

# COM Commit Request 4p14\_1

Richard Mellitz, Samtec

Adam Gregory, Samtec

Brandon Gore, Samtec

IEEE 802.3 Channel Operating Margin (COM) Open Source Project Ad Hoc

# Table of Contents

- ❑ Purpose
- ❑ Code Updates
- ❑ Summary

# Purpose for Commit Request

## ❑ COM Commit Request 4p14\_1

- SCMR\_CH is supposed to report for the passed channel
- For cable assemblies the added PCB was inadvertently included in the SCMR\_CH calculation
- All other configuration are OK
- This is a bug fix. The SCMR\_CH will be reported for the cable assembly s-parameters irrespective of the HL, HN, or HH configurations used in COM

# Code updates :

- ❑ Commit Request 4p14\_1
- ❑ 2 ../src files updated (see backup for details)
  - COM\_FD\_to\_TD.m
  - FD\_Processing.m
  - Just change the syntax from `chdata(i).scd21_raw` to `chdata(i).scd21_orig`
    - The PCB was included in the raw syntax
    - The passed s-parameters only were represented in the orig syntax
- ❑ No new keywords
  
- ❑ 1 test file added\*
  - `com_ieee8023_4p14p0_CA_SNDR_bug_fix.m`

\* [https://opensource.ieee.org/richard.mellitz/com\\_code/-/raw/S\\_tx\\_bug/release/com\\_ieee8023\\_4p13p0\\_S\\_tx\\_bug.m?ref\\_type=heads&inline=false](https://opensource.ieee.org/richard.mellitz/com_code/-/raw/S_tx_bug/release/com_ieee8023_4p13p0_S_tx_bug.m?ref_type=heads&inline=false)

# Summary

- ❑ SCMR\_CH for CA will be independent of the 200G configurations file
- ❑ No Keywords added.

# Thank You!

backup

# COM\_FD\_to\_TD.m (1)

- ❑ Add unequalized modal FIR for SCMR computations

```
src/COM_FD_to_TD.m Edit View file @ 985f3cbf
... @@ -47,7 +47,7 @@ for i=1:param.number_of_s4p_files
49 % and filtered by fr in FD_Processing.m and also has packages and
    added
50 % boards (if specified in the config sheet)
51 % s21_pkg is where the vtf and tr filter are added
52 - % chdata.sdd21_raw is the passed sdd21 and used for SCMR
53 % for chdata.sdd21_raw tr and fr filter are added
54 % ss =@(a) sum(abs(a(1:length(a))).^2);
55 % prior version did no have filter applied when computing SCMR
... @@ -69,7 +69,19 @@ for i=1:param.number_of_s4p_files
69     chdata(i).t_raw_fltr, ...
70     chdata(i).causality_correction_dB, ...
71     chdata(i).truncation_dB] =
    s21_to_impulse_DC(chdata(i).sdd21_raw.*H_filters ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
72 -     chdata(i).uneq_pulse_response_raw_filtered=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_imp_response_raw_filtered);
...
73 %
74 % check conditions to use channel amplitude and calculate common
... @@ -47,7 +47,7 @@ for i=1:param.number_of_s4p_files
49 % and filtered by fr in FD_Processing.m and also has packages and
    added
50 % boards (if specified in the config sheet)
51 % s21_pkg is where the vtf and tr filter are added
52 + % chdata.sdd21_orig is the passed sdd21 and used for SCMR
53 % for chdata.sdd21_raw tr and fr filter are added
54 % ss =@(a) sum(abs(a(1:length(a))).^2);
55 % prior version did no have filter applied when computing SCMR
... @@ -69,7 +69,19 @@ for i=1:param.number_of_s4p_files
69     chdata(i).t_raw_fltr, ...
70     chdata(i).causality_correction_dB, ...
71     chdata(i).truncation_dB] =
    s21_to_impulse_DC(chdata(i).sdd21_orig.*H_filters ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
72 +     % need unequalize PR of orig for SCMR_CH
    [chdata(i).uneq_imp_response_orig, ...
73 +     chdata(i).t_orig, ...
74 +     chdata(i).causality_correction_dB, ...
75 +     chdata(i).truncation_dB] =
    s21_to_impulse_DC(chdata(i).sdd21_orig ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
76 +     chdata(i).uneq_pulse_response_orig=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_imp_response_orig);
77 +     chdata(i).uneq_pulse_response_orig_filtered=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_pulse_response_orig);
78 +     [chdata(i).uneq_imp_response_orig_filtered, ...
79 +     chdata(i).t_orig_fltr, ...
80 +     chdata(i).causality_correction_dB, ...
81 +     chdata(i).truncation_dB] =
    s21_to_impulse_DC(chdata(i).sdd21_orig.*H_filters ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
82 +     chdata(i).uneq_pulse_response_orig_filtered=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_imp_response_orig_filtered);
83 +
84 +
85 %
86 % check conditions to use channel amplitude and calculate common
```

# COM\_FD\_to\_TD.m (2)

- ❑ Remove old comments lines
- ❑ Replace raw with orig

```
src/COM_FD_to_TD.m
74 % check conditions to use channel amplitude and calculate common
    to differential
75 USE_channel_amplitude = 1;
... @@ -82,35 +94,21 @@ for i=1:param.number_of_s4p_files
82 if USE_channel_amplitude
83
    chdata(i).uneq_imp_response=chdata(i).uneq_imp_response*chdata(i).A;
    % adjust IRx for amplitude
84 end
85 - % Common mode responses
86 - % DC
87 - % [chdata(i).uneq_dc_imp_response, ...
88 - %   chdata(i).t_DC, ...
89 - %   chdata(i).causality_correction_DC_dB, ...
90 - %   chdata(i).truncation__DC_dB] =
    s21_to_impulse_DC(chdata(i).sdc21 ,chdata(i).faxis, param.sample_dt,
    OP,param) ; % not used
91 - % chdata(i).uneq_pulse_DC_response_raw=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_dc_imp_response); % not used
92 - % CD is using raw passed s-parameters
93 - % [chdata(i).uneq_CD_imp_response, ...
94 - %   chdata(i).t_CD, ...
95 - %   chdata(i).causality_correction_CD_dB, ...
96 - %   chdata(i).truncation__CD_dB] =
    s21_to_impulse_DC(chdata(i).scd21_raw ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
97 - % chdata(i).uneq_pulse_CD_response_raw=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_CD_imp_response);
98 - % DC
99 [chdata(i).uneq_CD_imp_response_filtered, ...
100 chdata(i).t_CD_fltr, ...
101 chdata(i).causality_correction_CD_dB, ...
102 - chdata(i).truncation__CD_dB] =
    s21_to_impulse_DC(chdata(i).scd21_raw.*H_filters ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
103 chdata(i).uneq_pulse_CD_response_filtered=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_CD_imp_response_filtered);
104 % DC is using raw passed s-parameters

86 % check conditions to use channel amplitude and calculate common
    to differential
87 USE_channel_amplitude = 1;
... @@ -82,35 +94,21 @@ for i=1:param.number_of_s4p_files
94 if USE_channel_amplitude
95
    chdata(i).uneq_imp_response=chdata(i).uneq_imp_response*chdata(i).A;
    % adjust IRx for amplitude
96 end
97 [chdata(i).uneq_CD_imp_response_filtered, ...
98 chdata(i).t_CD_fltr, ...
99 chdata(i).causality_correction_CD_dB, ...
100 + chdata(i).truncation__CD_dB] =
    s21_to_impulse_DC(chdata(i).scd21_orig.*H_filters ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
101 chdata(i).uneq_pulse_CD_response_filtered=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_CD_imp_response_filtered);
102 % DC is using raw passed s-parameters
```

# COM\_FD\_to\_TD.m (3)

- ❑ Replace raw with orig

```
src/COM_FD_to_TD.m
Edit View file @ 985f3cbf

107 % chdata(i).causality_correction_DC_dB, ...
108 - % chdata(i).truncation__DC_dB] =
    s21_to_impulse_DC(chdata(i).sdc21_raw ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
109 % chdata(i).uneq_pulse_DC_response_raw=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_DC_imp_response);
110 [chdata(i).uneq_DC_imp_response_filtered, ...
111     chdata(i).t_DC_fldr, ...
112     chdata(i).causality_correction_DC_dB, ...
113 -     chdata(i).truncation__DC_dB] =
    s21_to_impulse_DC(chdata(i).sdc21_raw.*H_filters ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
114     chdata(i).uneq_pulse_DC_response_filtered=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_DC_imp_response_filtered);
115 %-----
116 % next find Pulse response (SBR) for each channel h^(k)(t)
... @@ -139,14 +137,14 @@ for i=1:param.number_of_s4p_files
139 chdata(i).VCM_CD_HF_struct=
    get_cm_noise(param.samples_per_ui,chdata(i).uneq_pulse_CD_response_fil
    tered,param.levels,param.P_peak,OP); % returns voltages % changed for
    d 2.2 from param.P_peak
140 chdata(i).VCM_DC_HF_struct=
    get_cm_noise(param.samples_per_ui,chdata(i).uneq_pulse_DC_response_fil
    tered,param.levels,param.P_peak,OP); % returns voltages % changed for
    d 2.2 from param.P_peak
141 % find the peak of the uneq_pulse_response_raw
142 - PR_RAW_fldr=chdata(i).uneq_pulse_response_raw_filtered; % the
    signal is defined only for the through channel
143 - ipeak=find(PR_RAW_fldr==max(PR_RAW_fldr),1,'first'); %#ok<NASGU>
144 - V_peak=PR_RAW_fldr(ipeak);
145  istart=mod(ipeak-1,M)+1;
146 - iend=floor((length(PR_RAW_fldr)/M))*M;
147 - PR_RAW_fldr_sampled=PR_RAW_fldr(istart:M:iend);
148 - P_signal=norm(PR_RAW_fldr_sampled)^2;
149 - sigma_ts=norm(PR_RAW_fldr_sampled);
150     chdata(i).P_signal=P_signal; % all files use the same P_signal
151
    chdata(i).SCMR_CD_ch_pk=10*log10(V_peak^2/chdata(i).VCM_CD_HF_struct.C
    Mn^2);
152
    chdata(i).SCMR_CD_ch=10*log10(P_signal/chdata(i).VCM_CD_HF_struct.CMn^
    2); % power after filters
...

105 % chdata(i).causality_correction_DC_dB, ...
106 + % chdata(i).truncation__DC_dB] =
    s21_to_impulse_DC(chdata(i).sdc21_orig ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
107 % chdata(i).uneq_pulse_DC_response_raw=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_DC_imp_response);
108 [chdata(i).uneq_DC_imp_response_filtered, ...
109     chdata(i).t_DC_fldr, ...
110     chdata(i).causality_correction_DC_dB, ...
111 +     chdata(i).truncation__DC_dB] =
    s21_to_impulse_DC(chdata(i).sdc21_orig.*H_filters ,chdata(i).faxis,
    param.sample_dt, OP,param) ; % just passed s-parameters
112     chdata(i).uneq_pulse_DC_response_filtered=filter(ones(1,
    param.samples_per_ui), 1, chdata(i).uneq_DC_imp_response_filtered);
113 %-----
114 % next find Pulse response (SBR) for each channel h^(k)(t)
... @@ -139,14 +137,14 @@ for i=1:param.number_of_s4p_files
137 chdata(i).VCM_CD_HF_struct=
    get_cm_noise(param.samples_per_ui,chdata(i).uneq_pulse_CD_response_fil
    tered,param.levels,param.P_peak,OP); % returns voltages % changed for
    d 2.2 from param.P_peak
138 chdata(i).VCM_DC_HF_struct=
    get_cm_noise(param.samples_per_ui,chdata(i).uneq_pulse_DC_response_fil
    tered,param.levels,param.P_peak,OP); % returns voltages % changed for
    d 2.2 from param.P_peak
139 % find the peak of the uneq_pulse_response_raw
140 + PR_ORIG_fldr=chdata(i).uneq_pulse_response_orig_filtered; % the
    signal is defined only for the through channel
141 + ipeak=find(PR_ORIG_fldr==max(PR_ORIG_fldr),1,'first'); %#ok<NASGU>
142 + V_peak=PR_ORIG_fldr(ipeak);
143  istart=mod(ipeak-1,M)+1;
144 + iend=floor((length(PR_ORIG_fldr)/M))*M;
145 + PR_ORIG_fldr_sampled=PR_ORIG_fldr(istart:M:iend);
146 + P_signal=norm(PR_ORIG_fldr_sampled)^2;
147 + sigma_ts=norm(PR_ORIG_fldr_sampled);
148     chdata(i).P_signal=P_signal; % all files use the same P_signal
149
    chdata(i).SCMR_CD_ch_pk=10*log10(V_peak^2/chdata(i).VCM_CD_HF_struct.C
    Mn^2);
150
    chdata(i).SCMR_CD_ch=10*log10(P_signal/chdata(i).VCM_CD_HF_struct.CMn^
    2); % power after filters
...
```

