simDM ACT BCI Simulation

Contribution to 802.3dm Task Force Ad Hoc

December 19, 2024

Ragnar Jonsson - Marvell

Introduction

- This presentation evaluates the performance of an ACT low data rate (camera) receiver in the presence of simulated Bulk Current Injection (BCI)
- The simulation is based on simulation framework presented in a separate presentation (jonsson_3dm_01_12_19_24.pdf), and the simulation code is provided for reference and to allow more thorough review
- The simulation assumes
  - 30mV peak voltage for each tone, based on the worse 2m case from Pischl_3dm_01a_1124.pdf
  - The good 15m cable from jonsson_3dm_02_09_15_24.pdf (note that BCI tests are normally conducted on about 2m cables)

Simulation Details

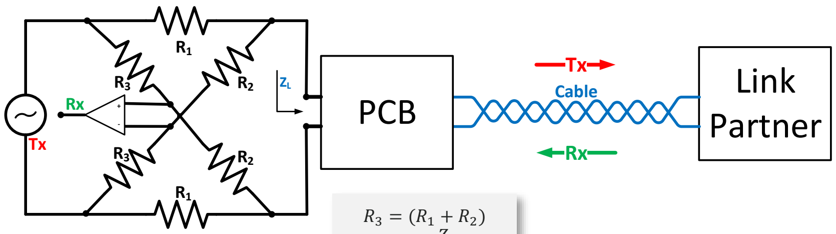
- The simulation uses a 22.5GHz sampling rate to represent analog signals, and analog signal levers are represented in Volts
- The simulation includes echo from the High Data Rate (HDR) transmitter
- The HDR path transmits at 2.5Gbps, because it causes the maximum echo into the LDR receiver
- The simulation combines the PCB echo and the Hybrid echo path into single filter in the pcb.echo_filter
- The simulation results in this presentation are captured as eye diagrams, to more clearly show how little impact the injected tones have at the LDR receiver

Modeling Hybrid as Simple Resistor Bridge

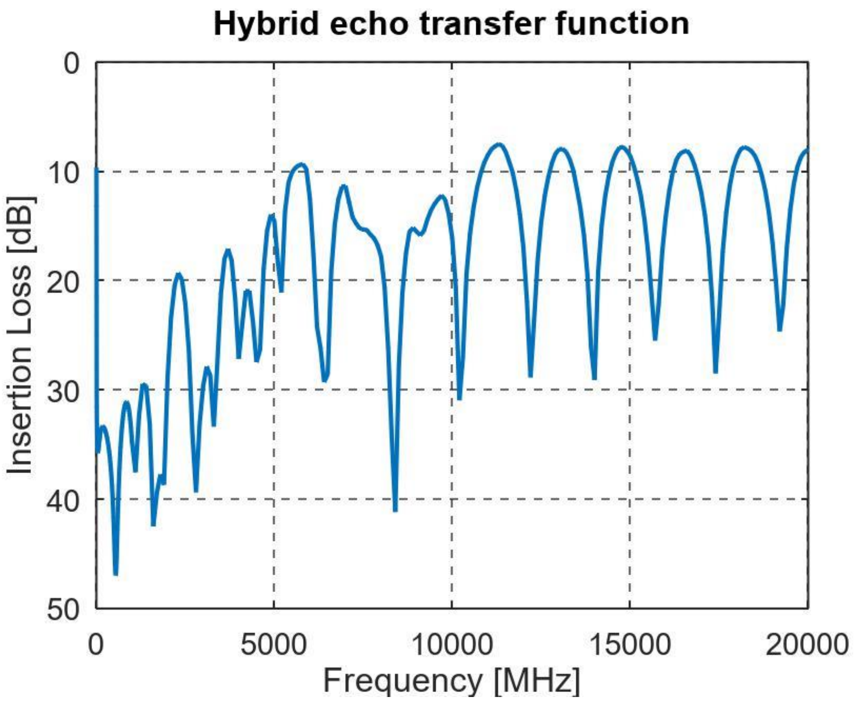
- Hybrid can be modeled as a simple resistor bridge (see figure below)
- For the simulation the transfer function from Tx to Rx is calculated as

$$H(f) \approx \frac{Z_L(f)}{2R_1 + Z_L(f)} - \frac{1}{2}$$

where R_1 = 50Ω for differential mode.



$$R_3 = (R_1 + R_2)$$
$$R_1 = \frac{Z_L}{2}$$
$$Z_L \ll R_2 + R_3$$

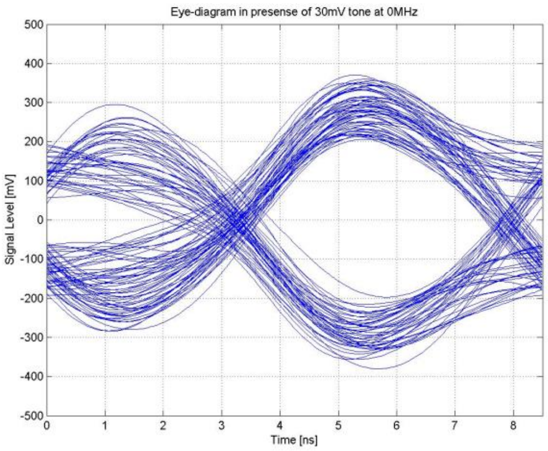


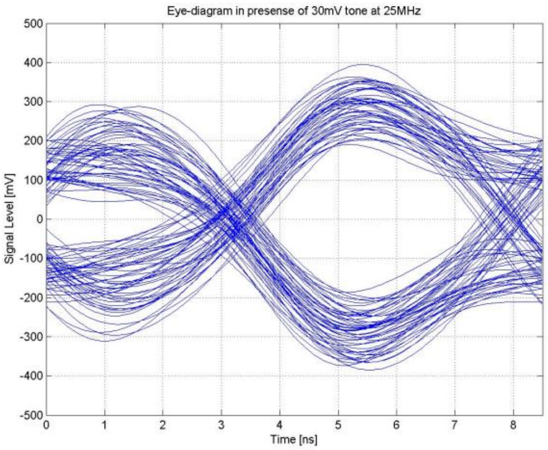
MATLAB Code for ACT simDM Simulation of BCI

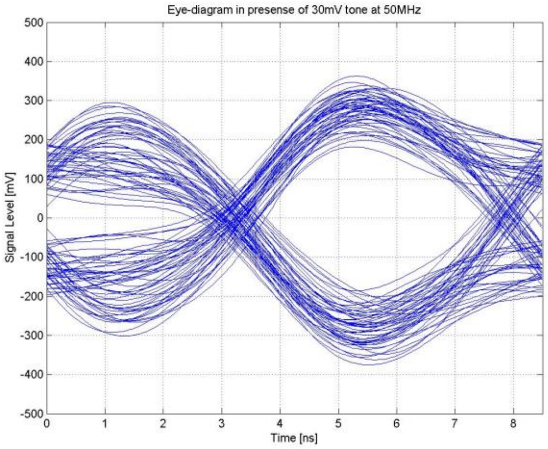
```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simple simulation of Bulk Current Injection and ACT camera receiver
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is simulation code provided to help with the development of
% IEEE 802.3dm.
%
% This code is provided for reference to allow independent evaluation
% of the accuracy and applicability of the simulation results shared in
% IEEE 802.3dm presentations by the author.
%
% Written by Ragnar Jonsson, affiliated with Marvell Technology, Inc.
% Version 1.0, December 16th, 2024
%
% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
% OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
% FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
% THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
% LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
% FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
% DEALINGS IN THE SOFTWARE.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% simulate 100M/2.5G link %%%
hdr_rate = 2.5;
is_coax = 1;
pcb_cuttoff = 10;
ldr_symbols = 100;

%%% load "good" cable from jonsson_3dm_02_09_15_24 %%%
load jonsson_3dm_02_09_15_24_good.mat

%%% load hybrid model %%%
load jonsson_e241216b_hybrid_echo.mat

%%% configure simulation variables %%%
t = [1:(192*ldr_symbols)]/22.5e9;
t200 = [0:(192-1)]/22.5e9;
print_plot = 1;

%%% define test cases %%%
f_tones = [0:25:400];
N_tones = length(f_tones);
A = 30;
for n = 1:N_tones,
    f_t = f_tones(n);
    noise_cases{n,1} = sprintf('state.out = %f*sin(t*2*pi*%f*1e6);',A/1000,f_t);
    noise_cases{n,2} = sprintf('%dmV tone at %dMHz',A,f_t);
end

%%% loop over all test cases %%%
N = length(noise_cases);
for m = 1:N,
  %%% create new simulation instance %%%
  act = simDM_ACT(hdr_rate,is_coax,pcb_cuttoff);

  %%% configure the hybrid echo filter %%%
  act.dut.pcb.echo_filter.b = h_hybrid_echo;

  %%% configure the channel model %%%
  act.channel.p11.b = h11;
  act.channel.p12.b = h12;
  act.channel.p22.b = h22;
  act.channel.p21.b = h21;

  %%% no AFE internal noise in this simulation %%%
  act.dut.afe.noise.eval = 'state.out = 0.000*randn(size(x));';
  act.lkp.afe.noise.eval = 'state.out = 0.000*randn(size(x));';

  %%% configure noise generator for the DUT %%%
  act.dut.pcb.noise.eval = noise_cases{m,1};

  %%% run 4 rounds of 100 LDR symbols %%%
  for n = 1:4,
    lkp_tx_symbol = round(rand(1,ldr_symbols))*2-1;
    dut_tx_symbol =
floor(3.99999*rand(1,act.ldr_oversampling*ldr_symbols/act.hdr_oversampling))*2-3;
    act = link(act,lkp_tx_symbol,dut_tx_symbol);
  end

  %%% plot the eye-diagram for each test run %%%
  figure
  ldr_out_oversampling = length(act.dut.afe.rx_out)/ldr_symbols;

plot(t200(1:(1/act.dut.afe.adc_sampling.rate):end)*1e9,reshape(act.dut.afe.rx_out,ldr_out_overs
ampling,ldr_symbols)*1e3,'b')
  xlabel('Time [ns]')
  ylabel('Signal Level [mV]')
  grid
  axis([0 8.5 -500 500]);
  title(sprintf('Eye-diagram in presense of %s',noise_cases{m,2}));
  if(print_plot)
    %print(sprintf('%s_eye_%s_%dMHz.jpg',mfilename(),strrep(noise_cases{m,2},' ','_')))
    saveas(gcf,sprintf('%s_eye_%s_%dMHz.jpg',mfilename(),strrep(noise_cases{m,2},'
','_')),'jpg')
  end
end
```

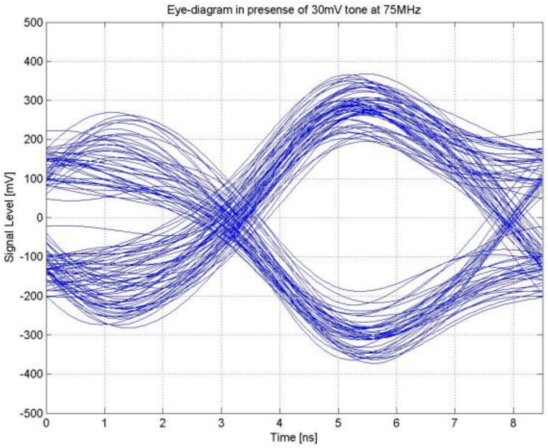
Eye-diagram
0 - 75MHz

- Eye is clearly open in all cases



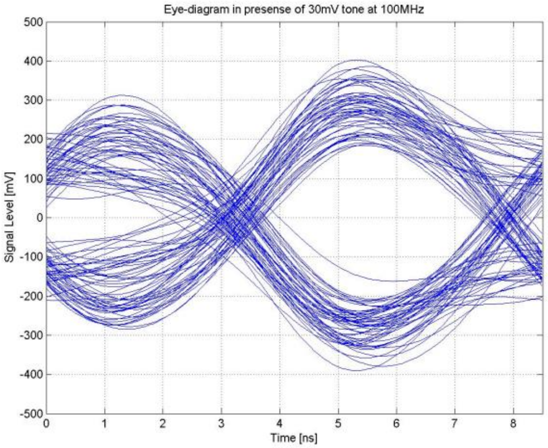


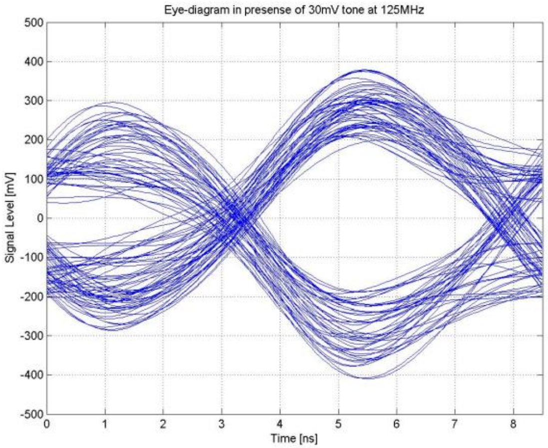


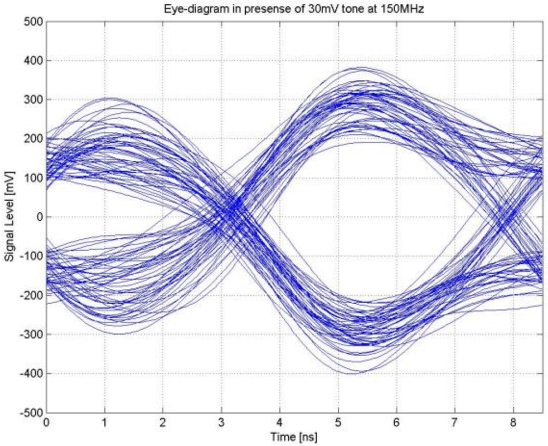


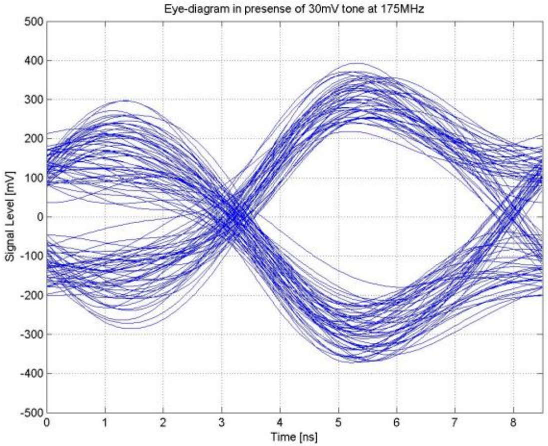
Eye-diagram 100 - 175MHz

- Eye is clearly open in all cases



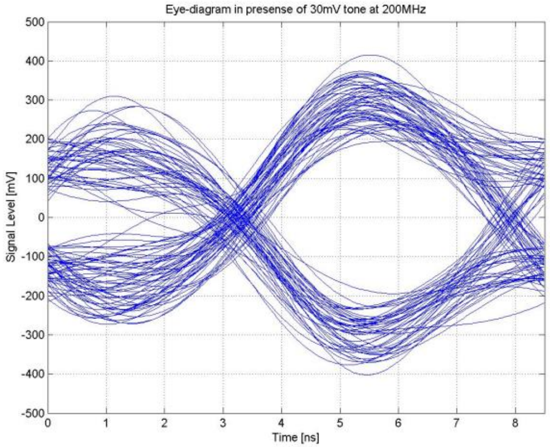


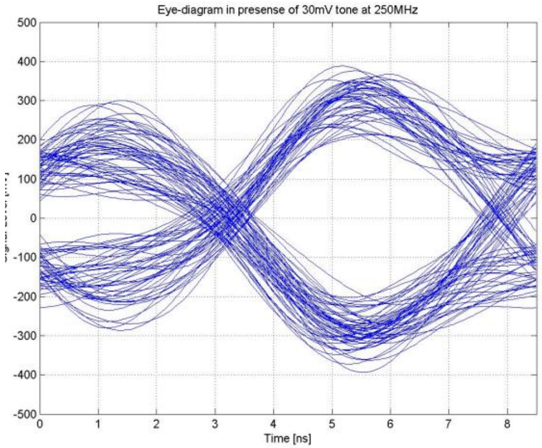


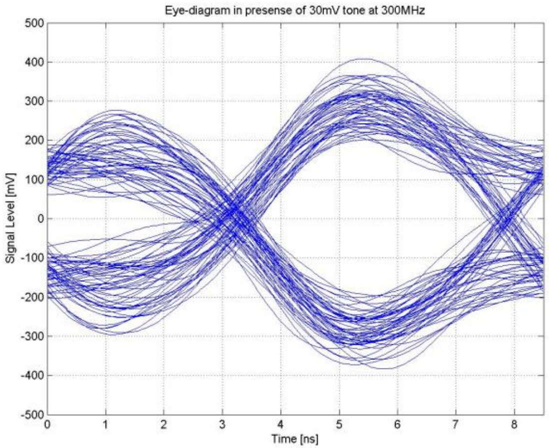


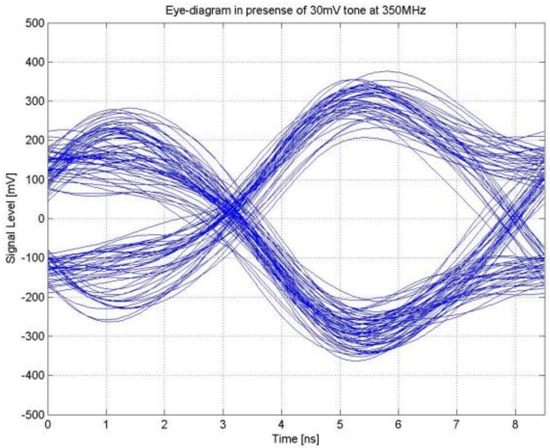
Eye-diagram 200 - 400MHz

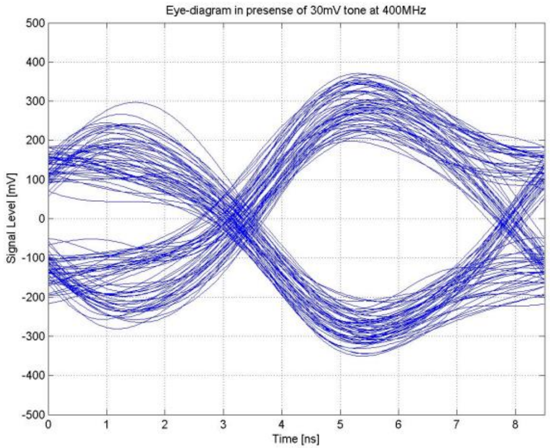
- Eye is clearly open in all cases











Summary

- This simulation shows that the ACT low data rate receiver (camera receiver) has robust performance in the presence of single tone BCI
- This is the first simulation done with the simDM framework that was introduced in jonsson_3dm_01_12_19_24.pdf, but more simulations are planed for the coming weeks
- The simulation code has been made available, and all feedback and suggestions are greatly appreciated

MARVELL™

Essential technology, done right™