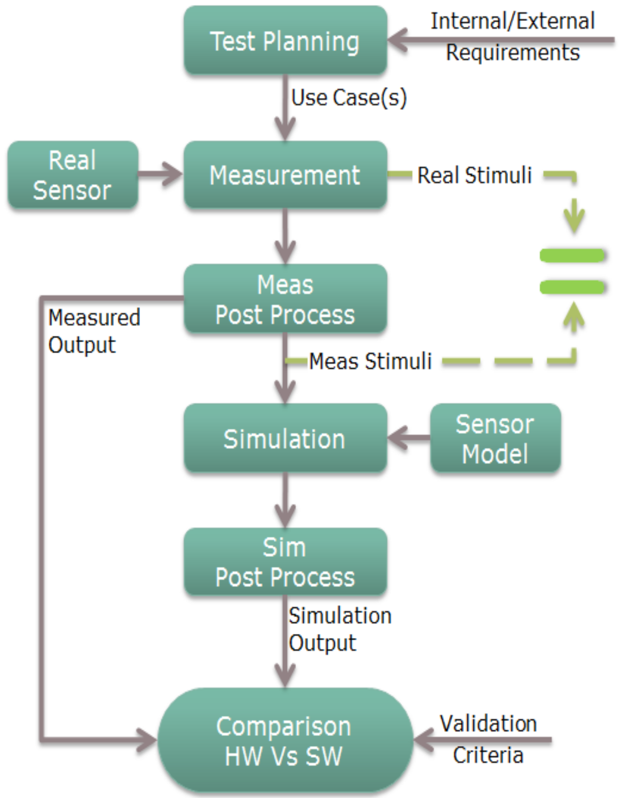
Moving to real HW

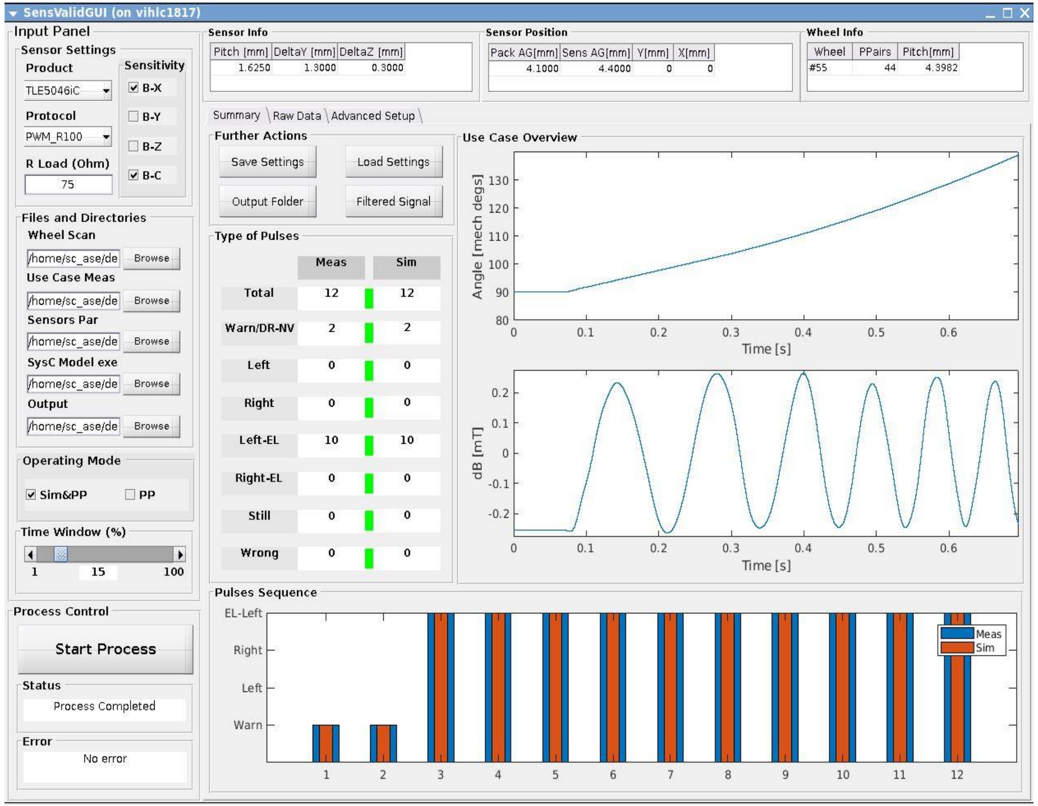


Measurements setup in the laboratory

Automatic measurement & simulations crosscheck




SensValidGUI (on vihlc1817)

Input Panel

Sensor Settings

Product: TLE5046iC

Protocol: PWM_R100

R Load (Ohm): 75

Sensitivity: [✓ B-X] B-Y B-Z [✓ B-C]

Files and Directories

Wheel Scan: /home/sc_ase/de Browse

Use Case Meas: /home/sc_ase/de Browse

Sensors Par: /home/sc_ase/de Browse

SysC Model exe: /home/sc_ase/de Browse

Output: /home/sc_ase/de Browse

Operating Mode: [✓ Sim&PP] PP

Time Window (%): 1 15 100

Process Control

Start Process

Status: Process Completed

Error: No error

Sensor Info

Pitch [mm]	DeltaY [mm]	DeltaZ [mm]
1.6250	1.3000	0.3000

Sensor Position

Pack AG[mm]	Sens AG[mm]	Y[mm]	X[mm]
4.1000	4.4000	0	0

Wheel Info

Wheel	PPairs	Pitch[mm]
#55	44	4.3982

Summary \ Raw Data \ Advanced Setup

Further Actions: Save Settings, Load Settings, Output Folder, Filtered Signal

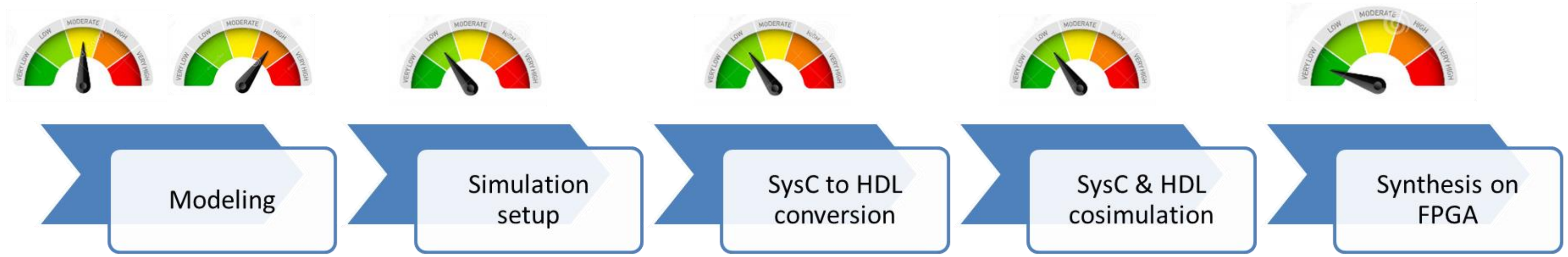
Type of Pulses

	Meas		Sim
Total	12	🟩	12
Warn/DR-NV	2	🟩	2
Left	0	🟩	0
Right	0	🟩	0
Left-EL	10	🟩	10
Right-EL	0	🟩	0
Still	0	🟩	0
Wrong	0	🟩	0

Agenda

Results and Conclusion (1/3)

How much effort was spent for the different steps in the flow?



- Modeling: 1 month if concept available (SysC reuse) / up to 1 year if concept has to be developed
- Simulation setup: straightforward, just the parameter sweeps and their steps have to be defined
- SysC to HDL conversion: achieved with a one-click approach using Mentor Catapult software
- SysC & HDL cosimulation: made possible by Coseda-Cadence-Bridge (CCB) with one click export
- Synthesis on FPGA possible without any need of modifications, using Xilinx ISE synthesizer

Results and Conclusion (2/3)

- How much time was saved by this methodology?
  - From virtual to real HW prototype: 3 to 6 man / months faster!

- What is the simulation speed of SysC vs. Matlab vs. SysC/HDL co-sim?
  - SystemC : 1ms of simulation → ca. 5 s in the real world
  - Matlab: does not affect the simulation speed, only used to handle the regression
  - SystemC / HDL co-simulation: around 6 times slower than SystemC due to RTL simulation time (dominant)

Results and Conclusion (3/3)

- One-click conversion finally possible
- HDL and SystemC match 1:1 in cosimulation
- Measurements ongoing, correct functionality already observed

- High level synthesis approach
  - Saves development resources and time
  - Increase reuse and speed
- Rapid prototyping approach
  - Increase design confidence
  - Allow better customers interaction

Questions & Answers

› Any questions?

Part of your life. Part of tomorrow.