SYSTEMC AMS PLATFORM FOR RADAR: THE RF BIST

SYSTEM VALIDATION CONCEPT WITH COSIDE®

Dominique Delbecq (NXP), Michel Vasilevski (Seamless Waves)

OCTOBER, 2020

COSEDA usergroup meeting | 2020

SECURE CONNECTIONS FOR A SMARTER WORLD

Seamless Waves Semiconductors

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

OUTLINE

- System C-ams platform for RADAR transceiver system exploration : reason for the choice of this platform
- Description of the pilot : BIST for RADAR
- Development of the models
- An example of a 77GHz simulation with ARM control
- Conclusion and some comparison

SystemC platform for RADAR system exploration

SECURE CONNECTIONS
FOR A SMARTER WORLD

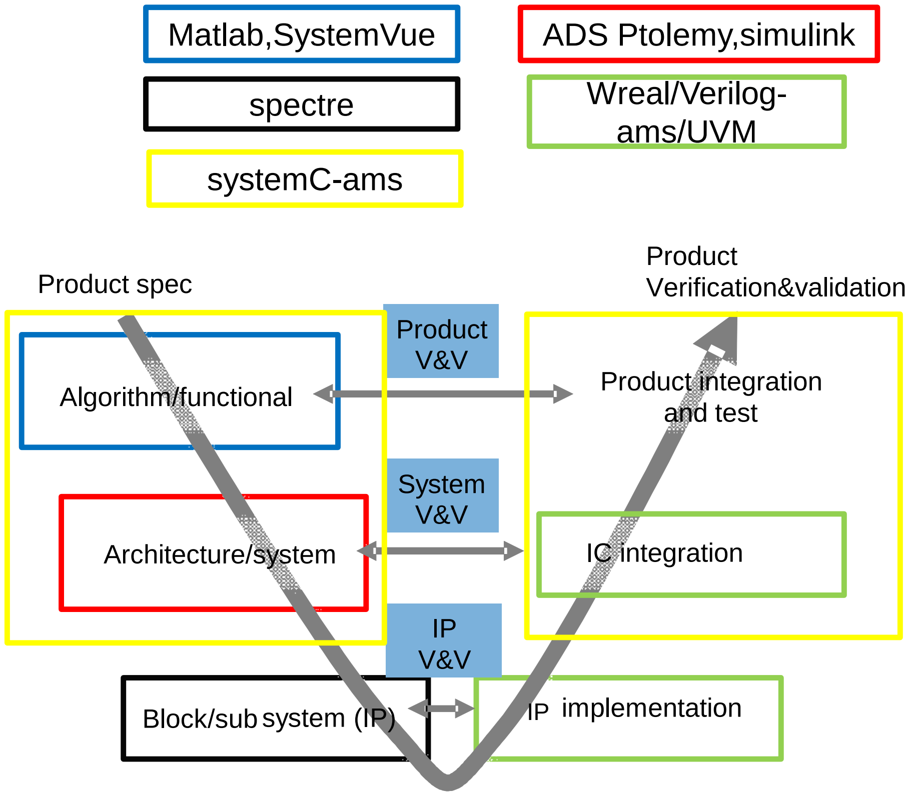
Seamless Waves
Semiconductors

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

SYSTEMC-AMS PLATFORM TO MANAGE COMPLEXITY

- **Automotive RADAR market is requiring more complexity (imaging radar) and more integration (cost)**
- A performant radar system is a smart combination of **heterogeneous** systems : PCB/antenna, MMW, RF, analog, digital, processing, software
- **Smart Partitioning (analog=>digital, hardware => software)** allows area reduction and flexibility (**software Design RADAR**). However the **system exploration** is difficult to validate by simulation without virtual simulation platform.
- There is today no virtual platform simulation that is capable of supporting a **continuity of check along the V development** (from algorithm to validation)
- Also all the platforms in development are **costly** to support (tool cost, people expertise)
- The serial approach prevent “**parallelization**” of the activities.
- **The virtual platform** should allow to model *RF imperfections*, simulate *control loops* (RF/digital/software) while keeping *simulation time fast*



Matlab,SystemVue

ADS Ptolemy,simulink

spectre

Wreal/Verilog-ams/UVM

systemC-ams

Product spec

Product Verification&validation

Algorithm/functional

Product V&V

Product integration and test

Architecture/system

System V&V

IC integration

IP V&V

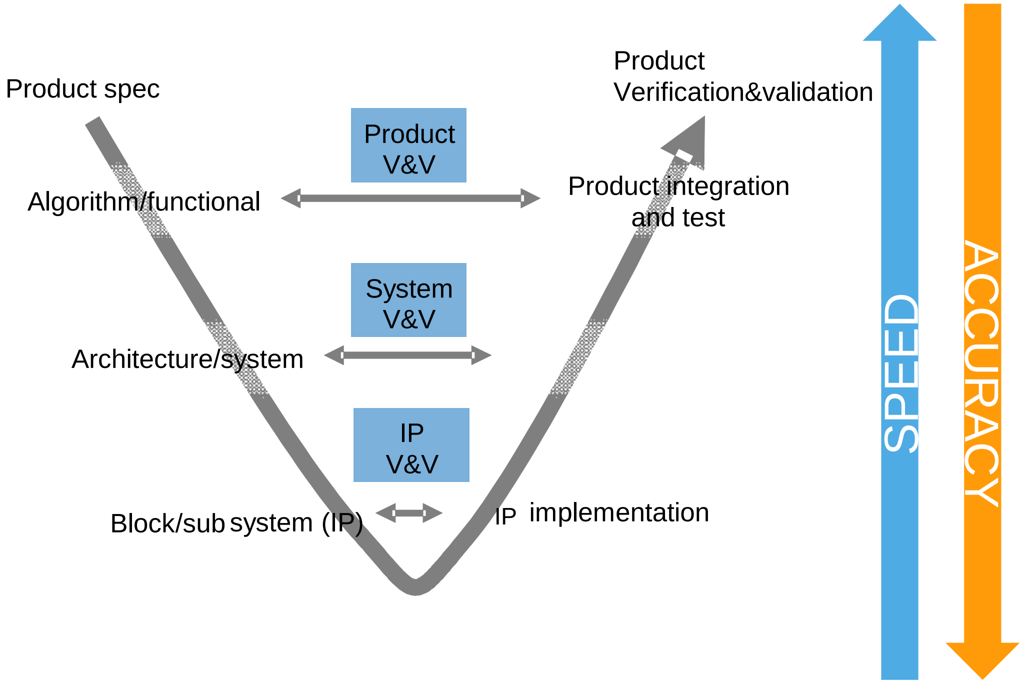
Block/sub system (IP)

IP implementation

WHY SYSTEMC-AMS

SystemC-ams is a language and not a tool that can be integrated in many platform.
Could be the best trade-off accuracy/speed

	SystemC-ams	Other platforms
speed	Compiled language	Simulink, Ptolemy, Verilog-ams are pretty slow simulation
RF modeling and flexibility in analog model	Coside models support envelop type	Proprietary models for ADS
Manage time domain and control loops	Tdf models	Limitation with Matlab
Multi platform	Well suited with software	Heavy Co-simulation
cost	Almost free	Market tools are expensive
Investment	Need investment to develop libraries and competences	Libraries available



RADAR BIST

SECURE CONNECTIONS
FOR A SMARTER WORLD

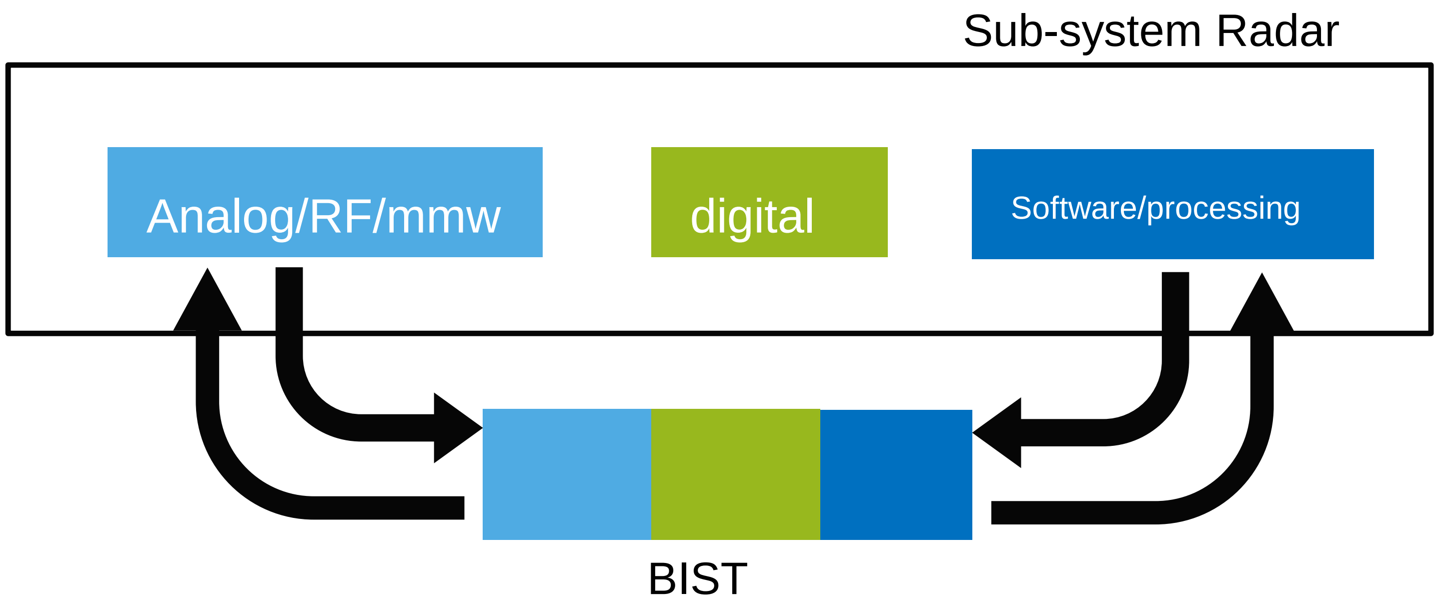
Seamless Waves
Semiconductors

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

BIST IN RADAR TRANSCEIVER

- **BIST (Build-in self test)** corresponds to additional circuitry in a circuit that allows to test functional circuitry (injecting/measuring)
- It is more and more important for :
  - ➢ Reducing production **test cost** by embedding tester on chip
  - ➢ Improving the **safety** coverage (ASIL for automotive) by guaranteeing performance over the life time
  - ➢ Allow continuous **calibration** of analog/RF in advanced cmos design



SYSTEM-AMS WITH BIST

BIST is a good exercise for SystemC-ams platform

This project was a collaboration between :

- NXP
- SeamlessWaves (for models writing)
- Coseda (Coside enhancement)

System simulation experiment

- BIST for RADAR receiver
- BIST for RADAR transmitter
- 77GHz PLL fracN generation controlled by ARM

	SystemC-ams
speed	Yes
RF modeling and flexibility in analog model	RF&MMW Need noise,linearity, impairements …
Manage time domain and control loops	Yes !!
Multi platform	Yes!!
Validation of software	Yes
Complexity of system	Yes

SystemC AMS models

SECURE CONNECTIONS
FOR A SMARTER WORLD

Seamless Waves
Semiconductors

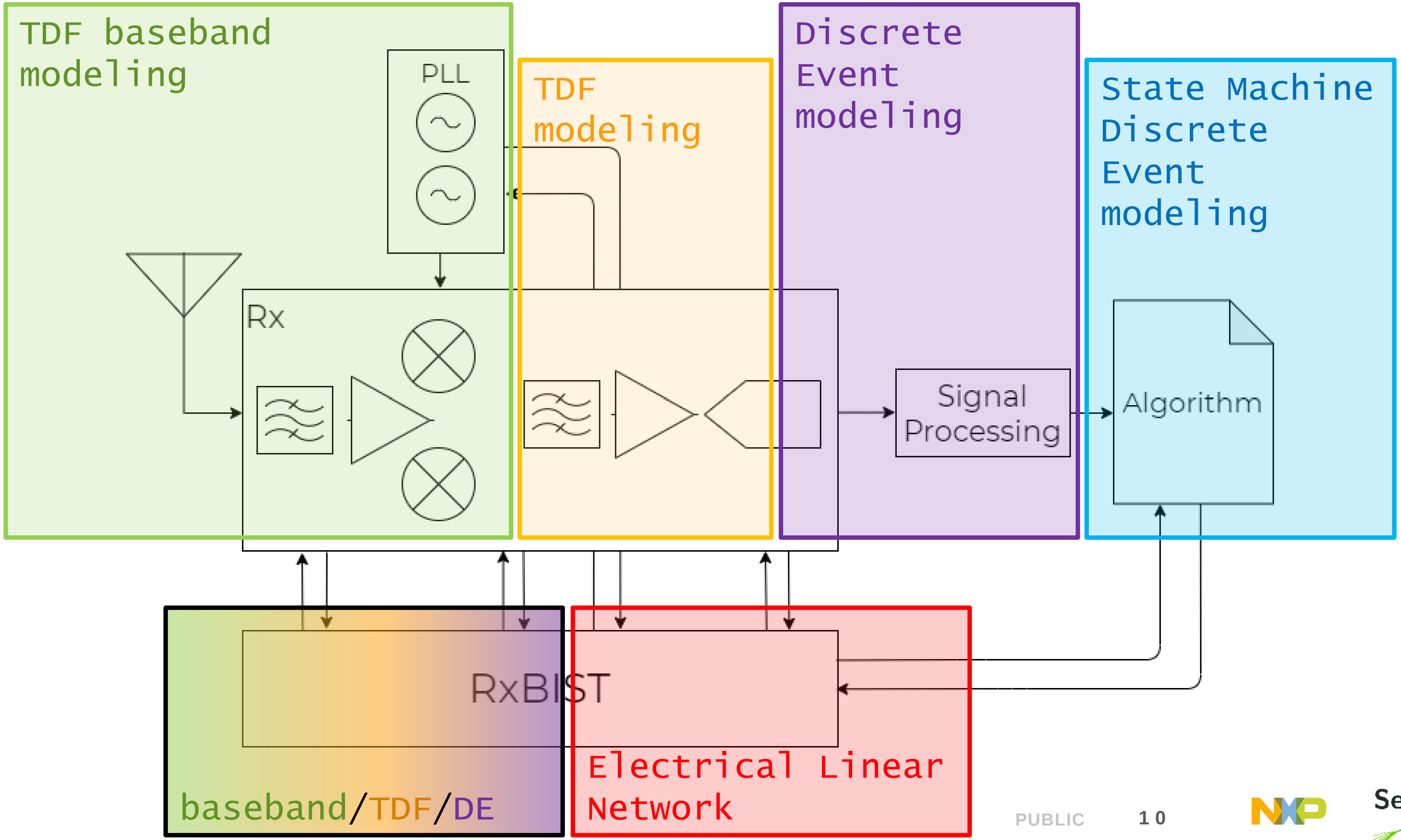
PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

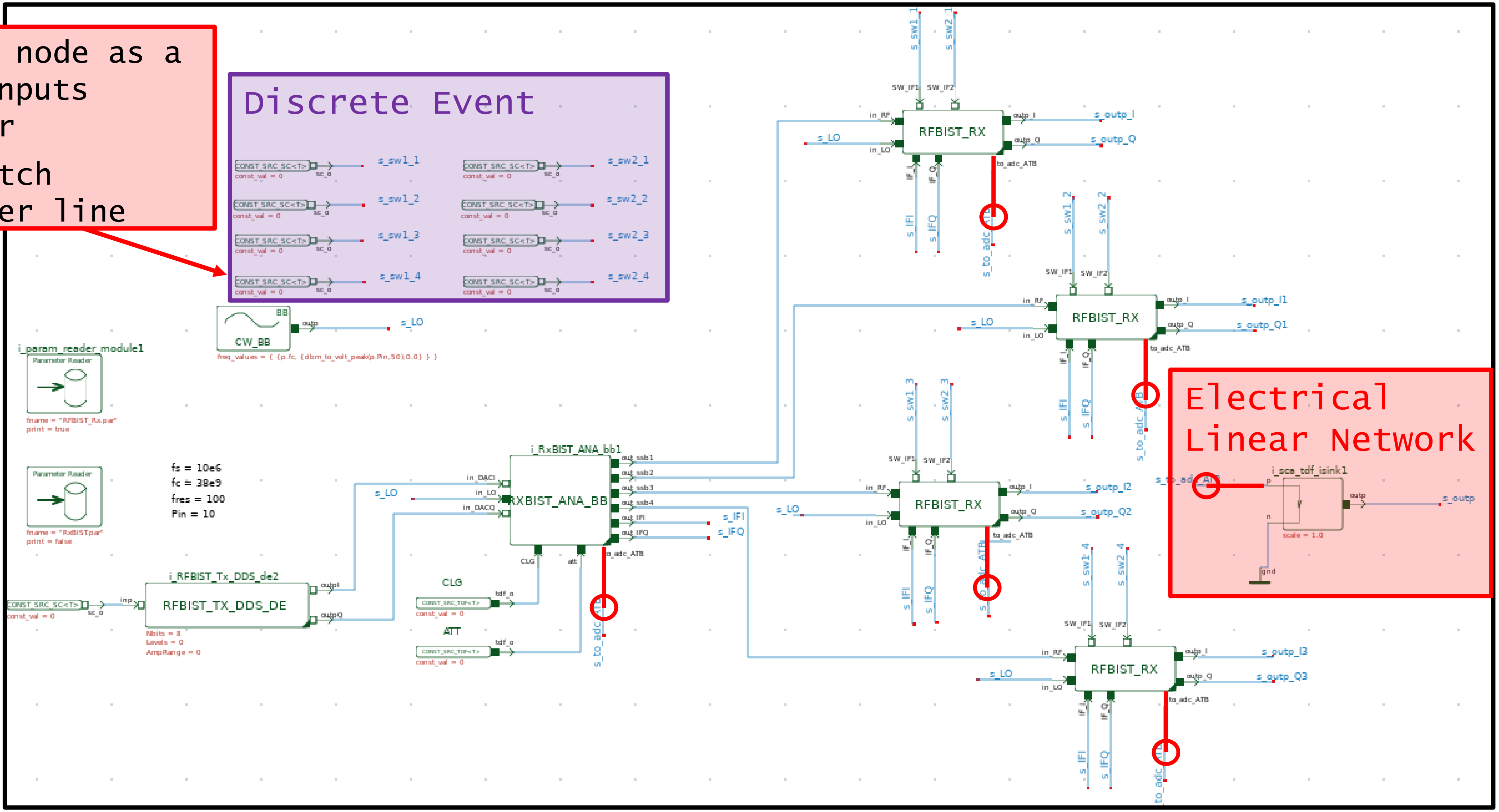
SYSTEMC AMS MODELING WITH COSIDE

- **Heterogeneous modeling**
  - TDF COSIDE baseband : RF signal path
  - TDF standard : IF Analog signal path
  - ELN (Electrical Linear Network) : Sensed Power for BIST purpose
  - Discrete Event : Discrete Tuning signal
  - State Machine Discrete Event : Algorithm
- **Simulation and postprocessing with COSIDE**
  - Embedded COSIDE graphical visualization: time domain and frequency domain
  - Python scripting for specific postprocessing computation
  - Link with Matlab/Octave software
  - Use of parameters configuration file

HETEROGENEOUS MODELING



HETEROGENEOUS MODELING



Use of ELN node as a multiple inputs multiplexer

One DE switch selector per line

Discrete Event

Electrical Linear Network

IMPLEMENTED COMPONENT SPECIFICATIONS

RF Amplifier

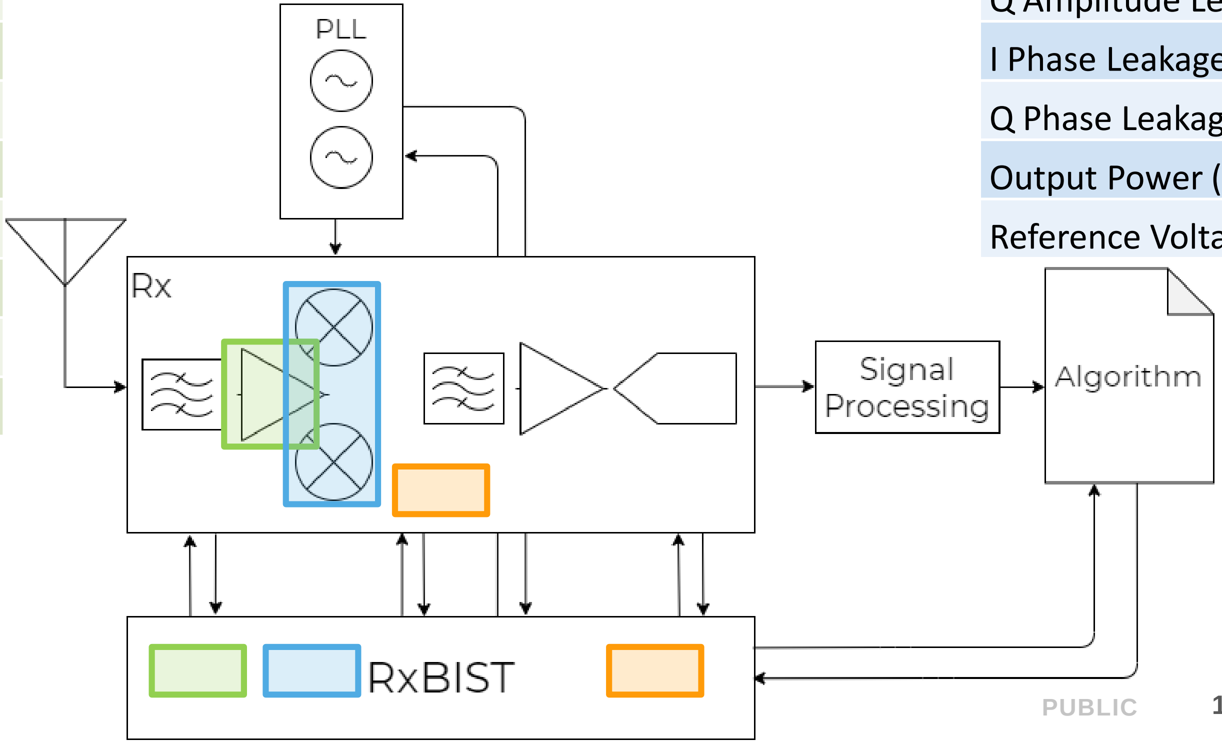
Description
Input Resistance (Ohm)
Output Resistance (Ohm)
Temperature (Kelvin)
Gain (dB)
Saturation Power (dBm)
Gain Compression at Saturation (dB)
Output 1dB Compression Point (dBm)
Noise Figure (dB)
AM Smoothness
PM Transition
PM Smoothness
PM Scaling
AM Noise vs Frequency (Hz,dBc/Hz)
PM Noise vs Frequency (Hz,dBc/Hz)

IQ DAC

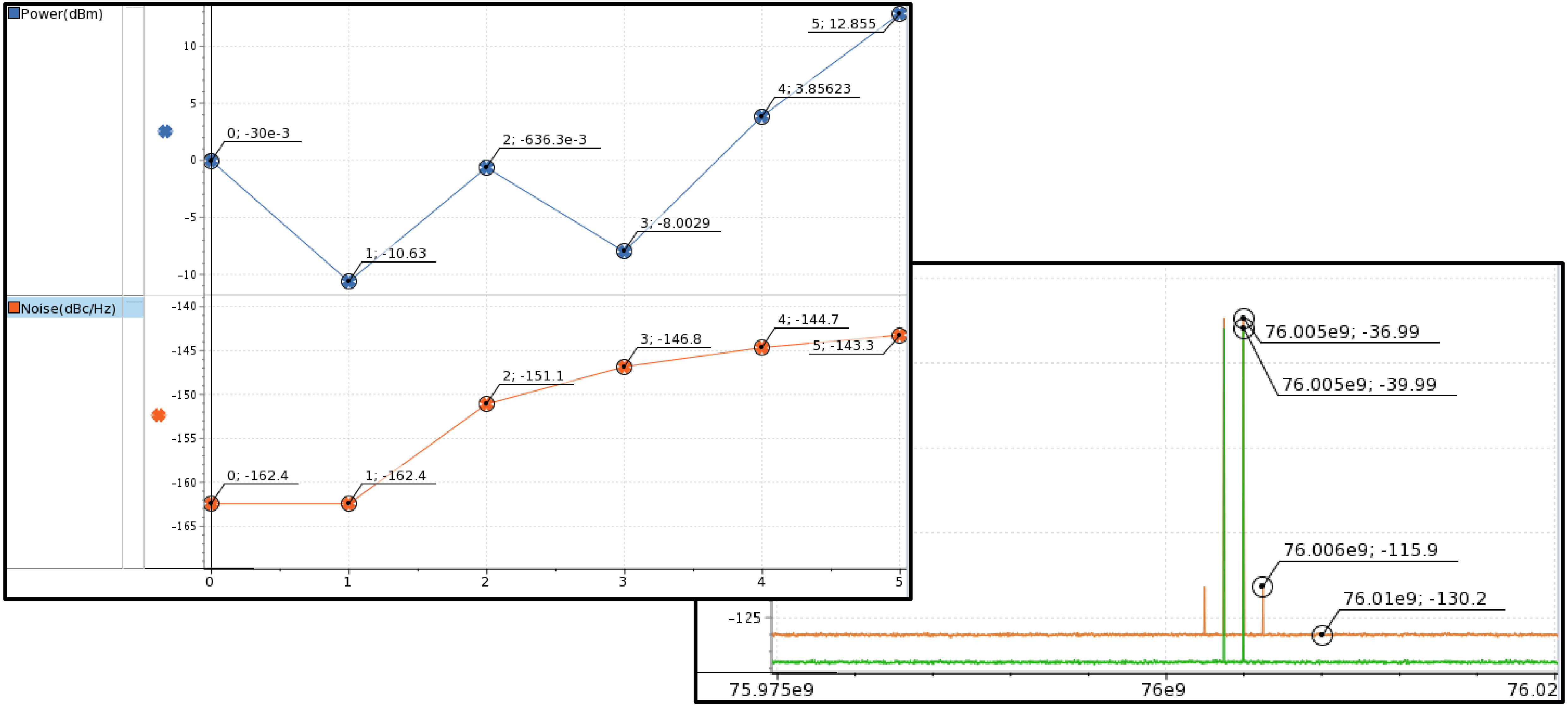
Description
Voltage Reference (V)
Number of Bits
Integral Non-Linearity
Number of Sigma for INL

IQ Modulator

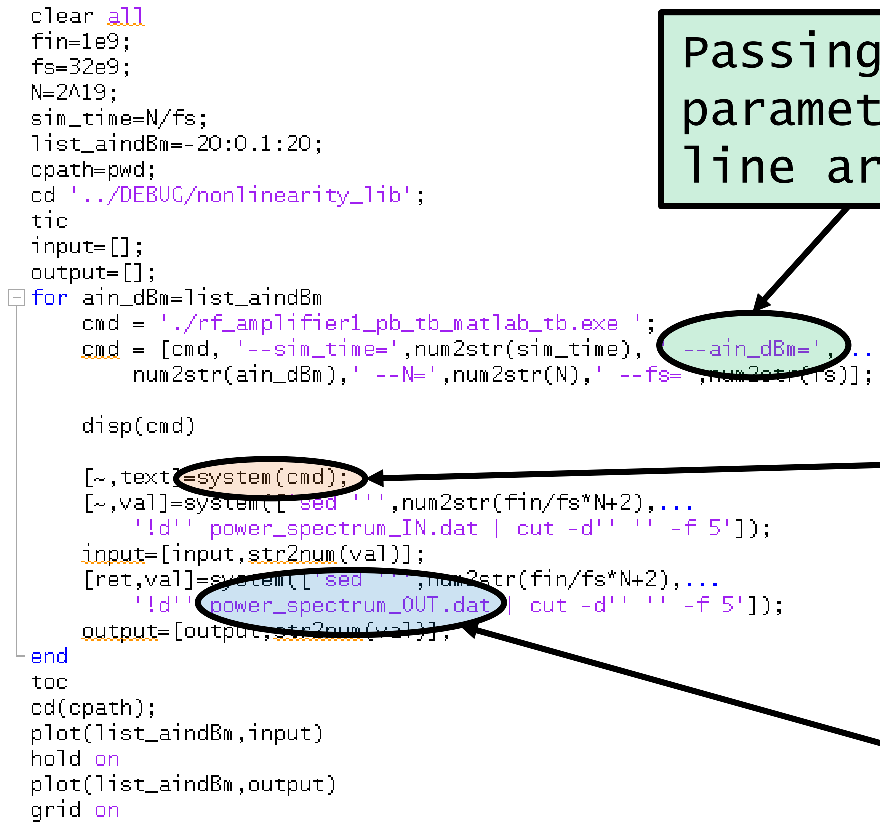
Description
Input Resistance (Ohm)
Output Resistance (Ohm)
Temperature (Kelvin)
I/Q Amplitude Error (dB)
I/Q Phase Error (deg)
I Amplitude Leakage referenced to LO-I (dBc)
Q Amplitude Leakage referenced to LO-Q (dBc)
I Phase Leakage referenced to LO-I (deg)
Q Phase Leakage referenced to LO-Q (deg)
Output Power (dBm)
Reference Voltage (V)



EMBEDDED GRAPHICAL VISUALIZATION

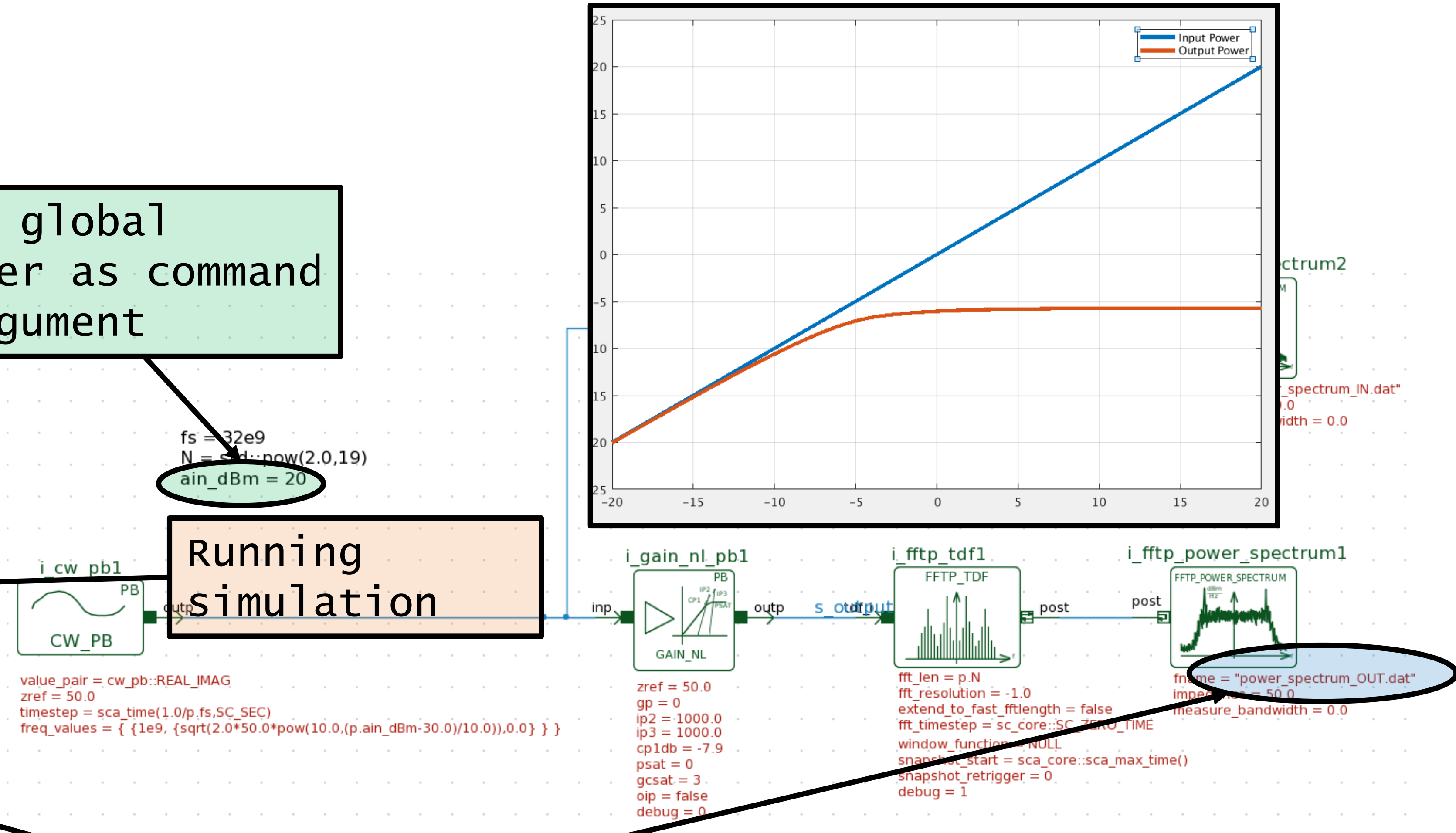


COSIDE SIMULATION — MATLAB POSTPROCESSING



```
clear all
fin=1e9;
fs=32e9;
N=2^19;
sim_time=N/fs;
list_aindBm=-20:0.1:20;
cpath=pwd;
cd '../DEBUG/nonlinearity_lib';
tic
input=[];
output=[];
for ain_dBm=list_aindBm
    cmd = './rf_amplifier1_pb_tb_matlab_tb.exe ';
    cmd = [cmd, '--sim_time=',num2str(sim_time), ' --ain_dBm=',...
        num2str(ain_dBm),' --N=',num2str(N),' --fs=',num2str(fs)];

    disp(cmd)

    [~,text]=system(cmd);
    [~,val]=system(['sed ''',num2str(fin/fs*N+2),...
        '!d'' power_spectrum_IN.dat | cut -d'' '' -f 5']);
    input=[input,str2num(val)];
    [ret,val]=system(['sed ''',num2str(fin/fs*N+2),...
        '!d'' power_spectrum_OUT.dat | cut -d'' '' -f 5']);
    output=[output,str2num(val)];
end
toc
cd(cpath);
plot(list_aindBm,input)
hold on
plot(list_aindBm,output)
grid on
```

Passing global parameter as command line argument



fs = 32e9
N = std::pow(2.0,19)
ain_dBm = 20

value_pair = cw_pb::REAL_IMAG
zref = 50.0
timestep = sca_time(1.0/p.fs,SC_SEC)
freq_values = { {1e9, {sqrt(2.0*50.0*pow(10.0,(p.ain_dBm-30.0)/10.0)),0.0} } }

zref = 50.0
gp = 0
ip2 = 1000.0
ip3 = 1000.0
cp1db = -7.9
psat = 0
gcsat = 3
oip = false
debug = 0

fft_len = p.N
fft_resolution = -1.0
extend_to_fast_fftlength = false
fft_timestep = sc_core::SC_ZERO_TIME
window_function = NULL
snapshot_start = sca_core::sca_max_time()
snapshot_retrigger = 0
debug = 1

fname = "power_spectrum_OUT.dat"
impedance = 50.0
measure_bandwidth = 0.0

Running simulation

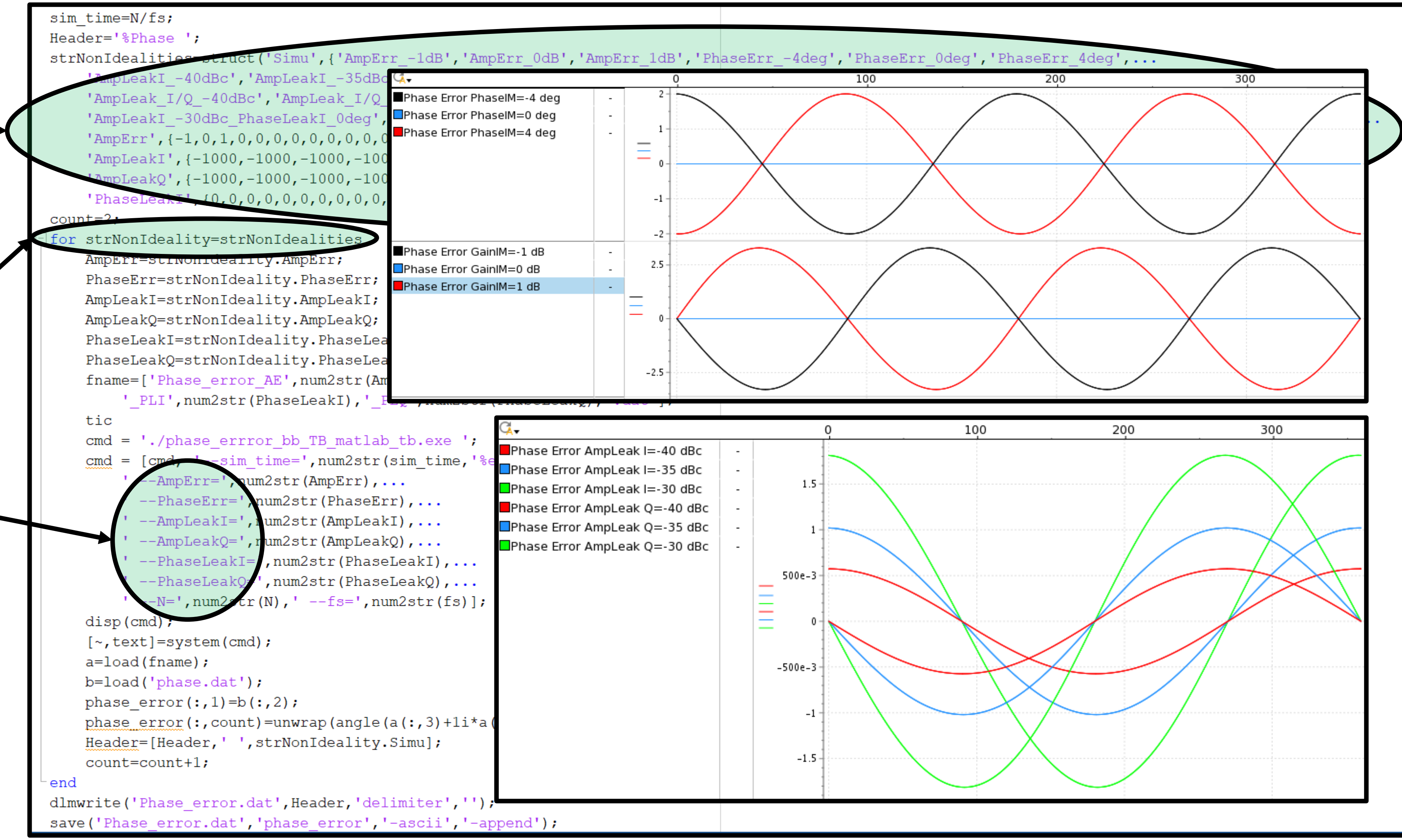
Analyzing generated file

BATCH COMMAND LINE SPECIFICATION EXECUTION

Non-ideality configuration

Loop on each configuration

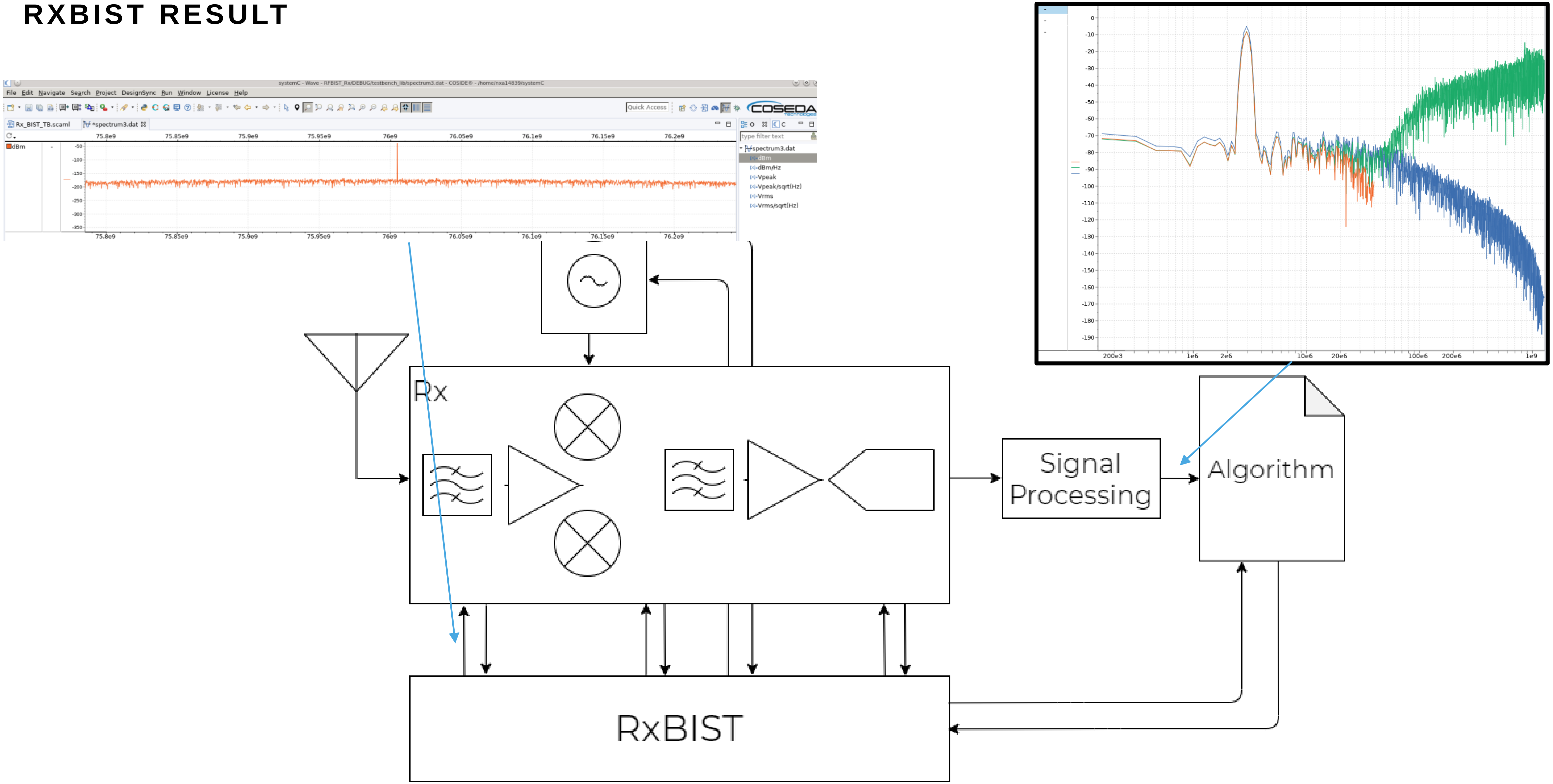
Passing global parameter as command line argument



```
sim_time=N/fs;
Header='%Phase ';
strNonIdealities=struct('Simu',{'AmpErr_-1dB','AmpErr_0dB','AmpErr_1dB','PhaseErr_-4deg','PhaseErr_0deg','PhaseErr_4deg',...
    'AmpLeakI_-40dBc','AmpLeakI_-35dBc
    'AmpLeak_I/Q_-40dBc','AmpLeak_I/Q_
    'AmpLeakI_-30dBc_PhaseLeakI_0deg',
    'AmpErr',{-1,0,1,0,0,0,0,0,0,0,0,0,0
    'AmpLeakI',{-1000,-1000,-1000,-100
    'AmpLeakQ',{-1000,-1000,-1000,-100
    'PhaseLeakI',{0,0,0,0,0,0,0,0,0,0,
count=2;
for strNonIdeality=strNonIdealities
    AmpErr=strNonIdeality.AmpErr;
    PhaseErr=strNonIdeality.PhaseErr;
    AmpLeakI=strNonIdeality.AmpLeakI;
    AmpLeakQ=strNonIdeality.AmpLeakQ;
    PhaseLeakI=strNonIdeality.PhaseLea
    PhaseLeakQ=strNonIdeality.PhaseLea
    fname=['Phase_error_AE',num2str(Am
        '_PLI',num2str(PhaseLeakI),'_P
    tic
    cmd = './phase_errror_bb_TB_matlab_tb.exe ';
    cmd = [cmd, ' --sim_time=',num2str(sim_time,'%e
        ' --AmpErr=',num2str(AmpErr),...
        ' --PhaseErr=',num2str(PhaseErr),...
        ' --AmpLeakI=',num2str(AmpLeakI),...
        ' --AmpLeakQ=',num2str(AmpLeakQ),...
        ' --PhaseLeakI=',num2str(PhaseLeakI),...
        ' --PhaseLeakQ=',num2str(PhaseLeakQ),...
        ' --N=',num2str(N),' --fs=',num2str(fs)];
    disp(cmd);
    [~,text]=system(cmd);
    a=load(fname);
    b=load('phase.dat');
    phase_error(:,1)=b(:,2);
    phase_error(:,count)=unwrap(angle(a(:,3)+1i*a(
    Header=[Header,' ',strNonIdeality.Simu];
    count=count+1;
end
dlmwrite('Phase_error.dat',Header,'delimiter','');
save('Phase_error.dat','phase_error','-ascii','-append');
```

0
100
200
300
🟥 Phase Error PhaseIM=-4 deg
🟦 Phase Error PhaseIM=0 deg
🟥 Phase Error PhaseIM=4 deg
2
1
0
-1
-2
🟥 Phase Error GainIM=-1 dB
🟦 Phase Error GainIM=0 dB
🟥 Phase Error GainIM=1 dB
2.5
0
-2.5
0
100
200
300
🟥 Phase Error AmpLeak I=-40 dBc
🟦 Phase Error AmpLeak I=-35 dBc
🟩 Phase Error AmpLeak I=-30 dBc
🟥 Phase Error AmpLeak Q=-40 dBc
🟦 Phase Error AmpLeak Q=-35 dBc
🟩 Phase Error AmpLeak Q=-30 dBc
1.5
1
500e-3
0
-500e-3
-1
-1.5

RXBIST RESULT



PLL-ARM example

SECURE CONNECTIONS
FOR A SMARTER WORLD

Seamless Waves
Semiconductors

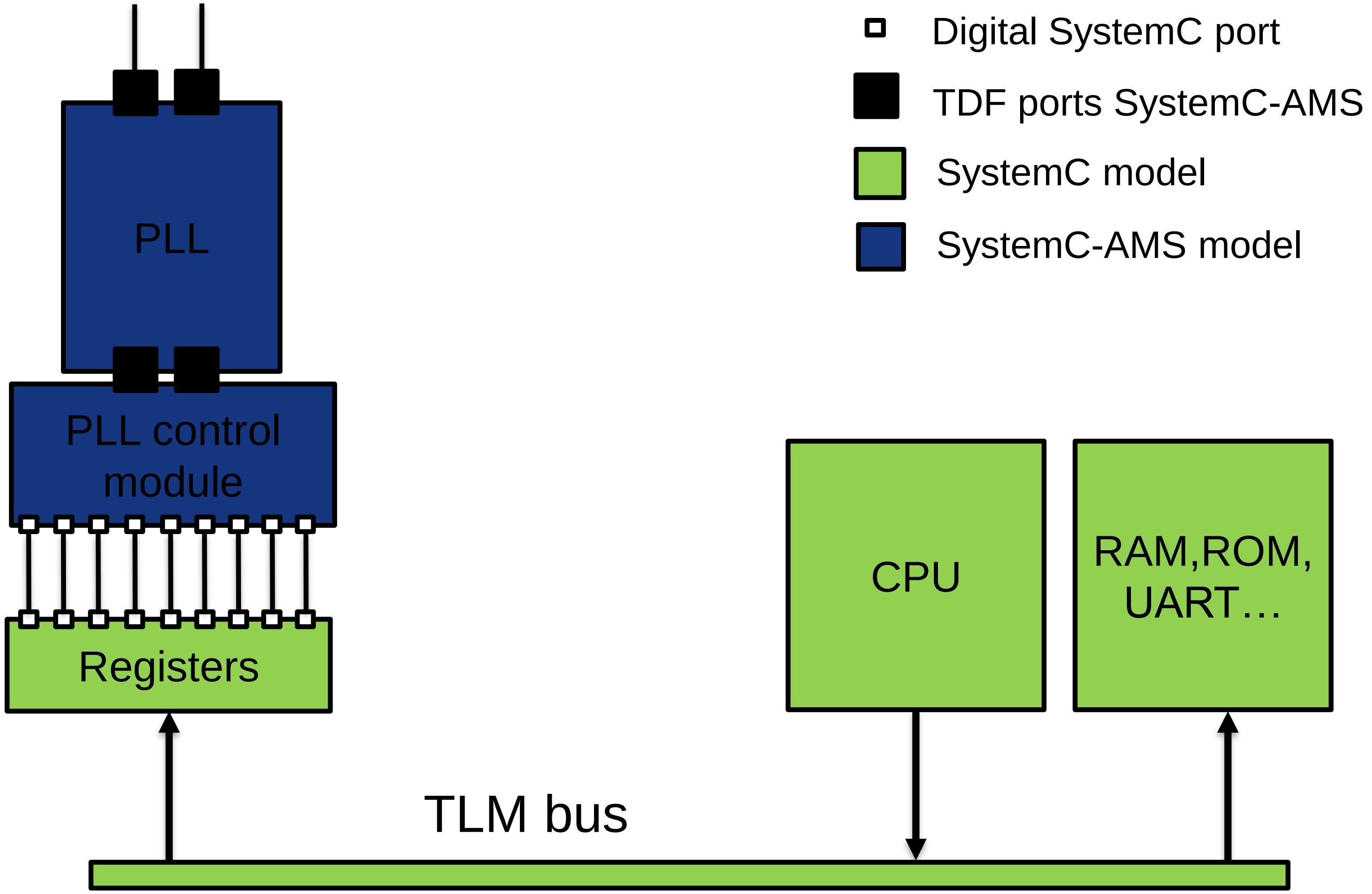
PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

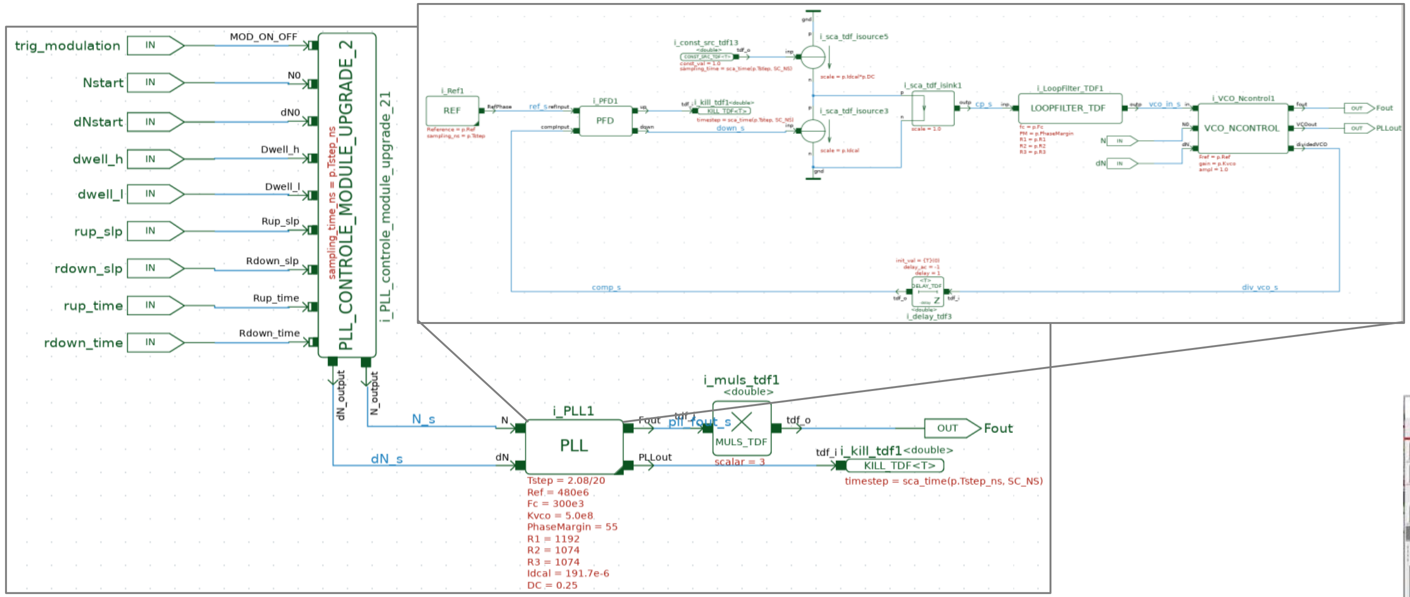
SYSTEMC PLL CONTROL WITH REGISTER INTERFACE

CONTROL OF A PLL CHIRP GENERATION WITH SOFTWARE AND ARM FASTMODEL . INTEGRATION IN CADENCE XCELIUM / SIMVISION

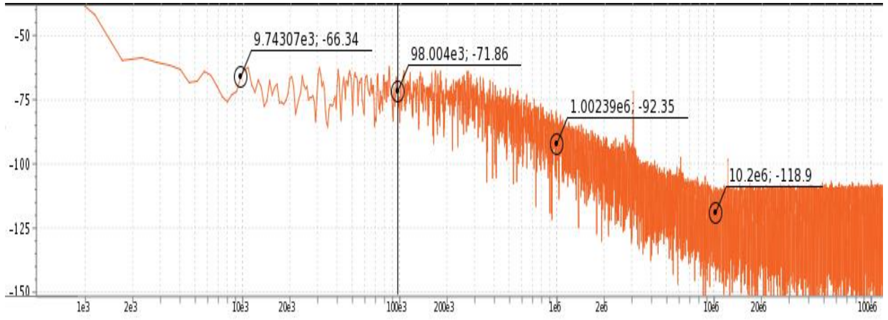
- PLL contains behavior implemented in TDF
- PLL control module : input SystemC ports, output TDF control for PLL module
- CPU subsystem including ARM model and peripherals (e.g. timer, interrupt controller, etc.)
- Bus infrastructure for communication between CPU subsystem and PLL (TLM)



PLL IN RADAR TRANSCEIVER



PLL contains behavior implemented in TDF



PLL phase noise

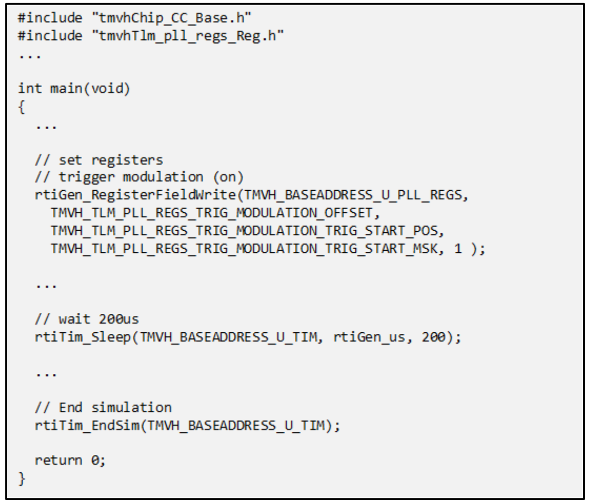
Tools and libraries:

SystemIntegrator / Magillem

ARM Fastmodels

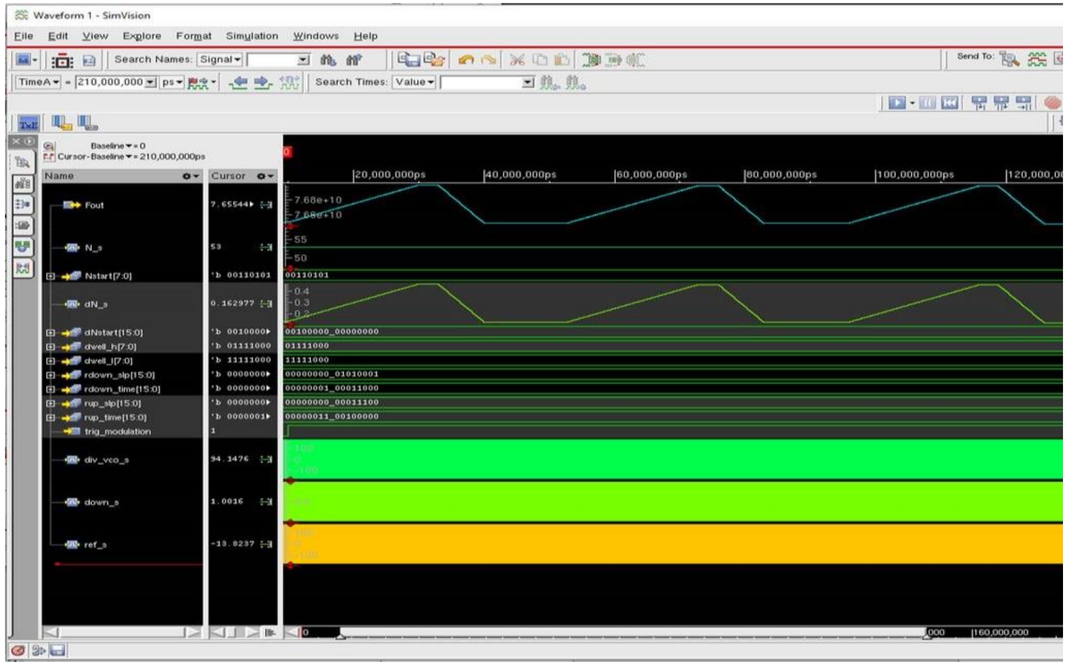
COSIDE

Cadence Xcelium



```
#include "tmvhChip_CC_Base.h"
#include "tmvhTlm_pll_regs_Reg.h"
...

int main(void)
{
  ...

  // set registers
  // trigger modulation (on)
  rtiGen_RegisterFieldWrite(TMVH_BASEADDRESS_U_PLL_REGS,
    TMVH_TLM_PLL_REGS_TRIG_MODULATION_OFFSET,
    TMVH_TLM_PLL_REGS_TRIG_MODULATION_TRIG_START_POS,
    TMVH_TLM_PLL_REGS_TRIG_MODULATION_TRIG_START_MSK, 1 );

  ...

  // wait 200us
  rtiTim_Sleep(TMVH_BASEADDRESS_U_TIM, rtiGen_us, 200);

  ...

  // End simulation
  rtiTim_EndSim(TMVH_BASEADDRESS_U_TIM);

  return 0;
}
```

Control software running on ARM



Visualization of signals with SimVision

NXP

Seamless Waves Semiconductors

Conclusion

SECURE CONNECTIONS
FOR A SMARTER WORLD

Seamless Waves
Semiconductors

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPARISON SYSTEMC/AMS VS ADS PTOLEMY

- Goal : Get a fair time comparison between tools
- Same sampling frequency, same simulation time, same cpu (2 proc xeon 2.4G, 16Gram)

	PLL 35us sim time w/ noise (Fs=10GHz)	RXBIST+RX (1ms of sim) Fs=3GHz
ADS(Ptolemy)	430s	14 min
SystemC (before time optimization*)	119s	4 min

* Further increase in simulation speed expected:

- Sampling frequency and timestep can be optimized along the chain, multi-rate system modeling supported
- COSIDE tool offers "profiling" capability to identify where the computation is high, enabling making local optimizations

CONCLUSION

- It is possible to simulate RF front end model with high accuracy (UPN, non linearity, ADC oversampling, PLL frac-N chirp generations) and validate specifications and architecture concepts
- The loops, calibrations can be easily simulated using a system-level model: BIST processing can be validated before receiving samples
- The COSIDE environment offers a “friendly user interface” for system modeling in SystemC and SystemC-AMS
  - ❖ However a specific expertise is necessary to create your own models in C++ coding + RF system
- The simulation times remain acceptable and models can be reused in different platforms (matlab, Cadence,…)
- Simulation with processor models (e.g. ARM) is possible and allow software validation <u>before</u> receiving silicon
  - Opens the door to extended use of the platform (e.g. pre-validation, pre-verification, …)

NXP

SECURE CONNECTIONS
FOR A SMARTER WORLD

Seamless Waves
Semiconductors

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.