Modeling aspects for DSL Systems

The ever growing demand for increased bandwidth and data rate performance on the local loop paired with flexibility of various configuration possibilities lead to an explosion of the complexity in DSL Systems. This explosion can simply be shown by comparing ISDN with VDSL2. The frequency bandwidth (125 kHz for ISDN, 30 MHz for VDSL) has increased by more than a factor of 200 and the data rate performance improved by more than a factor of 700. As a matter of fact also the channel models became more complex as crosstalk has severe influence over this higher frequency bandwidth.
Given these constraints a simulation platform is necessary which allows handling the above mention complexity on one hand and leaving enough room for design space exploration on the other hand. This implies ease of modeling and sufficient simulation speed.
Furthermore it is required that this simulation platform provides reference models for the design of analogue mixed signal blocks, digital blocks and serves as a platform for developing algorithms on the physical layer and higher abstraction layers as well.
Not to forget the support of standardization activities in various communities like ITU or ETSI with simulation results in order to show the influence of certain parameters and architectural concepts on performance in an early phase to drive the standardization into the right direction.
To address the above mentioned requirements a modular approach is chosen which utilizes Matlab for high complexity calculations and optimization in frequency domain and a C/C++ Modeling approach for the time domain.
The Matlab platform is mainly used to support all linear mixed signal design aspects like

- frequency response and noise modeling
- digital and analog filter optimization
- hybrid optimization (minimizing number of external components)
- calculation of the channel capacity
- configuration of bandplans and profiles
- configuration of mixed signal blocks, channel and crosstalk models of the C/C++ platform.

The C/C++ platform features are as follows:

- Modular in order to be able to include different models with different abstraction levels from various design locations into a single platform.
- Different abstraction levels varying from floating point behavioral to fixed point, down to bit true level.
- Simple selection of abstraction level by choosing the appropriate model library.
- Highly efficient channel models and analog models as these are the most simulation resource demanding blocks due to the high sampling rate.

The results show that with this platform concept we are able to handle the above mentioned complexity very well but still being flexible and fast enough to provide results in a timely manner and to extend the features further according to ever growing demands.

Modelling Aspects for DSL

Challenges and Solutions

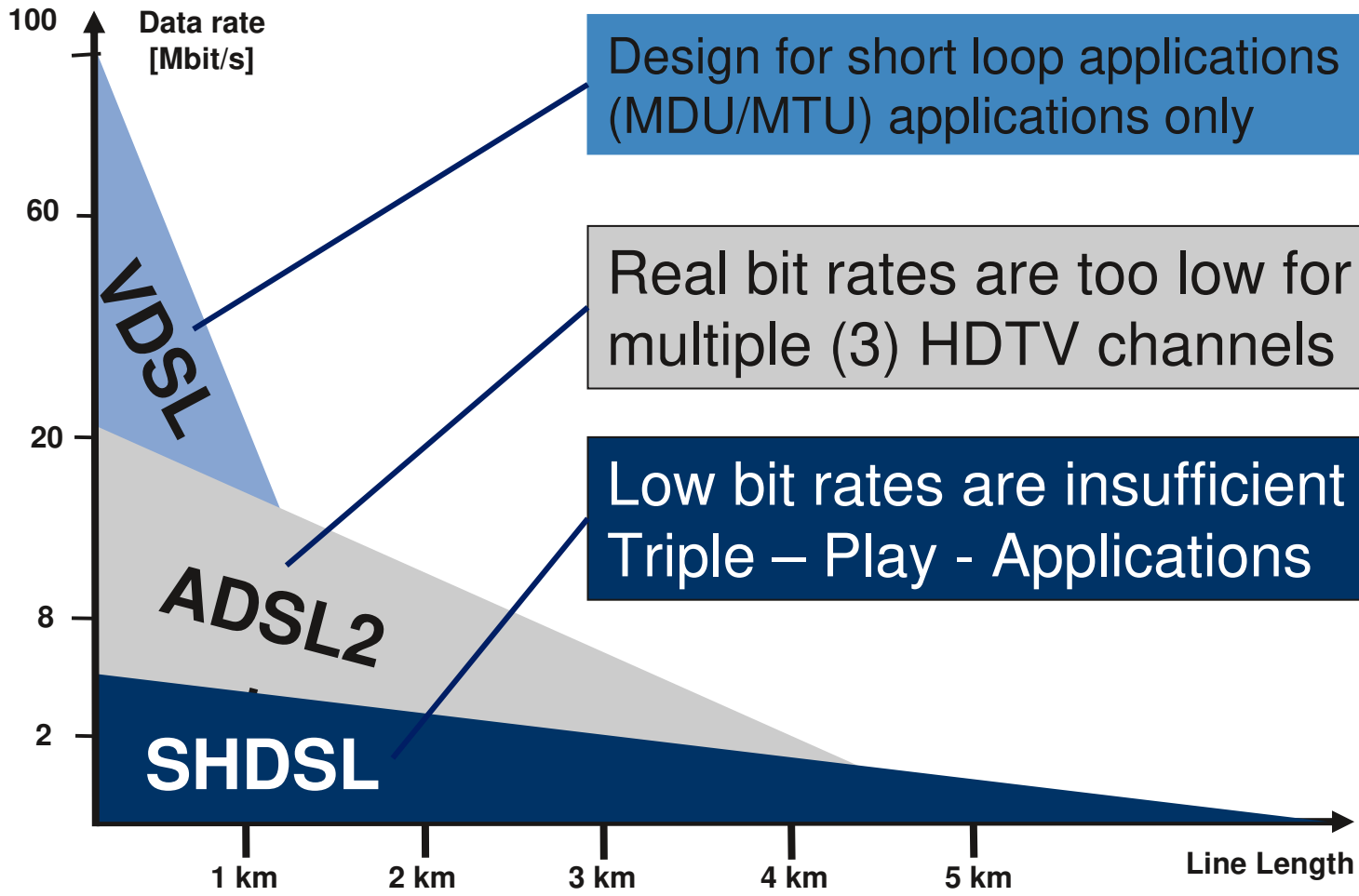
H. Zojer COM AC SE DSL
A. Clausen COM AC SE DSL
R. Stolle FHA - Fachbereich Elektrotechnik

Never stop thinking

Modelling Aspects for DSL

- Introduction to highest performing DSL: VDSL2
  - Basic Architecture
  - Modulation scheme
  - Requirements
- Frequency Domain simulations
- Time Domain Simulations
- Benchmark Results
- Outlook

… And Existing High Speed Technologies Do Not Solve the Problem of Bottlenecks…



Design for short loop applications (MDU/MTU) applications only

Real bit rates are too low for multiple (3) HDTV channels

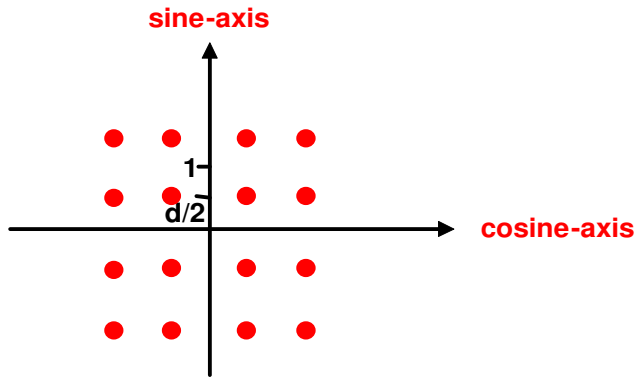
Low bit rates are insufficient for Triple – Play - Applications

VDSL2 = VDSL Speeds with ADSL/2+ Reach and Flexibility

VDSL Modulation

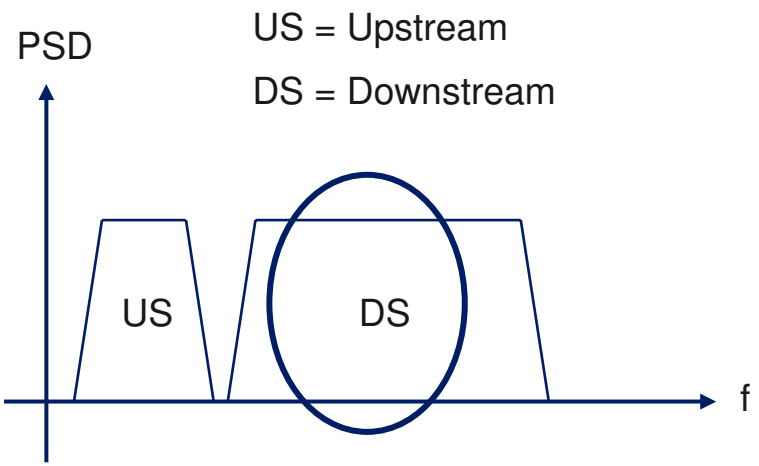
- Discrete-Multi-tone Modulation: similar to OFDM (Orthogonal Frequency Division Multiplexing) which is used in many wireless standards (Wireless LAN, DVB-T, DVB-H)
- Up to 4096 tones are used to carry data via QAM modulated signals

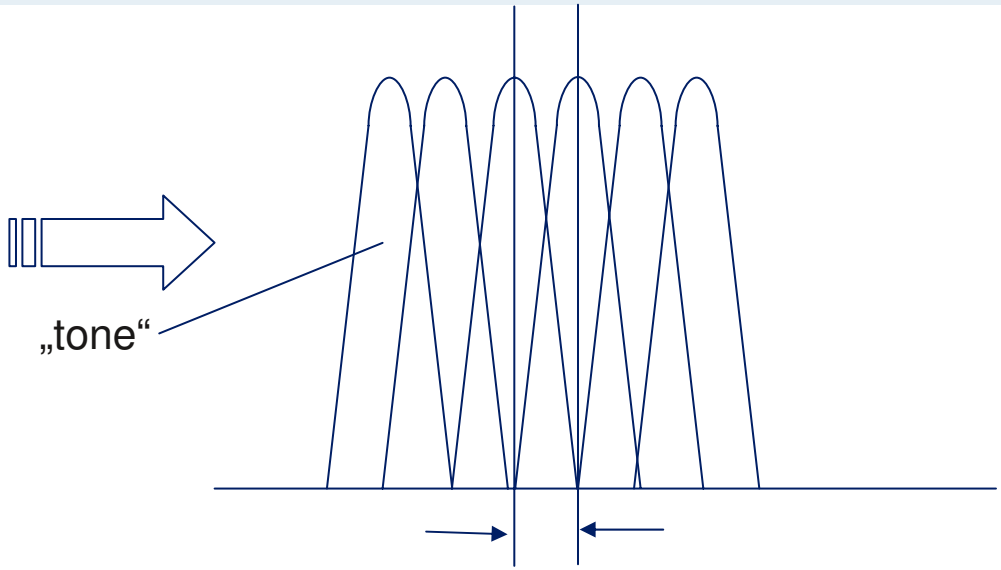
Example: 16 point-QAM



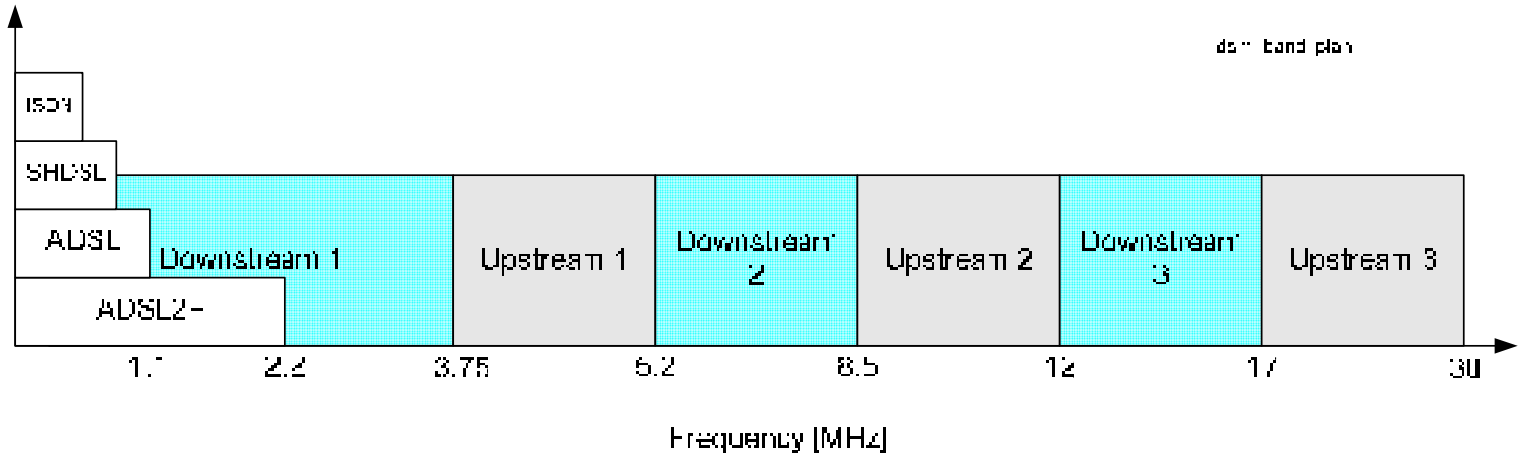
VDSL uses 2 to 32768 point QAM
1…15 bits per constellation

Spectrum: Example

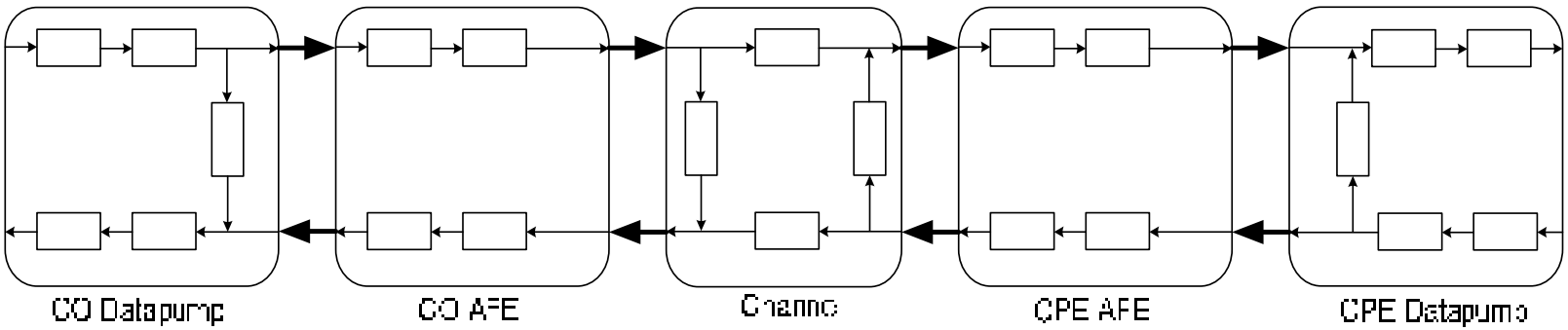




Tonespacing: 4.3125 kHz



VDSL2 Basic Block Diagram



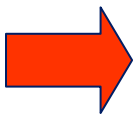
Requirements for the Simulation Platform

What should it be used for?

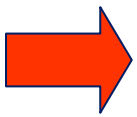
- Algorithm Development
- Hardware-/software partitioning
- Study Quantization effects
- Determine Data flow, data throughput, DSP requirements
- Hardware Verification
- Firmware Development
- Performance Estimation

What other features do we expect?

- Very Fast
- Accurate
- Ease of Modeling
- Ease of Use
- Modular
- Multi Level, Multi Domain
- Multi Site, Multi User
- Independent of Operating System

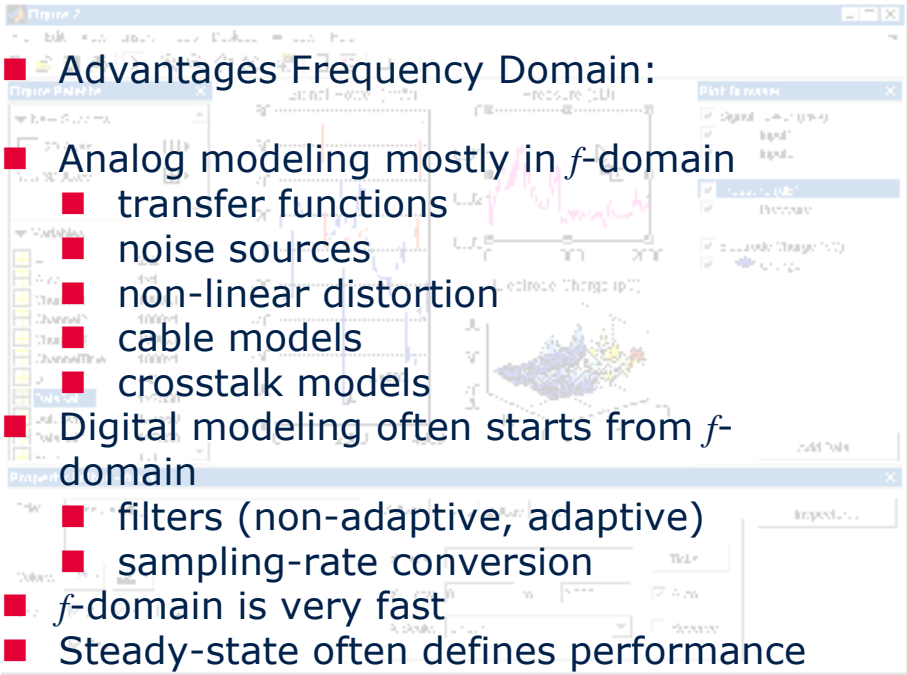


Complexity of current projects does not allow for a single simulation platform to cover all aspects.

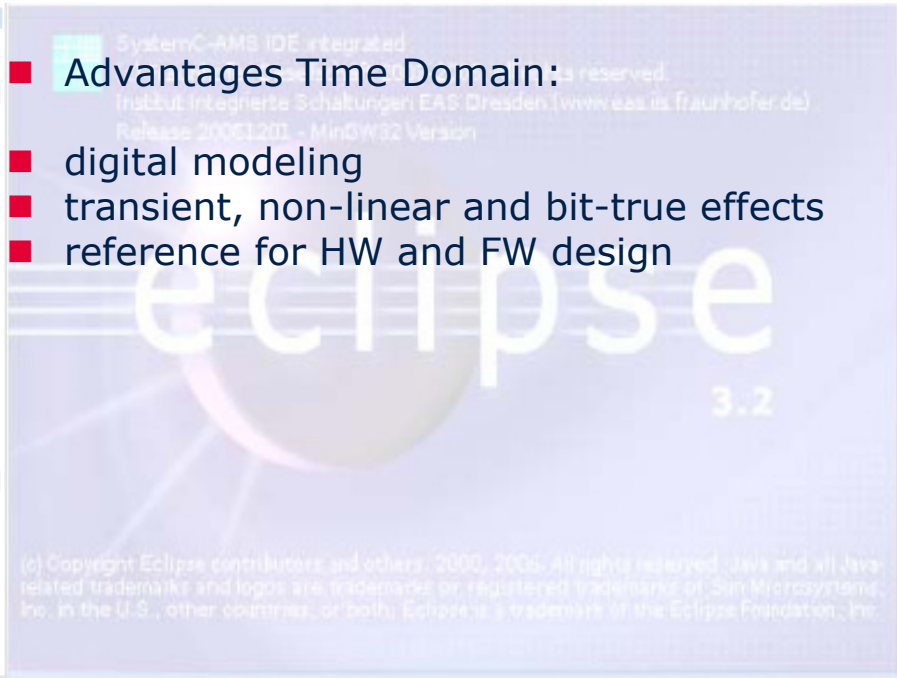


Solution: creating two platforms with “smart” interaction.

Time vs. Frequency Domain



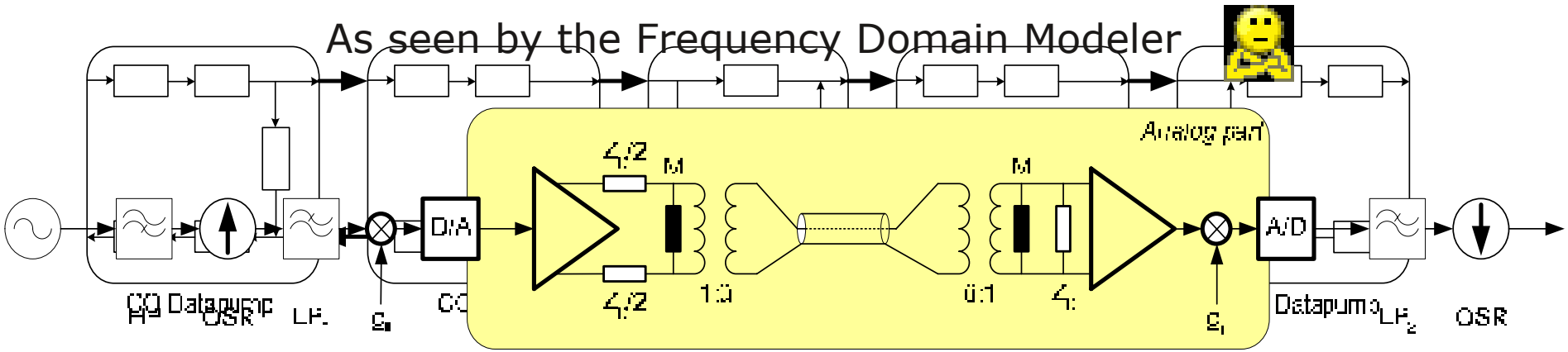
- Advantages Frequency Domain:
- Analog modeling mostly in f-domain
  - transfer functions
  - noise sources
  - non-linear distortion
  - cable models
  - crosstalk models
- Digital modeling often starts from f-domain
  - filters (non-adaptive, adaptive)
  - sampling-rate conversion
- f-domain is very fast
- Steady-state often defines performance targets
- f-domain can be enhanced by analytical procedures

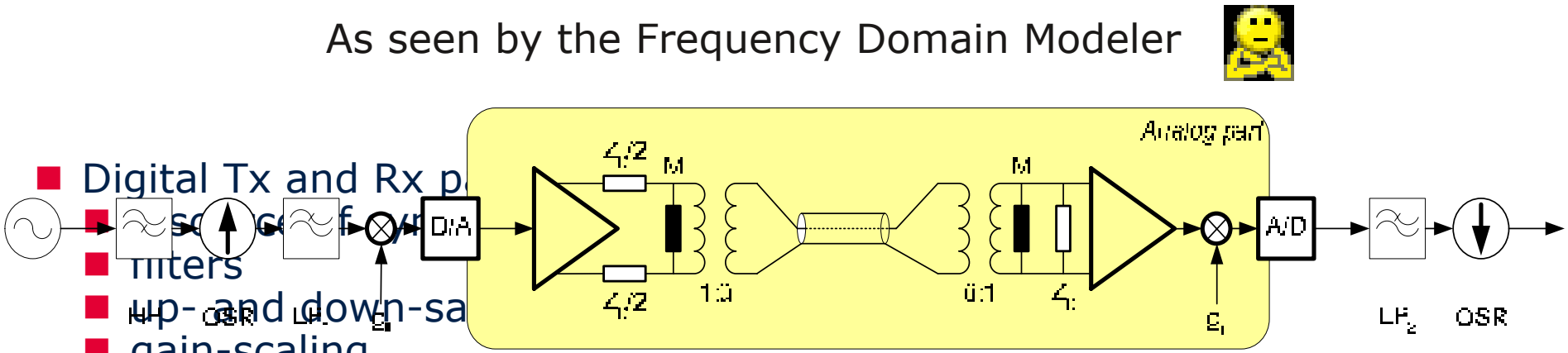


- Advantages Time Domain:
- digital modeling
- transient, non-linear and bit-true effects
- reference for HW and FW design

both environments are necessary and must use the same database!

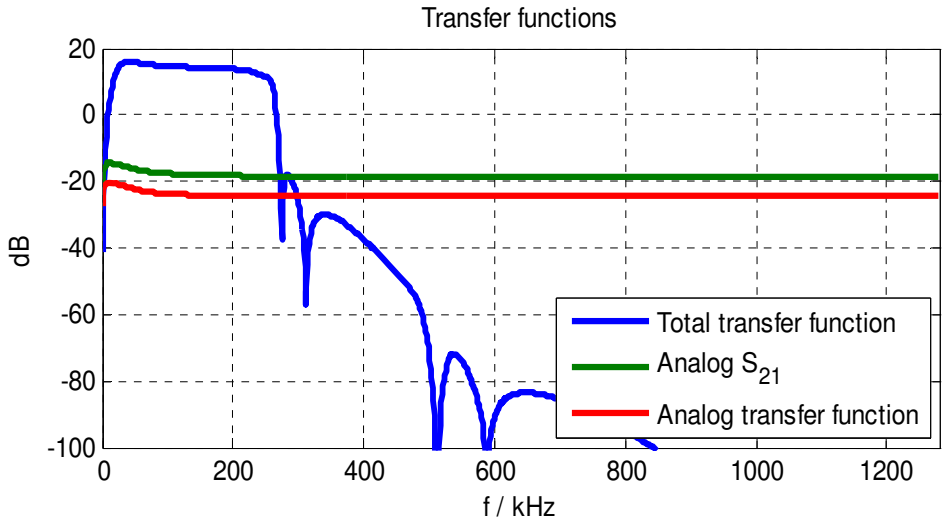
The exemplary communication system

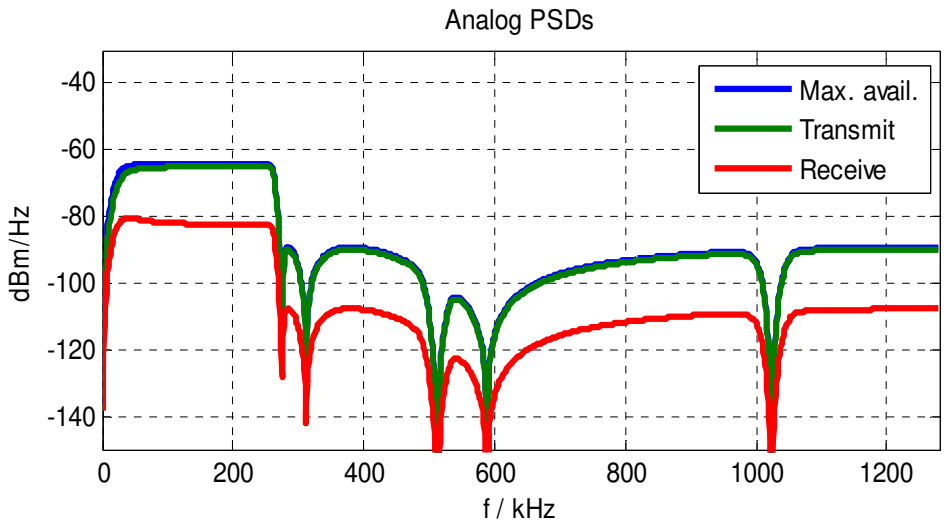




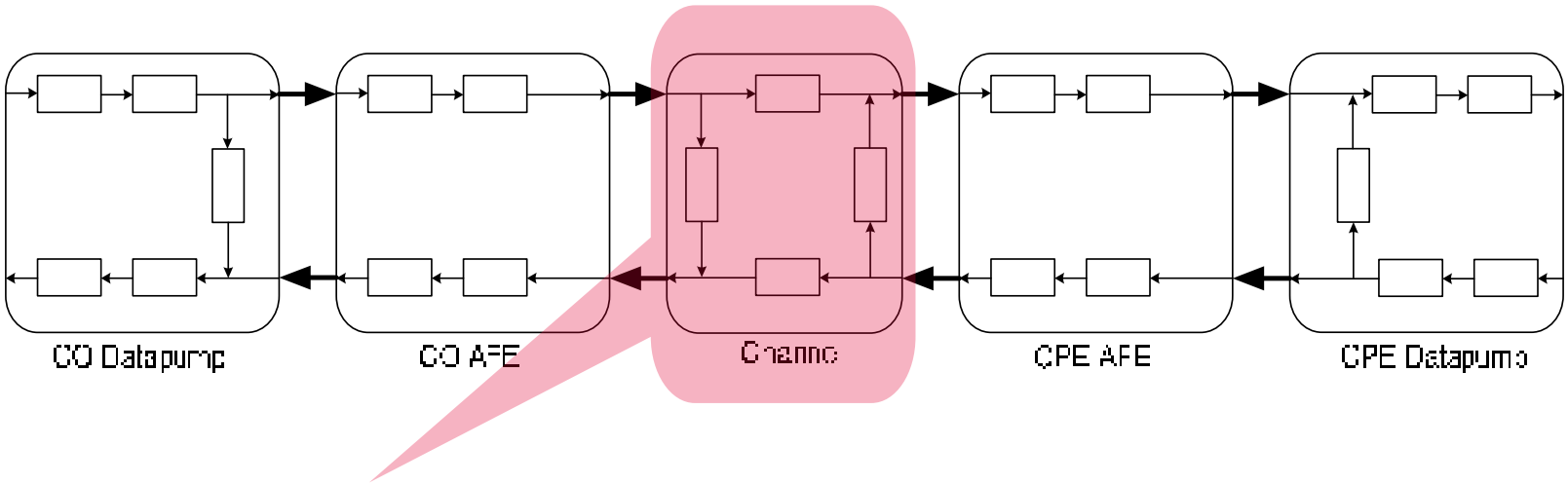
- Digital Tx and Rx p
  - sc
  - f
  - filters
  - up- and down-sa
  - gain-scaling
- Analog part, consisting of
  - a cable
  - analog circuitry (resistors, inductors, transformers)
  - gain-scaling (e. g. Automatic Gain Control)

Result from f Domain Modeling: Transfer functions and PSDs





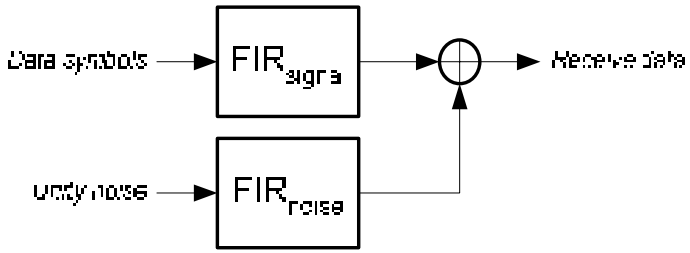
Time-domain Simulation



- Configured by f-domain simulation
- runs at smallest required rate (preferrably) nyquist rate

- consists of CO Transceiver, Channel, and CPE Transceiver Model
- separate (simplified) state machines for CO and CPE
- includes all signal processing from bit-loading to the slicer
- supports different simulation modes: full system, upstream only, downstream only, transmit only, …
- channel models can run at different rates

Build a noise and channel filters for t-domain simulation



- Channel and Noise Transfer-function are converted into an FIR filters
- The FIR coefficients are provided for the time-domain simulation
- The noise source is a random number generator with gaussian distribution

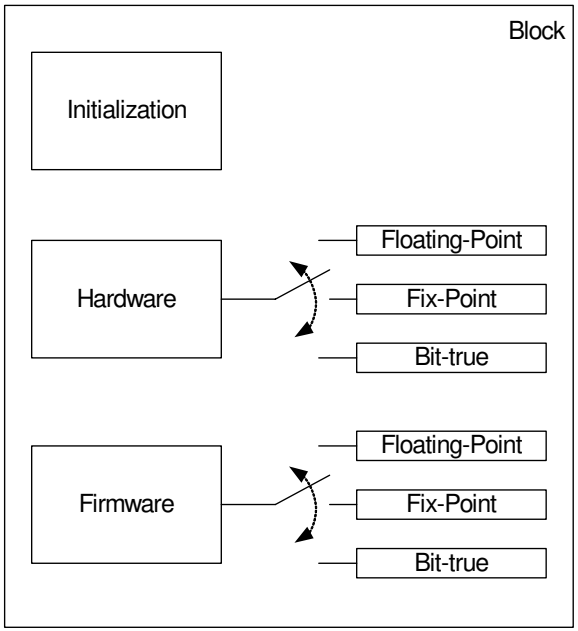
```
% interpolate Wndec on an appropriate frequency vector
Nfreq_int = 2^nextpow2(Nfreq/OSR);
Freq_Fsym_int = (0:Nfreq_int-1)'.*(Fsym/2/Nfreq_int);
Wndec_int = interp1(Freq_Fsym,Wndec,Freq_Fsym_int,'pchip');
% use the FFT method
IRn = fftshift(real(ifft(...
      [sqrt(Wndec_int(1));sqrt(Wndec_int(2:end)).*2],...
      2*Nfreq_int)));

% rip off 2*Nrip some samples to shorten the filter
Nrip = 240; % total length = 2*256
IRnshort = IRn(Nrip+1:2*Nfreq_int-Nrip) ...
           .* hanning(2*(Nfreq_int-Nrip),'periodic');
% turn into minimum phase filter
[ans,IRnminp] = rceps(IRnshort);

% verification of filters
Nsim = 2^13;

% re-simulate noise PSD
xn = randn(2*(Nfreq_int-Nrip)+Nsim,1); % input to noise-filter
disp('Simulating noise-filter')
yn = filter(IRnminp,1,xn); % output of noise-filter
yn = yn(2*(Nfreq_int-Nrip)+1:end); % rip off transients
[Wyy,Fn] = pwelch(yn);
% renormalize such that total power is the PSD average:
Wyy = Wyy.*pi;
% renormalize frequency vector:
Fn = Fn./pi; % Fn==1 at Nyquist frequency
```

Time-domain: Partitioning



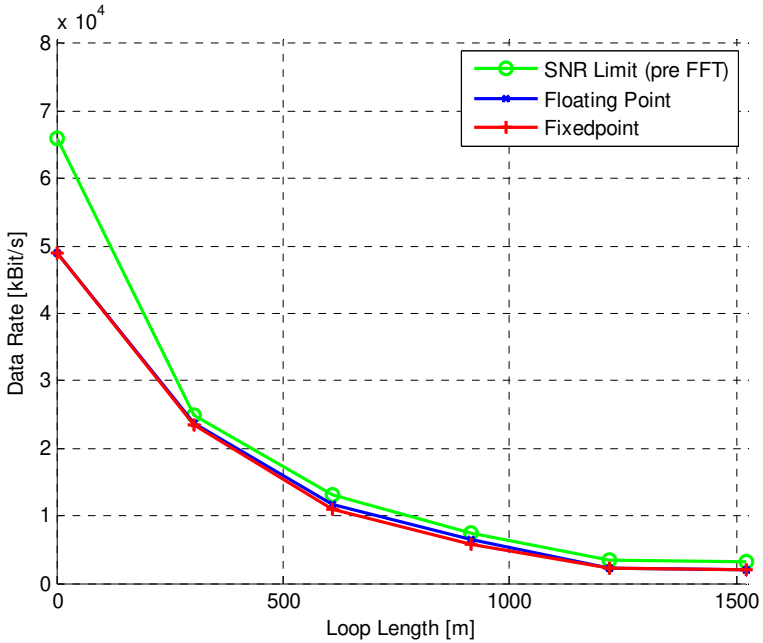
- Hardware/Software Partitioning: each block is partitioned into hardware and software functionality
- Quantization: for each block, hardware and software routines can be switched between floating-point, fix-point, and bit-true

Time-domain Simulation Environment

- Criteria for Decision:
  - Speed
  - Cost (Licenses)
  - Data transfer between simulation platforms
  - Portability --- Platform (UNIX, PC)
  - Reuse
  - Version Control

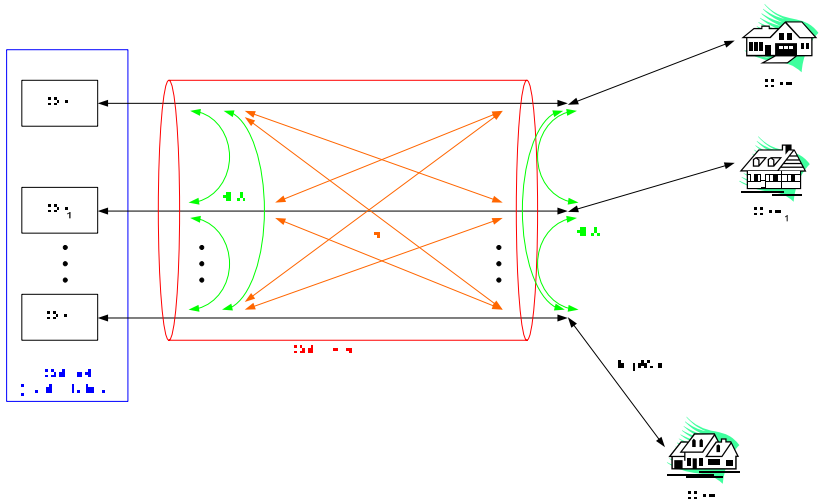
- Solution:
  - Simulation Environment in C++
  - ANSI C for all blocks which must be shared
  - Bit-true Models of AFE in System-C

Benchmark Results



- Rate & Reach performance
- Processor
  - 64 Pentium IV Linux
- Simulation time for one Performance Point
  - Frequency domain < 1min
  - Time Domain ~75min

Challenge for the future



- Up to 100% Performance Gain can be achieved by Crosstalk Cancellation
- Typically 50 lines in one binder
- Focus on dynamic issues like adding and dropping lines
- This means 50 times more complexity
- 50 times simulation speed
- New modeling techniques and concepts must be developed


Infineon
technologies
Never stop thinking
www.infineon.com