Follow-up to COM Commit Request Number 4p8_5

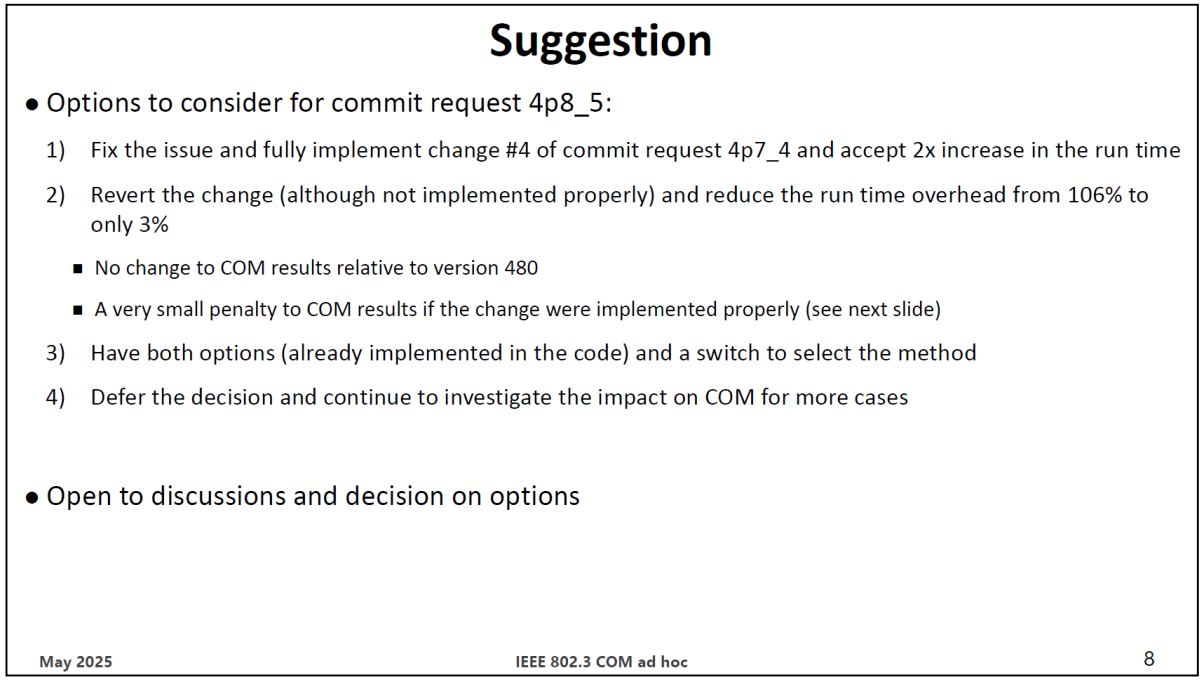
COM Commit Request Number 4p9_1

Hossein Shakiba
Huawei Technologies Canada
June 17 2025

Background

- Commit request 4p8_5 was presented in the COM ad hoc meeting on May 05, 2025 during the May interim in New Orleans (shakiba_3dj_COM_03_2505.pdf)
- The request was to address an issue with implementation of an earlier commit request (change #4 of commit request 4p7_4) as well as to decide on the opportunity to reduce the runtime when quantization noise feature is enabled
- Four options were presented →
- Consensus was to proceed with Option 3
- A follow-up was requested to provide more content on option 3 and a code submission request through the open source repository
- Since now version 4p90 is available, this follow-up presentation and the code change request are relative to version 4p90

Slide 8 of "shakiba_3dj_COM_03_2505.pdf"



Suggestion

- Options to consider for commit request 4p8_5:
  1) Fix the issue and fully implement change #4 of commit request 4p7_4 and accept 2x increase in the run time
  2) Revert the change (although not implemented properly) and reduce the run time overhead from 106% to only 3%
     - No change to COM results relative to version 480
     - A very small penalty to COM results if the change were implemented properly (see next slide)
  3) Have both options (already implemented in the code) and a switch to select the method
  4) Defer the decision and continue to investigate the impact on COM for more cases
- Open to discussions and decision on options

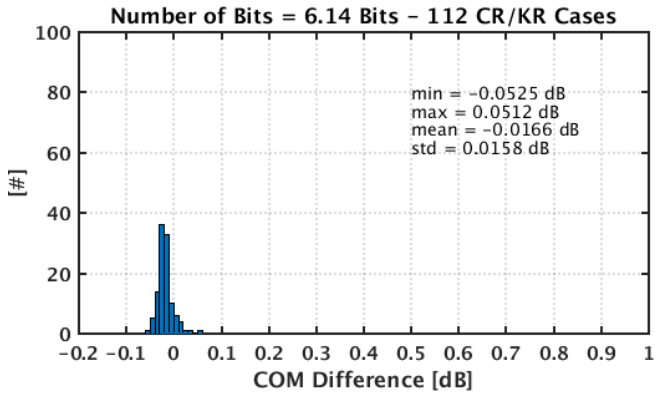
May 2025 IEEE 802.3 COM ad hoc 8

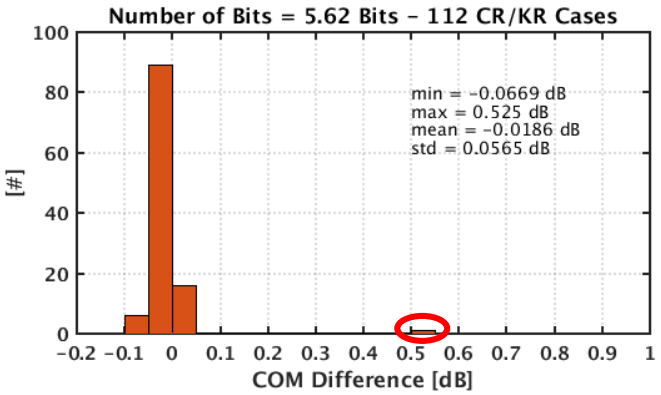
Introduction

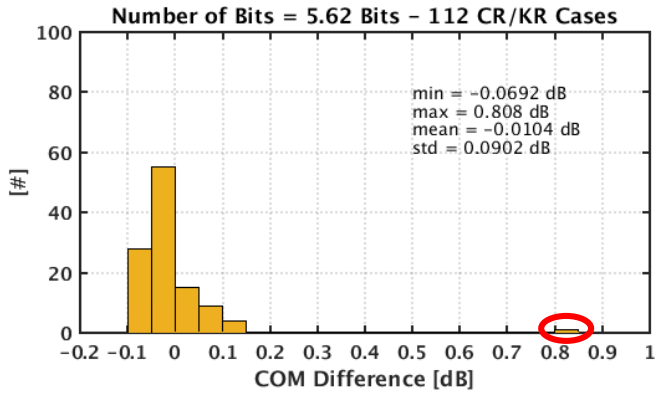
- Two methods have been considered for calculation of quantizer clip level during the optimization loop
  1) "Fast" (less accurate)
  2) "Slow" (more accurate)

Average Runtime Overhead "Fast" Method	Average Runtime Overhead "Slow" Method
3% Overhead	106% Overhead

- For 3x112 of test cases COM difference between two methods is almost negligible except for two cases







- Option 3 implements both methods and enables the user to select one through a switch defined as a parameter in the COM configuration

Description of the Change to Implement Option 3

- Both methods are already available in the code (in function “get_PSDs”)
- What the change does:
  1) Addition of a switch to select between two methods in the “get_PSDs” function
     a) Bypass calculation of pulse response during optimization iterations if “Fast” method is selected
     b) Only calculate signal PDF during optimization iterations if “Slow” method is selected
  2) Addition of a new parameter in the parameter section of “com_ieee8023_4p90” to select the method
- Further runtime reduction is expected for the “Fast” method due to the additional saving of 1)a) above
- Link to the branch containing new version of the code with the above changes:

  https://opensource.ieee.org/shakiba/com_code/-/tree/Quantization_Noise?ref_type=heads
- Link to the merge request:

  https://opensource.ieee.org/802-com/com_code/-/merge_requests/7

Change 1)a) “diff”

```
4891  4891        %% S_tn from eq 178A-17
4892  4892        %% if not in the opimization use value found in optimize_fom times |Hrxffe|^2
4893  4893        %% Transmitter noise power spectral density
4894        -     if ~OP.TDMODE
4895        -         htn=filter(ones(1,M),1,chdata(1).ctle_imp_response); % ctle_imp_response does not have TxFFE included
4896        -     else % only use when the input was a pulse response not s-parameters
4897        -         if isfield(chdata(1),'ctle_pulse_response')
4898        -             htn=chdata(1).ctle_pulse_response;
      4894  +     if ~OP.COMPUTE_COM || strcmp(param.clip_method, 'Slow') % "if" to "end" section changed by Hossein Shakiba to implement commit request 4p9_1
      4895  +         if ~OP.TDMODE
      4896  +             htn=filter(ones(1,M),1,chdata(1).ctle_imp_response); % ctle_imp_response does not have TxFFE included
      4897  +         else % only use when the input was a pulse response not s-parameters
      4898  +             if isfield(chdata(1),'ctle_pulse_response')
      4899  +                 htn=chdata(1).ctle_pulse_response;
      4900  +             else
      4901  +                 htn=filter(ones(1,param.samples_per_ui),1, chdata(1).ctle_imp_response);
      4902  +             end
      4903  +         end
      4904  +         htn=htn(mod(cursor_i,M)+1:end-mod(cursor_i,M)); % align to sample point
      4905  +         htn=reshape(htn,1,[]); % make row vectors
      4906  +         htn=[ htn(1:floor(length(htn)/M)*M) ];
      4907  +         htn= [htn zeros(1,num_ui*M-length(htn)) ];
      4908  +         htn=htn(1:M:end);% resample
      4909  +         if num_ui>length(htn)
      4910  +             hext=[htn zeros(1,num_ui-length(htn))];
4899  4911            else
4900        -               htn=filter(ones(1,param.samples_per_ui),1, chdata(1).ctle_imp_response);
      4912  +               hext=htn(1:num_ui);
4901  4913            end
4902  4914        end
4903        -     htn=htn(mod(cursor_i,M)+1:end-mod(cursor_i,M)); % align to sample point
4904        -     htn=reshape(htn,1,[]); % make row vectors
4905        -     htn=[ htn(1:floor(length(htn)/M)*M) ];
4906        -     htn= [htn zeros(1,num_ui*M-length(htn)) ];
4907        -     htn=htn(1:M:end);% resample
4908        -     if num_ui>length(htn)
4909        -         hext=[htn zeros(1,num_ui-length(htn))];
4910        -     else
4911        -         hext=htn(1:num_ui);
4912        -     end
4913  4915        if ~OP.COMPUTE_COM
4914  4916            result.S_tn=sigma_X2*10^(-SNR_TX/10)*(abs(fft(hext))).^2/param.fb; % this corresponds to +/- pi
4915  4917            result.S_tn_rms = sqrt(sum(result.S_tn)* delta_f);
4916        -     else
      4918  +     elseif OP.COMPUTE_COM % "elseif" condition changed by Hossein Shakiba to implement commit request 4p9_1
4917  4919            result.S_tn=result.S_tn.*H_rxffe_2;
4918  4920            result.S_tn_rms = sqrt(sum(result.S_tn)* delta_f);
4919  4921        end
```

Change 1)b) “diff”

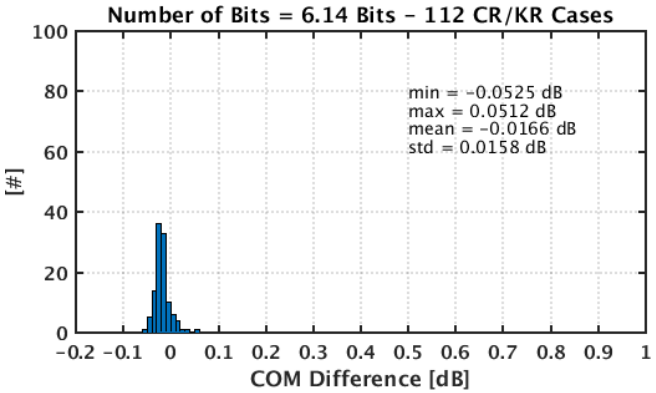
```
4954 4956          result.S_rj_rms = sqrt(sum(result.S_rj_jn)* delta_f);
4955 4957      end
4956 4958      %  result.S_qn
4957      -    if(param.N_qb ~=0)
4958      -        hext_txffe=filter(txffe,1,hext);
4959      -        sig_aftert_ctle_pdf = get_pdf_from_sampled_signal(hext_txffe,param.levels,OP.BinSize);
4960      -        noise_after_ctle_pdf = sig_aftert_ctle_pdf;
4961      -        sigma_noise = sqrt(result.S_rn_rms^2+result.S_xn_rms^2++result.S_tn_rms^2++result.S_rj_rms^2);
4962      -        noise_after_ctle_pdf.y = 1/(sqrt(2*pi)*sigma_noise)*exp(-noise_after_ctle_pdf.x.^2/(2*sigma_noise^2))*OP.BinSize;
4963      -        sig_noise_after_ctle_pdf= conv_fct(sig_aftert_ctle_pdf,noise_after_ctle_pdf);
4964      -        sig_noise_after_ctle_cdf = cumsum(sig_noise_after_ctle_pdf.y);
4965      -        ctle_signal_sigma = sqrt(sum((sig_noise_after_ctle_pdf.x.^2).*sig_noise_after_ctle_pdf.y));
4966      -        adc_clip=-CDF_inv_ev(param.P_qc, sig_noise_after_ctle_pdf,sig_noise_after_ctle_cdf);
4967      -        adc_lsb=2*adc_clip/(2^param.N_qb-1);
4968      -        sigma_Q=adc_lsb/sqrt(12);
4969      -        S_qn=sigma_Q^2/f_b*ones(size(hext));
4970      -        result.adc_clip=adc_clip;
4971      -        result.ctle_signal_sigma=ctle_signal_sigma;
4972      -        result.S_qn=S_qn;
4973      -        result.s_qn_rms=sqrt(sum(result.S_qn)*delta_f);
4974      -        if OP.INCLUDE_CTLE == 1
4975      -            eq_ir = TD_CTLE(chdata(1).uneq_imp_response, param.fb, param.CTLE_fz(1), param.CTLE_fp1(1), param.CTLE_fp2(1), 0_DC, param.samples_per_ui);
4976      -            eq_ir = TD_CTLE(eq_ir, param.fb, param.f_HP(1), param.f_HP(1), 100e100 , 0_DC2, param.samples_per_ui);
4959 4977      +    if(param.N_qb ~=0) % "if" to "else" section changed by Hossein Shokiba to implement commit request 4p9_1
4960 4978      +        if strcmp(param.clip_method, 'Slow')
4961 4979      +            hext_txffe=filter(txffe,1,hext);
4962 4980      +            sig_aftert_ctle_pdf = get_pdf_from_sampled_signal(hext_txffe,param.levels,OP.BinSize);
4963 4981      +            noise_after_ctle_pdf = sig_aftert_ctle_pdf;
4964 4982      +            sigma_noise = sqrt(result.S_rn_rms^2+result.S_xn_rms^2++result.S_tn_rms^2++result.S_rj_rms^2);
4965 4983      +            noise_after_ctle_pdf.y = 1/(sqrt(2*pi)*sigma_noise)*exp(-noise_after_ctle_pdf.x.^2/(2*sigma_noise^2))*OP.BinSize;
4966 4984      +            sig_noise_after_ctle_pdf= conv_fct(sig_aftert_ctle_pdf,noise_after_ctle_pdf);
4967 4985      +            sig_noise_after_ctle_cdf = cumsum(sig_noise_after_ctle_pdf.y);
4968 4986      +            ctle_signal_sigma = sqrt(sum((sig_noise_after_ctle_pdf.x.^2).*sig_noise_after_ctle_pdf.y));
4969 4987      +            adc_clip=-CDF_inv_ev(param.P_qc, sig_noise_after_ctle_pdf,sig_noise_after_ctle_cdf);
4970 4988      +            adc_lsb=2*adc_clip/(2^param.N_qb-1);
4971 4989      +            sigma_Q=adc_lsb/sqrt(12);
4972 4990      +            S_qn=sigma_Q^2/f_b*ones(size(hext));
4973 4991      +            result.ctle_signal_sigma=ctle_signal_sigma;
4977 4994          else
4978      -            eq_ir = chdata(1).uneq_imp_response;
4979      -        end
4980      -        ctle_pulse = filter(ones(1, param.samples_per_ui), 1, eq_ir);
4981      -        ind_max = find(ctle_pulse == max(ctle_pulse));
4982      -        adc_clip = sum(abs([ctle_pulse(ind_max-param.samples_per_ui:-param.samples_per_ui:1); ctle_pulse(ind_max:param.samples_per_ui:end)]));
4983      -        adc_lsb = 2*adc_clip/(2^param.N_qb-1);
4984      -        sigma_Q = adc_lsb/sqrt(12);
4985      -        S_qn = sigma_Q^2/(length(result.S_rn)*delta_f)*ones(size(result.S_rn));
4975 4995      +            if OP.INCLUDE_CTLE == 1
4976 4996      +                eq_ir = TD_CTLE(chdata(1).uneq_imp_response, param.fb, param.CTLE_fz(1), param.CTLE_fp1(1), param.CTLE_fp2(1), 0_DC, param.samples_per_ui);
4977 4997      +                eq_ir = TD_CTLE(eq_ir, param.fb, param.f_HP(1), param.f_HP(1), 100e100 , 0_DC2, param.samples_per_ui);
4978 4998      +            else
4979 4999      +                eq_ir = chdata(1).uneq_imp_response;
4980 5000      +            end
4981 5001      +            ctle_pulse = filter(ones(1, param.samples_per_ui), 1, eq_ir);
4982 5002      +            ind_max = find(ctle_pulse == max(ctle_pulse));
4983 5003      +            adc_clip = sum(abs([ctle_pulse(ind_max-param.samples_per_ui:-param.samples_per_ui:1); ctle_pulse(ind_max:param.samples_per_ui:end)]));
4984 5004      +            adc_lsb = 2*adc_clip/(2^param.N_qb-1);
4985 5005      +            sigma_Q = adc_lsb/sqrt(12);
4986 5006      +            S_qn = sigma_Q^2/(length(result.S_rn)*delta_f)*ones(size(result.S_rn));
4987 5007      +        end
4988 5008      +        result.adc_clip=adc_clip;
4986 5009          result.S_qn = S_qn;
4987      -        result.qn_rms = sqrt(sum(result.S_qn)* delta_f);
4990 5010      +    result.S_qn_rms = sqrt(sum(result.S_qn)* delta_f);
4988 5011      else
4989      -        result.S_qn=0;
4992 5014      +    result.S_qn = 0;
4990 5015          result.S_qn_rms = 0;
4991 5016          %  result.S_n
4992 5017      end
```

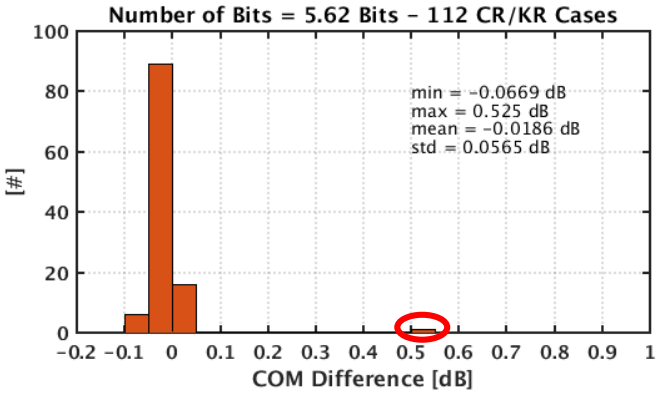
Change 2) “diff”

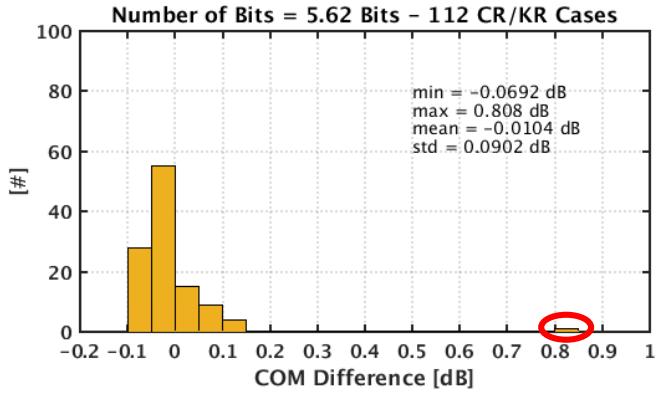
```
8912  8915    param.DER_CDR = xls_parameter(parameter, 'DER_CDR',true,1e-2); % min DER required for a CDR
8913  8916    param.N_qb = xls_parameter(parameter, 'N_qb',true,0); % adc number of bits if 0 do not apply quantization
8914  8917    param.P_qc= xls_parameter(parameter, 'P_qc',true,2*param.specBER); % adc clipping probability
      8918  +  param.clip_method = xls_parameter(parameter, 'Clip Method', false, 'Fast'); % "Clip Method" parameter added by Hossein Shakiba to implement commit request 4p9_1
8915  8919    param.pass_threshold = xls_parameter(parameter, 'COM Pass threshold',false,0); % the pass fail threshold for COM in dB
8916  8920    param.add_rx_noise = xls_parameter(parameter, 'add_rx_noise', true, param.pass_threshold); % additional receiver noise target in dB
8917  8921    param.ERL_pass_threshold = xls_parameter(parameter, 'ERL Pass threshold',false,0); % the pass fail threshold for ERL in dB
```

Test Results and Final Suggestion

- After adding the switch, the same 3x112 test cases were run again and exact same COM difference between two methods was confirmed







- Runtime overheads with two “Fast” and “Slow” methods relative to when quantization noise is disabled demonstrated an almost 2x slower runtime for the “Slow method”
- “Fast” method overhead reduced from 3% to 1% due to additional saving explained in slide 4

Average Runtime Overhead “Fast” Method	Average Runtime Overhead “Slow” Method
1% Overhead	99% Overhead

- It is suggested to proceed with the change and default the “Clip Method” switch to “Fast”

Thank You ☺

Hossein Shakiba
Huawei Technologies Canada