# Co-simulation
# Virtuoso AMS Simulators (Cadence)
# &
# Simulink (Mathworks)
# on real designs

André Baguenier (Cadence), Bertrand Cesbron, Didier Depreeuw

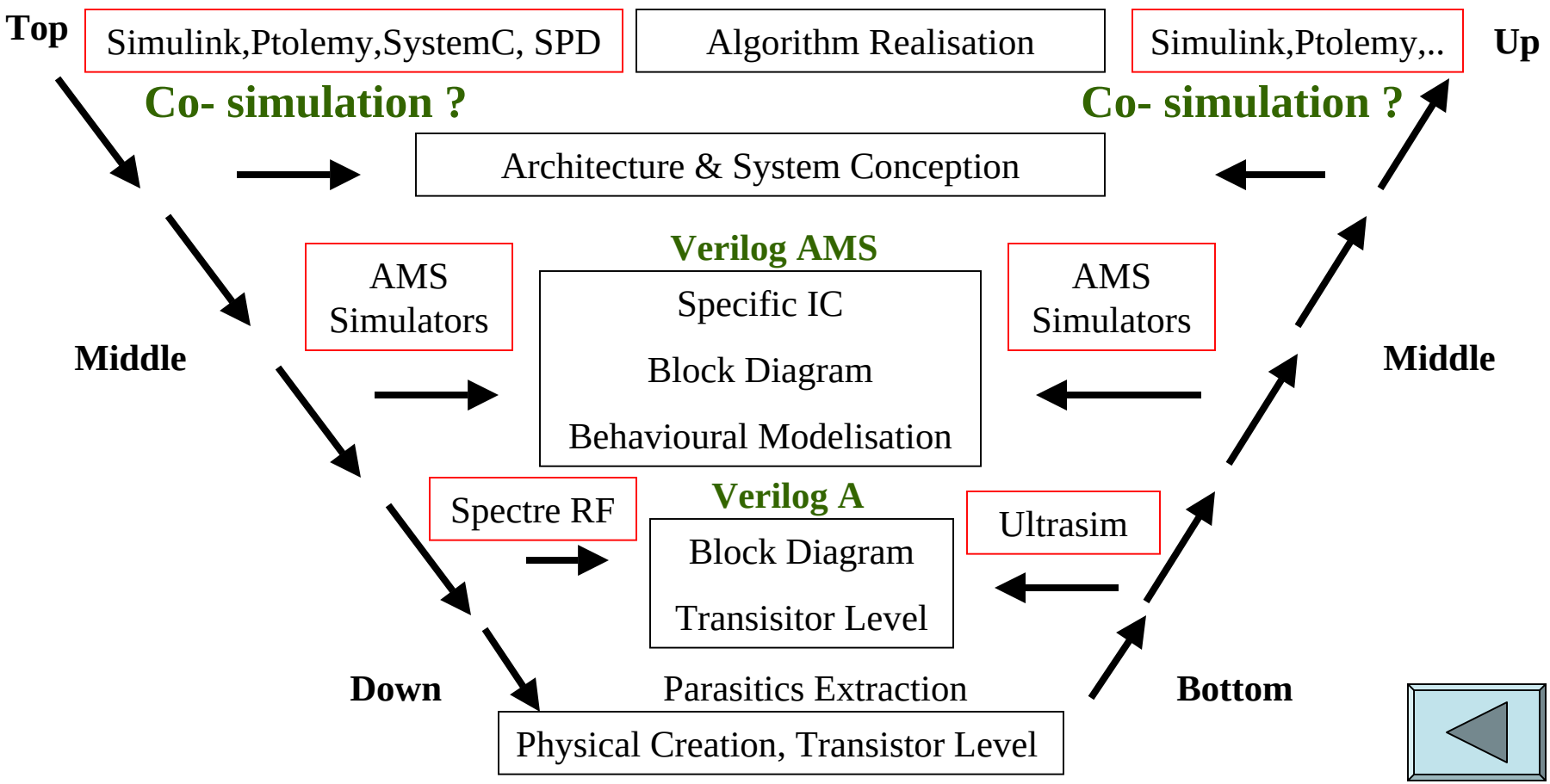
Philips Semiconductors

CDN Live Nice

Monday, June 26, 2006

# AMS-Simulators & Simulink Co-simulation Content

# 1. Objective: Top-Down Methodology, Bottom-up Verification



# 2. BLs Cooperation / Real designs / System Design Methodology

Converter : Philippe Gandy, Christophe Erdmann, Sylvain Dumont:

Multibits Pipeline ADC 14 Bits:
- Electrical / Transistor level : Sample and Hold
- RTL VHDL

Cordless: Fernand Courtois, Fabian Riviere, Tony Vasseaux:

RF Receiver chain:
- Polyphase filter: Behavioural VHDL
- GFSK demodulator, RTL VHDL : “black box”

RF System Design Methodology

# 2. BLs Cooperation / Test Cases / System Design Methodology

ADC 14 Bits: Sample & hold transistor level

ADC 14 Bits: VHDL

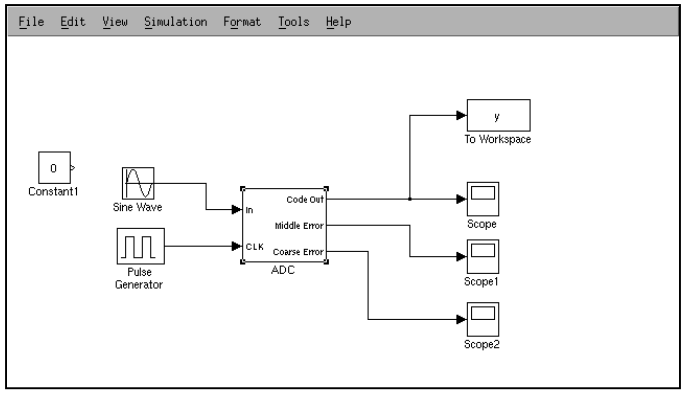
RF Receiver Chain: Polyphase filter: VHDL

RF Receiver Chain: GFSK demodulator: VHDL

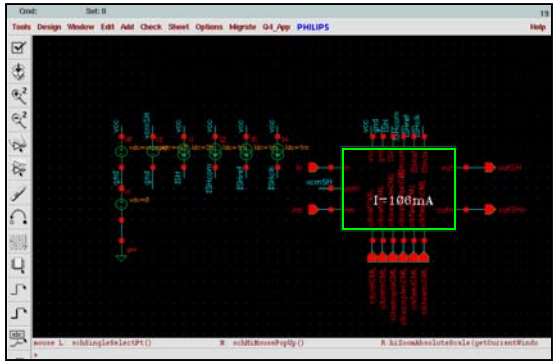
Multi-competence expertise:

- Designers,
- RF System architect
- EDA vendors (Cadence and Matkworks)
- CAD digital engineer
- CTO engineers

# 3. Co-simulation: Sample & Hold transistor level within ADC 14 Bits




ADC 14 Bits System test bench made in Simulink at the beginning of the project:
Top-down methodology
but no co-simulation available
**GAP between system and implementation**

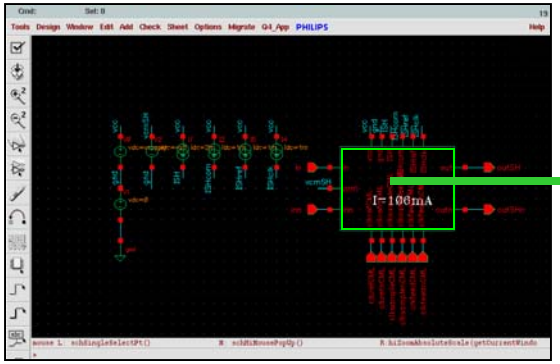



Sample and hold transistor level implemented and verified with Virtuoso Cadence analog simulator: Spectre

Complete implemented ADC 14 Bits verified by Ultrasim before tape-out

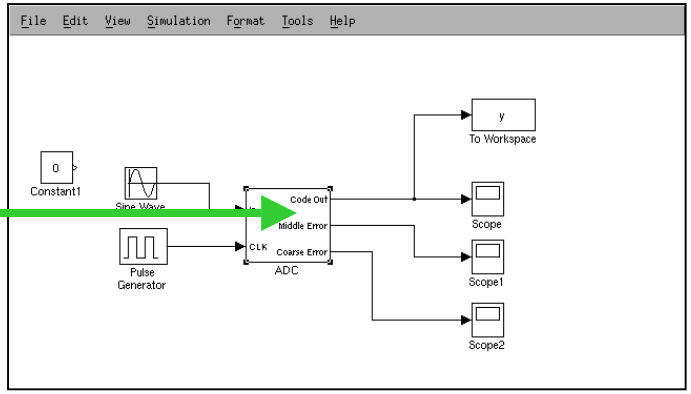
# 3. Co-simulation: Sample & Hold transistor level within ADC 14 Bits

The need is to verify a Sample & hold transistor level within the complete ADC 14 Bits system test bench (Bottom-up Verification)




Sample & Hold
Analog / transistor level
Continuous time
Virtuoso AMS Simulators
with Spectre and UltraSim
Analog solvers
(Cadence)

by
Co-simulation
(VPI)

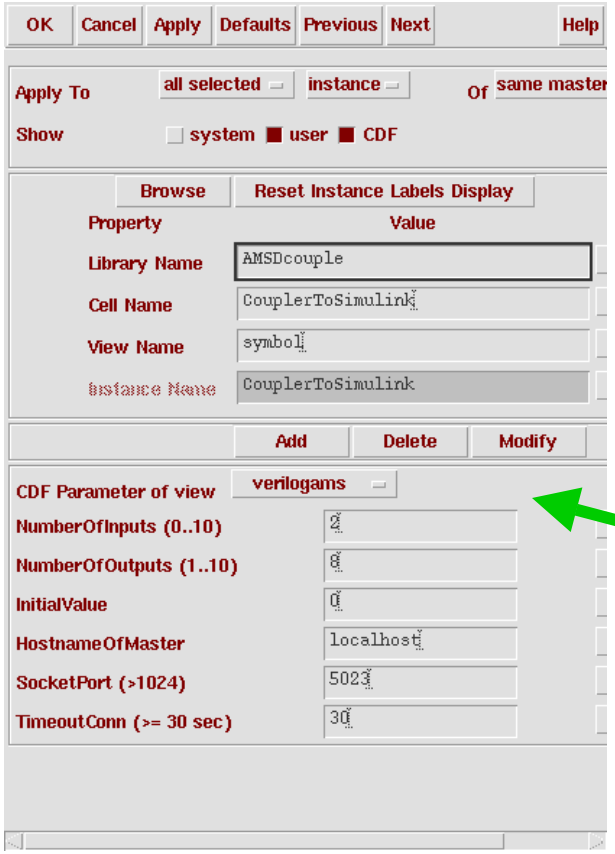


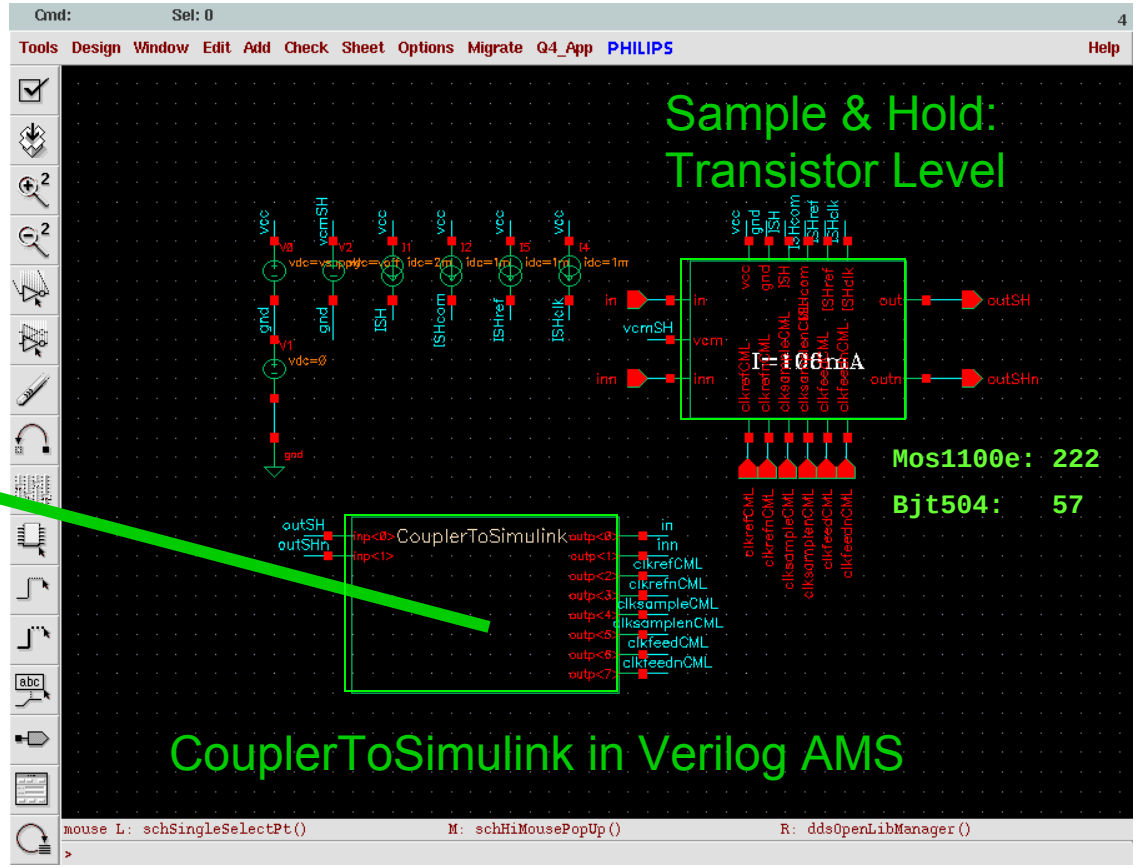

ADC 14 Bits
System test bench
Data Flow
Simulink (Mathworks)

**Close the GAP Between System And Implementation**

# 3. Sample & Hold transistor level within ADC 14 Bits

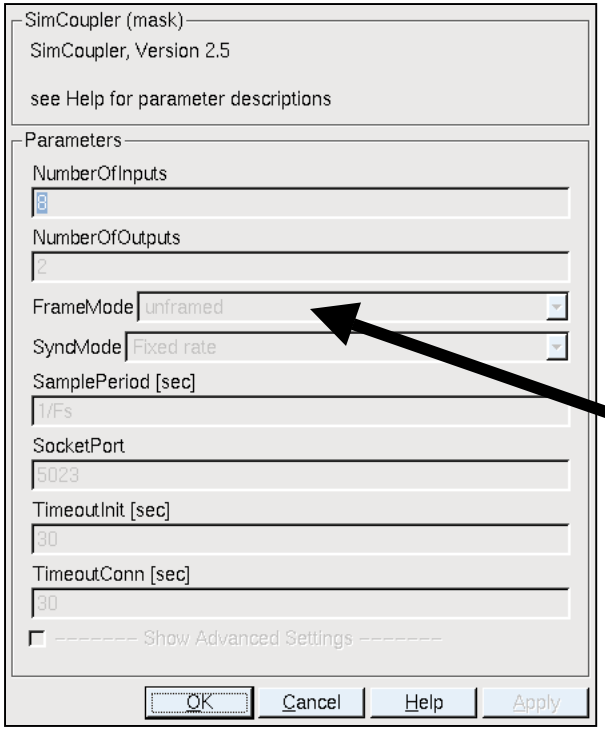
## Cadence Virtuoso: Coupler pcell properties

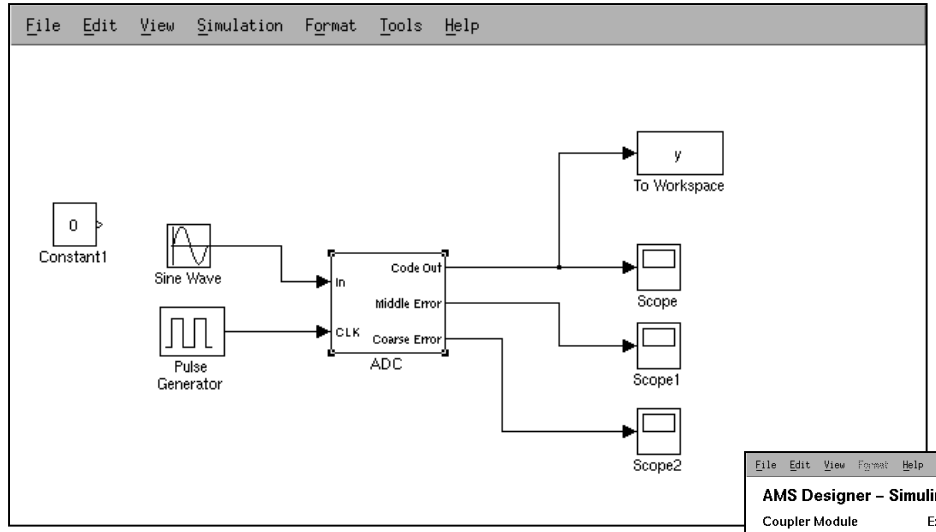


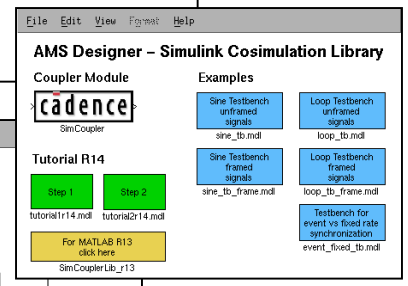



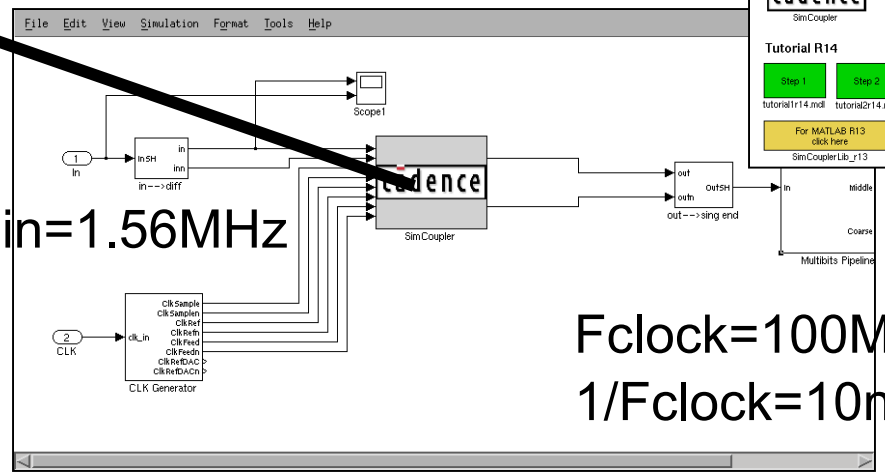
# 3. Sample & Hold transistor level within ADC 14 Bits

## Simulink: Coupler




Fin=1.56MHz

Fclock=100MHz
1/Fclock=10ns

Fs=40*Fclock oversampling
Fs=4GHz, 1/Fs=250ps

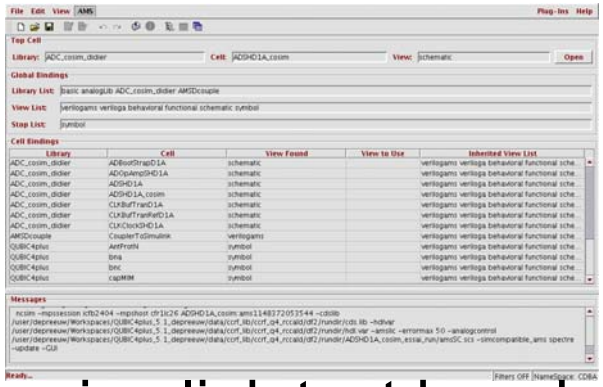
Stop Time = 0.75us

# 3. Sample & Hold transistor level within ADC 14 Bits

## Simulation:

What we made: 2 Xterms on same 32 Bits server and same cadenv: launch icfb& and matlab& on each xterm

Use the Hierarchic Editor or ADE for AMS Simulators, Netlist, Compile, Elaborate and Simulate in order to start the simvision Environment and link to the libvpi.so



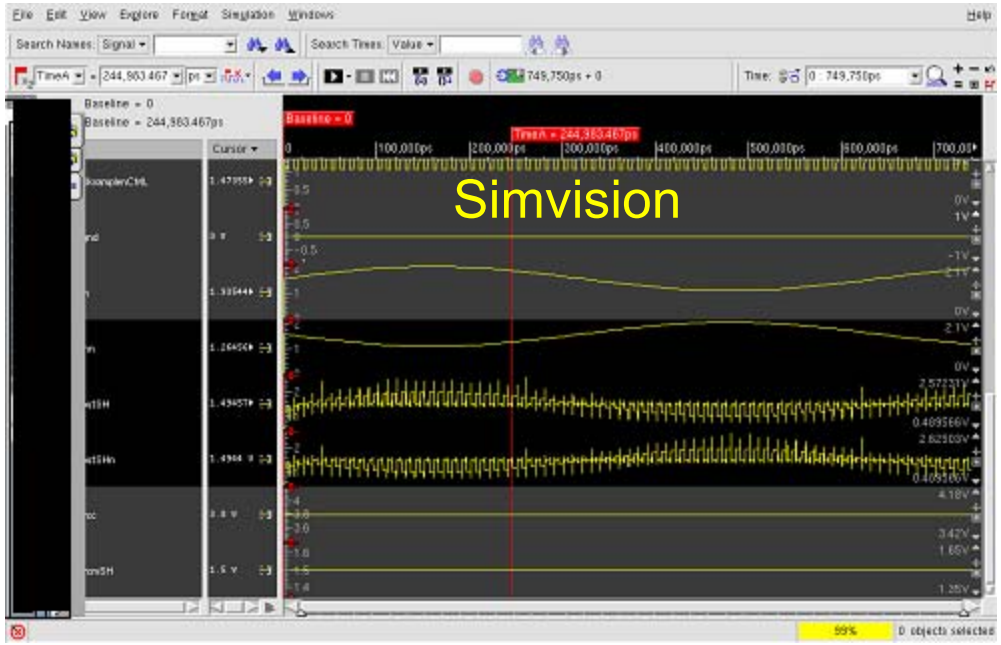
Push Simulation Start of the ADC 14 Bits simulink test bench before the run simulation of the the simvision environment

# 3. Sample & Hold transistor level within ADC 14 Bits

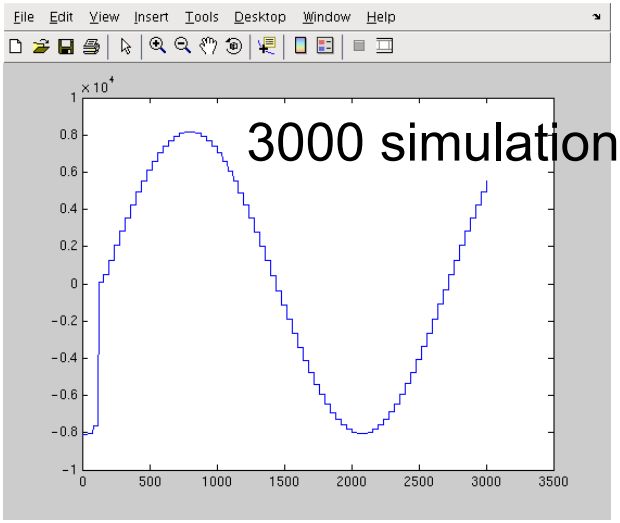
## Simulation time Cadence and Simulink at the same time

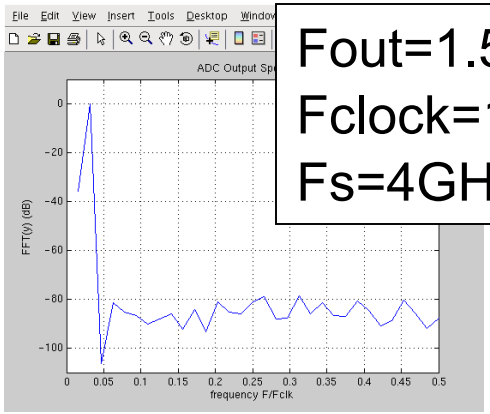
**5 Minutes**

AMS simulator with UltraSim, probing all electrical signals and marching waveform in Simvision environment




ADC 14 Bits output:




Fout=1.56MHz
Fclock=100MHz
Fs=4GHz

# 4. GFSK demodulator

- The need is to verify a GFSK demodulator in RF receiver chain
- GFSK demodulator (VHDL / RTL) made and implemented by a System engineer (not anymore within Philips)
- Script based Compilation, Elaboration and Simulation
- RF architect wanted to make several co-simulations / Loops without “seeing” Cadence Environment for
  **BER simulations as a function of Input Signal Power**

# 4. GFSK demodulator

Custom Coupler Generation

AMS Simulator Testbench

Only one script to launch the complete co-sim : run_flow

SIMULINK : master

master_modulator.m :

1. Matlab variable processing file
2. Generate the tstop.tcl
3. Launch AMS Simulator
4. Start Simulink simulation

AMS Simulator : Slave;

top-demod.tcl (TCL script controlling AMS) source tstop.tcl

to debug the VHDL code if necessary add –GUI instead of exit

Simulink Testbench

# 4. GFSK demodulator: Coupler Generation

Awk script :

input config file = inp_outp.config

output vams file : **coupler_cosim.vams**

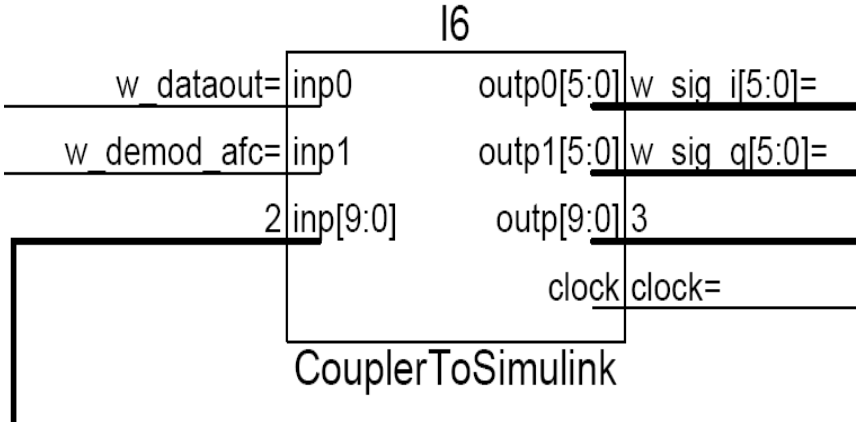
% **build_coupler** inp_outp.config

inp_outp.config example file :

Matlab
real

AMS

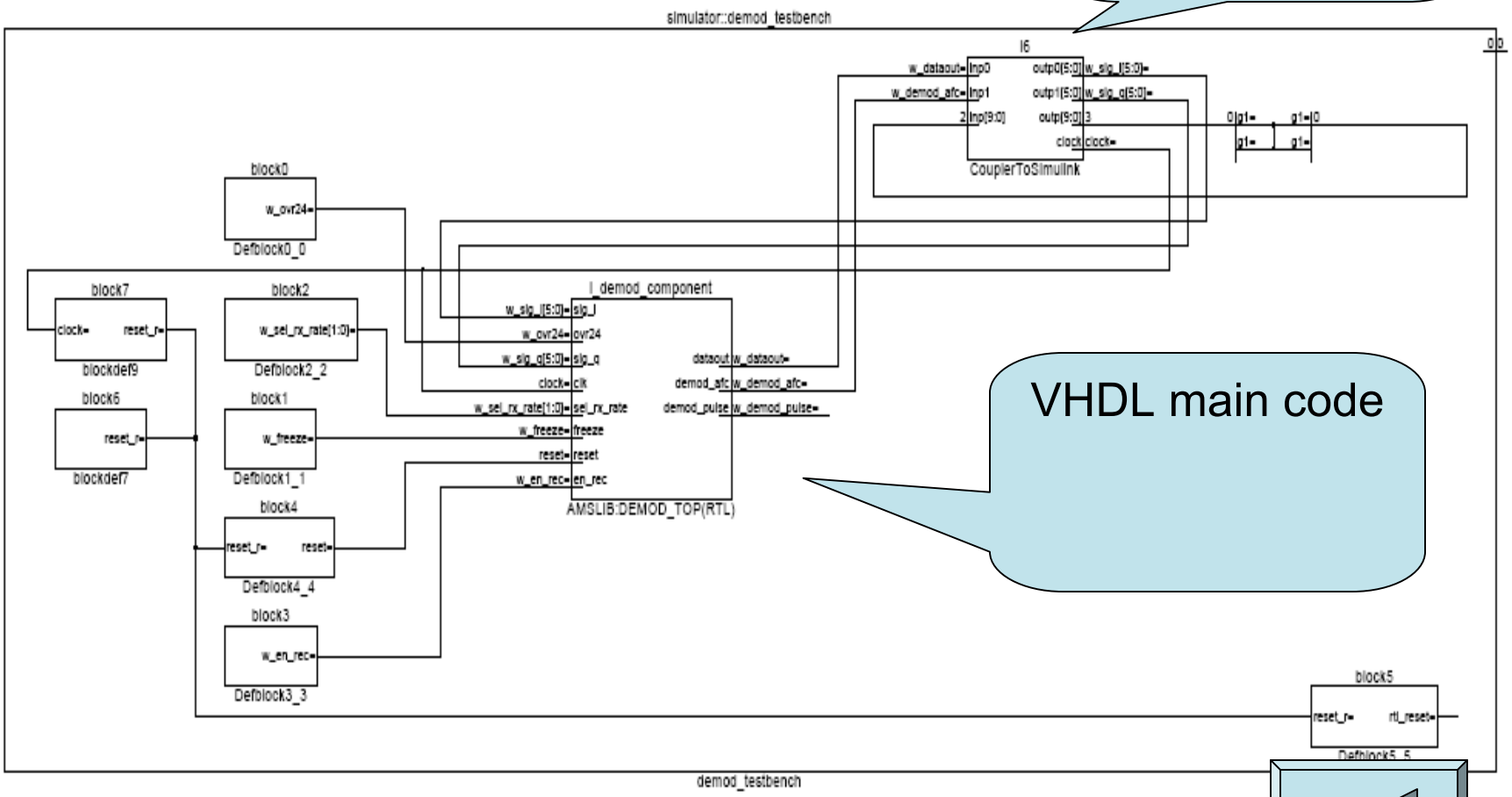
in0 ; [5:0] sig_i
in1 ; [5:0] sig_q
out0 ; dataout
out1 ; demod_afc




For information the data
Passed internally by real numbers

# 4. GFSK demodulator: AMS-Simulators test bench

Coupler created With the build_coupler command




VHDL main code

# 4. GFSK demodulator: run_flow script

```
#!/bin/ksh
#==============================================================================
#                COPYRIGHT (c) Philips 2005
#       All rights are reserved. Reproduction in whole or in part is
#       prohibited without the written consent of the copyright owner.
#------------------------------------------------------------------------
#
#   Author      : B.Cesbron
#   Rel/Date    : 1.0 - 05/01/2006
#   Purpose     :
#==============================================================================
if [ "$DEBUG" != "" ]; then set -x; fi
PROGNAME=`basename $0`

if [ "$LSB_JOBID" = "" -o "`arch`" != "i686"  ]; then
 exec bsh -q linux -R rh3ws32 $PROGNAME "$@"
else

matlab -r master_modulator &
 fi
```

Select the right CPU
Starts Matlab
Load the right mdl files and script

# 4. GFSK demodulator: Examples of Mathworks script : Master_modulator.m

```
clear all; close all;clc;

Upsampler=12;          % oversampling
f_symb=1152e3;         % clock reference frequency
T_symb=1/f_symb;       % symbol period
Tstep=T_symb/Upsampler;        % Time step

Frf=864e3;
Ntot=2^20;             % samples number
S=1/(4*T_symb);        % Sensibility

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Filter characteristic
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load coeff.txt                 % coefficient of gaussian filter
coefficient=(coeff(1:41))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
open_system('modulator_cosim.mdl');
for i=1 : 2
        Tstop=Ntot*Tstep; %
        Tstop=Tstop/i;
        fid = fopen('./tstop.tcl','w');
        count = fprintf(fid, 'set tstop %es',Tstop);
        fclose(fid);

        ! run_demod > demod.log &
        sim('modulator_cosim');
end
```

```
%trace=outputRF;
%N=length(trace); %length(i);
%Ts=Tstep;
%Nfft=length(trace);
%fq=(1:Nfft/2)/(Nfft*Ts);
%fenetre=blackman(Nfft);
%yfen=trace.*fenetre;
%yfft=fft(yfen);
%Py=yfft.*conj(yfft);
%Puissance=10*log10(Py);
%figure(1)
%plot(fq,Puissance(1:Nfft/2));
%xlabel('Frequency (Hz)');
%ylabel('Power spectral density (dB)');
%grid;
```

Generate a TCL command file with the same tstop for AMS

Start AMS compilation elaboration & simulation

Start Simulink cosim

# 4. GFSK demodulator: Examples of AMS Simulator script run_demod

```
#!/bin/csh -f
clear

ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_demod_pkg.p.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_shift.e.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_shift_rtl.a.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_biqu.e.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_biqu_rtl.a.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_lowpass.e.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_lowpass_rtl.a.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_rotate.e.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_rotate_rtl.a.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_ieq.e.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_ieq_rtl.a.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_r_pll.e.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_r_pll_rtl.a.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_demod_iq.e.vhdl
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/bmp_demod_iq_rtl.a.vhdl

ncvhdl -update   ../../RTL_DEMOD/RTL_test/clock_generator.a.vhd
ncvhdl -update   ../../RTL_DEMOD/RTL_test/reset_generator.a.vhd
ncvhdl -update   ../../RTL_DEMOD/RTL_Vone/RTL211005/demod_top.vhd
#
#
ncvlog  -update -work amslib -ams -message coupler_demod_cosim.vams

ncvlog -update -work amslib -ams -message demod_testbench.vams

#Elaboration

ncelab -update -ACCESS +rwc demod_testbench -TIMESCALE "1s / 1s" -work amslib -MESSAGES \
   -loadvpi $CADENV_HOME/.caddata/matlab_coupling_module/lib/vpi/libvpi:register_my_systfs

#Launch simulation
ncsim demod_testbench  -messages -logfile \
    -amslic -analogcontrol top_demod_cosim.scs -input top_demod.tcl
```

No –linedebug in hdl.var or ncvlog or ncvhdl to decrease the run_time

| | |
|---|---|
| Components: | 14 |
| Default bindings: | 13 |
| Processes: | 164 |
| Signals: | 222 |

VHDL Design hierarchy:

| | |
|---|---|
| Components: | 14 |
| Default bindings: | 13 |
| Processes: | 164 |
| Signals: | 222 |

# Examples of AMS Simulator script top-demod.tcl

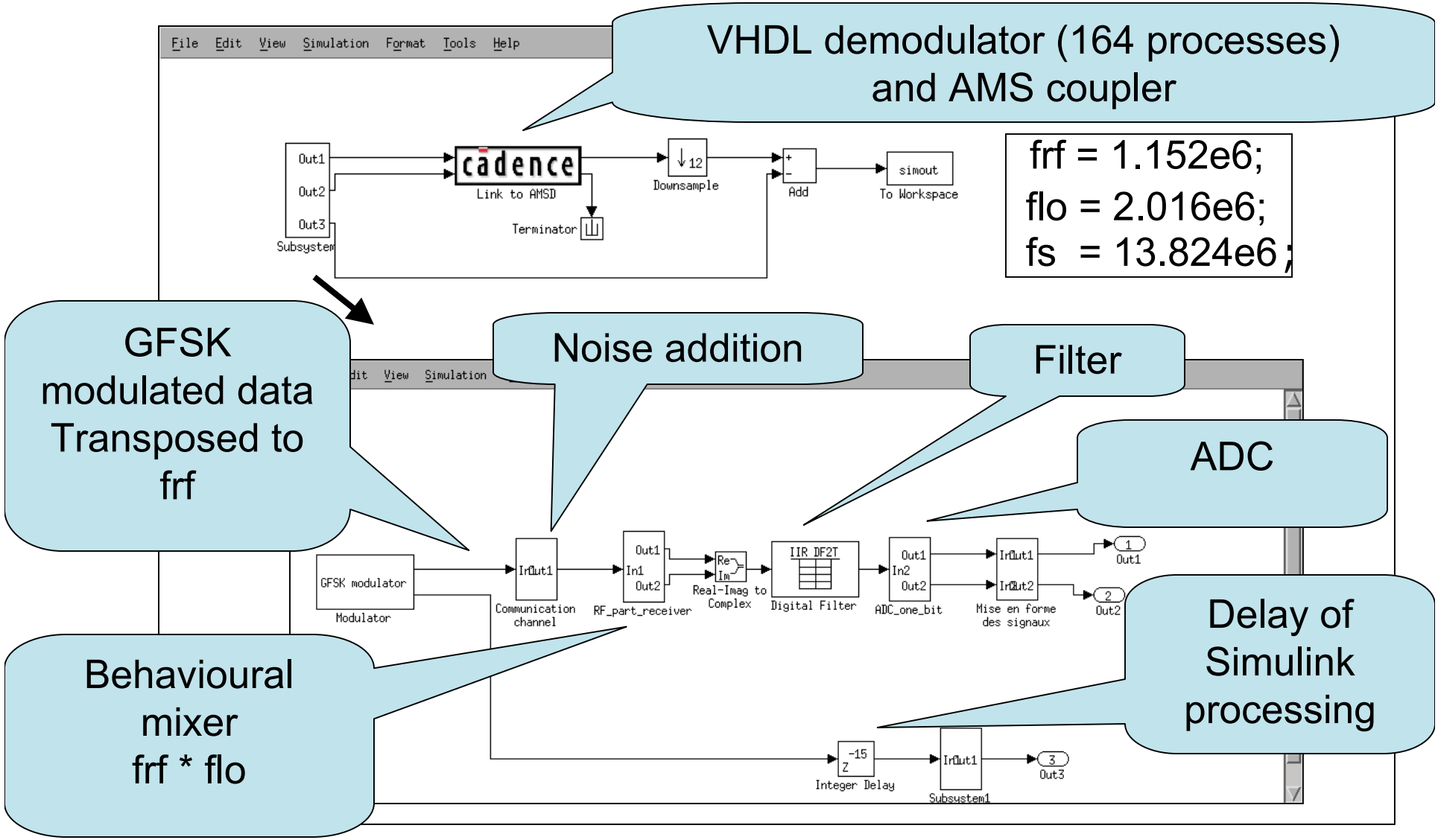
```
source tstop.tcl
run $tstop
exit
```

## tstop.tcl

```
set tstop 2.370370e-03s
```

Matlab Generates top.tcl TCL command file with the same tstop for AMS

# 4. GFSK demodulator: Simulink test bench



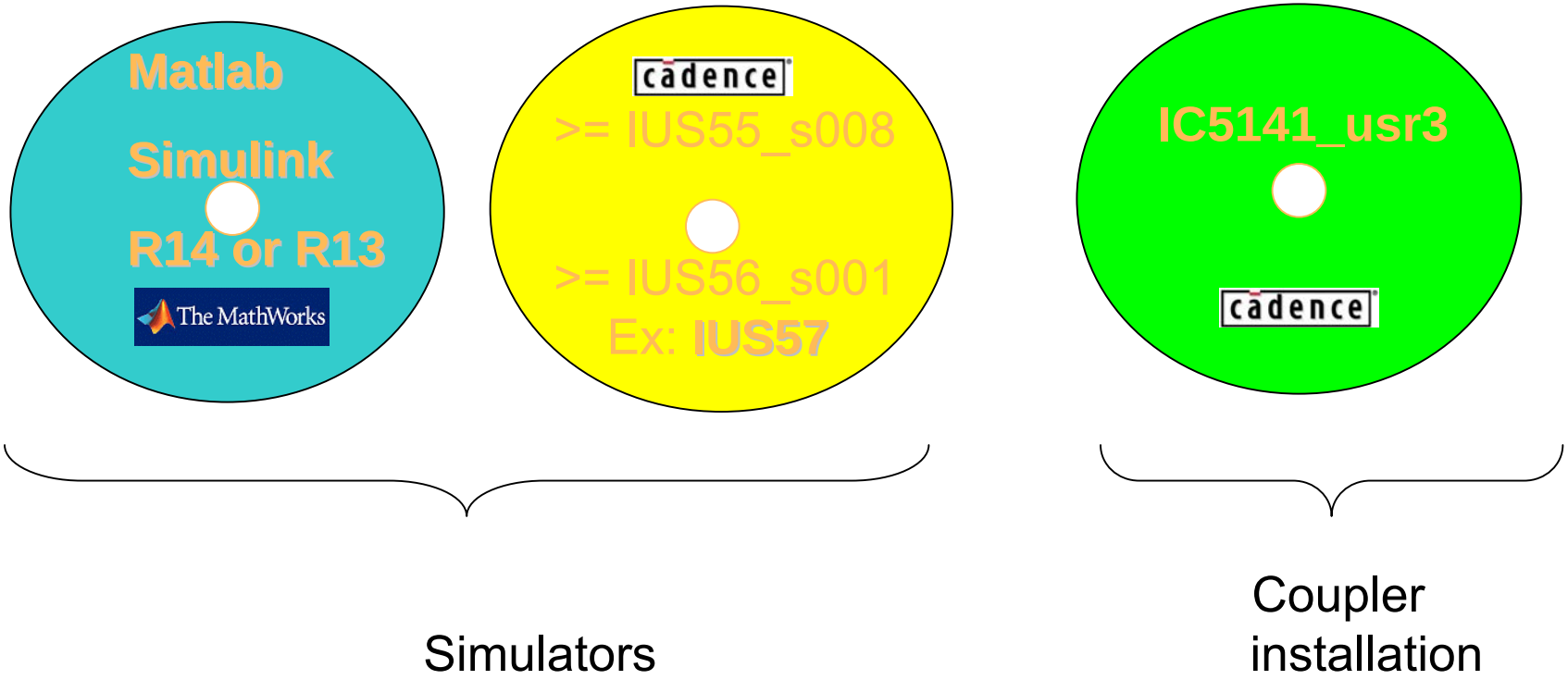
# 4. GFSK demodulator

GFSK demodulator is co-simulated sucessfully :
1 Millions samples in 8 minutes :
(CPU : CPU0 AMD Opteron(tm) Processor 250 2.4 GHz)

# 5. Which software do you need?

Cadence's Slide

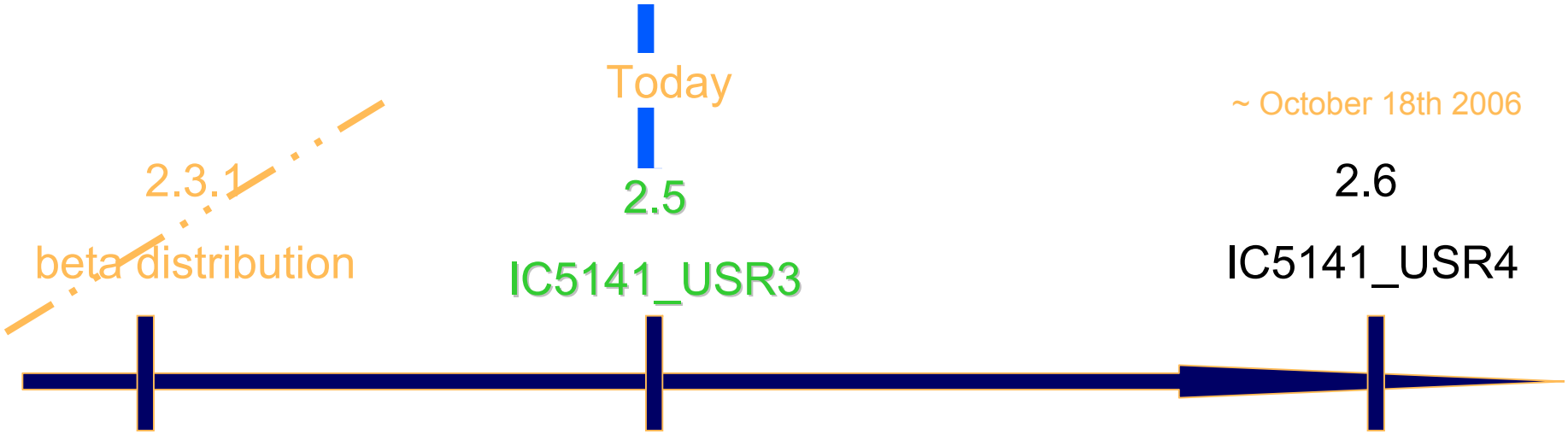
- This includes Linux, Solaris and Windows (Matlab only) support.




- The co-simulation will work together only with IUS55_s008/IUS56_s001/IUS57 or higher.

  Please make sure that your IUS version meets that requirement

# 5. Coupler versions & co-simulation roadmap




- The v2.5 is the official engineering release in IC5141_USR3

Cadence's Slide

# 7. Conclusion

- Convenient pcell coupler in Virtuoso : the analog designer is able to use the coupler without having to change the Verilog AMS code (electrical inputs / ouputs)

- Sample & Hold : Test bench easily made, fast co-simulation time
  (5 mn versus 15 mn for Spectre but sample & Hold with buffers and less simulation points)

- Polyphase filter and GFSK demodulator (VHDL) co-simulated with success and with fast run time
  - GFSK demodulator is simulated sucessfully : 1 Millions samples in 8 minutes:
    (CPU : CPU0 AMD Opteron(tm) Processor 250 2.4 GHz)

- Simulink can start AMS-Simulators actions (ncvhdl+Ncvlog+ncelab+ncsim)

- The designer can create iterations for loaching several co-simulations
  - Advantages:
    - Parametric
    - Statistical
    - Increase number of bits, sampling frequency, BER as a function of Input Signal power,…

# 7. Conclusion / Next steps

- Converter: MDAC within ADC 14 Bits: Electrical / Transistor level : more transistors (analog 700) and inputs (40), concatenation to be made : modification of the Coupler Verilog AMS

- BLs Cordless: Complete RX chain with the integration of a Sigma-Delta ADC: Top down design / Bottom-up Verification

- CTO: Co-simulation AMS Simulators – Simulink: from development to production and deployment in 2007 ?

# Acknowledgments:

André Baguenier: Cadence
Martin Barnasconi: CTO Philips
Bertrand Cesbron:CAD Philips
Fernand Courtois: System & Design Philips
Sylvain Dumont: System Philips
Christophe Erdmann: Design Philips
Olivier Girard: CTO Philips
Olivier Jamin: Design Philips
Philippe Gandy: Design Philips
Olivier Giard: CTO Philips
Marcel Hansen: ED&T Philips
Eric Lehouelleur: CTO Philips
Fabian Rivière: Design Philips
Tony Vasseaux: System Philips
Ascension Vizinho-Coutry: Mathworks
Giorgia Zuchelli: ED&T Philips

PHILIPS