

SNDR (REF) commit request 4p8p_7

Richard Mellitz, Samtec

IEEE 802.3 Channel Operating Margin (COM) Open Source Project Ad Hoc,
New Orleans, LA May 2025 Interim

Purpose for Commit Request, SNDR_REF

- ❑ Reference: mellitz_3dj_02_2503
 - [Fork Namespace: Richard Mellitz / com_code](#)
 - [Branch: SNDR_REF](#)
- ❑ Implement calculations in section 179.9.4.5.3 Reference SNDR
- ❑ Added subfunction file: SNDR_Ref.m
- ❑ Changes subfunction files:
 - [com_ieee8023_.m](#)
 - [read_ParamConfigFile.m](#)

SNDR_ref

- ❑ Normally the present would be defined in the configuration file.

```
src/SNDR_ref.m 0 → 100644 +71 -0 View file @ 58a81f68
1 + function [results] = SNDR_ref(PR_Ref,param)
2 + %% License Notice
3 + %
4 + % Copyright 2025 802-COM Authors
5 + %
6 + % Redistribution and use in source and binary forms, with or without
7 + % modification, are permitted provided that the following conditions are
8 + % met:
9 + %
10 + % - Redistributions of source code must retain the above copyright
11 + % notice, this list of conditions and the following disclaimer.
12 + %
13 + % - Redistributions in binary form must reproduce the above copyright
14 + % notice, this list of conditions and the following disclaimer in the
15 + % documentation and/or other materials provided with the distribution.
16 + %
17 + % - Neither the name of the copyright holder nor the names of its
18 + % contributors may be used to endorse or promote products derived from
19 + % this software without specific prior written permission.
20 + %
21 + % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
22 + % "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
23 + % LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
24 + % A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
25 + % HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
26 + % SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
27 + % LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
28 + % DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
29 + % THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
30 + % (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
31 + % OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
32 + %
33 + % SPDX-License-Identifier: BSD-3-Clause
34 + if ~isfield(param,'preset')
35 +     param.preset(1).txffe= [0      0      0      1      0 ];
36 +     param.preset(2).txffe= [0      0      0      0.5  0];
37 +     param.preset(3).txffe= [0      0     -0.075  0.75  0];
38 +     param.preset(4).txffe= [0      0.05  -0.20  0.75  0];
39 +     param.preset(5).txffe= [-0.025 0.075 -0.25  0.65  0];
40 +     param.preset(6).txffe= [0      0      0      0.75  0];
```

SNDR_ref (Cont'd)

- ❑ Implements slide 6 of mellitz_3dj_02_2503
- ❑ $SNDR^{(ref)}(c_i) = 10\log_{10}(\sigma_{ts}^2 / \sigma_{tn}^2)$
 - With respective variable names `sigma_ts_sqrd` and `sigma_tn_sqrd`
- ❑ Results include the vector `SNDR_ref` and `sigma_iL`

```
src/SNDR_ref.m 0 → 100644 +71 -0 View file @ 58a81f68

41 + end
42 +
43 + ss =@(a) sum(abs(a(1:length(a))).^2);
44 + SNR_TX=param.SNR(1);
45 + M=param.samples_per_ui;
46 + PR_noFFE=PR_Ref;
47 + ipeak=find(PR_noFFE==max(PR_noFFE),1,'fir');
48 + istart=mod(ipeak,M)+1;
49 + iend=floor((length(PR_noFFE)/M)*M);
50 + PR_noFFE_sampled=PR_noFFE(istart:M:iend);
51 + for ipst=1:length(param.preset)
52 +     sigma_tn_sqrd=ss(PR_noFFE_sampled);
53 +     PR_FFE=FFE( param.preset(ipst).txffe , 3, M, PR_noFFE );
54 +     ipeak=find(PR_FFE==max(PR_FFE),1,'fir');
55 +     hss= PR_FFE(-param.D_p*M+ipeak:M:param.N_p*M+ipeak);
56 +     numUI=floor(length(PR_FFE)/M)-1;
57 +     sigma_iL(ipst)=sqrt(ss(hss));
58 +     sigma_ts_sqrd(ipst)=ss(hss)*10^(SNR_TX/10);
59 +     sigma_tn_sqrd(ipst)=ss(PR_noFFE_sampled);
60 +     rat(ipst)=10*log10(ss(hss)/ss(PR_noFFE_sampled));
61 +     SNDR_ref(ipst)=10*log10(sigma_ts_sqrd(ipst)/sigma_tn_sqrd(ipst));
62 + end
63 + results.SNDR_ref=SNDR_ref;
64 + results.SNDR_ref_p1=SNDR_ref(1);
65 + results.SNDR_ref_p2=SNDR_ref(2);
66 + results.SNDR_ref_p3=SNDR_ref(3);
67 + results.SNDR_ref_p4=SNDR_ref(4);
68 + results.SNDR_ref_p5=SNDR_ref(5);
69 + results.SNDR_ref_p6=SNDR_ref(6);
70 + results.sigma_iL=sigma_iL(1);
71 + end

\ No newline at end of file
```

com_ieee8023_

- ❑ Flag for OP.SNDR_REF invokes computation of SNDR_REF in the subfunction SNDR_REF
- ❑ Addition of code to output arguments for SNDR_REF

```
src/com_ieee8023_m @@ -694,7 +694,11 @@ while (OP.RX_CALIBRATION==1 || DO_ONCE==true)
... 694 end
695 [msg] = end_display_control(msg,param,OP,output_args,COM,min_ERL,ERL,
VE0_mV,VEC_dB,threshold_DER,OP.DISPLAY_WINDOW); % {} forces no ERL print
696
697 -
697 + % only for SNDR_REF
698 + if OP.SNDR_REF
699 +     SNDR_REF_results=SNDR_ref(chdata(1).uneq_pulse_response,param);
700 +     output_args.SNDR_ref=SNDR_REF_results.SNDR_ref;
701 + end
698 702 %% Output Args
699 703 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
700 704 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
... 728 732 fprintf(' Die to die loss = %g dB \n',output_args.IL_db_die_to_die_at_Fnq)
729 733 fprintf('run time = %g min \n',output_args.rtmin)
730 734 end
731 -
735 + if OP.SNDR_REF
736 +     fprintf(' SNDR ref s = [')
737 +     for ii=1:length(output_args.SNDR_ref)
738 +         fprintf(' %g ',output_args.SNDR_ref(ii))
739 +     end
740 +     fprintf(' ] db \n')
741 + end
732 742 if nargout==0
733 743     fprintf('<strong>--- Testcase %d results ---</strong>\n', package_testcase_i);
734 744     disp(output_args)
... 745
```

read_ParamConfigFile

- ❑ D_p and N_p are required for the SNDR_REF calculation
- ❑ They are defaulted respectively to 4 and 400.
 - Can be set in the configuration file as “D_p” and “N_p”.
- ❑ Flag for SNDR_REF computation define as OP.SNDR_REF
- ❑ “preset_txffe” read from configuration file as row of preset with values in respective column

```
src/read_ParamConfigFile.m
...    @@ -158,7 +158,8 @@ param.tx_ffe_cp3_values = xls_parameter(parameter, 'c(3)', true,0); % TX equaliz
158    158    end
159    159    param.ndfe = xls_parameter(parameter, 'N_b'); % Decision feedback fixed equalizer (DFE) length
160    160    param.N_v = xls_parameter(parameter, 'N_v',true,param.ndfe); % number of UI used to compute Vf
161    161    - param.D_p = xls_parameter(parameter, 'D_p',true, 4 ); % number of precursor UI's used to compute Vf Default to 10
162    162    + param.D_p = xls_parameter(parameter, 'D_p',true, 4 ); % number of precursor UI's used to compute Vf Default to 4
163    163    + param.N_p = xls_parameter(parameter, 'N_p',true, 400 ); % number of precursor UI's used to compute Vf Default to 400
164    164    param.N_bx = xls_parameter(parameter, 'N_bx', true, param.ndfe ); % Used for ERL to Compensate for a number of Ui associated
165    165    % support for floating taps
166    166    param.N_bg=xls_parameter(parameter, 'N_bg', true,0); % number of group of floating tap. Used as a switch, 0 means no float
...    @@ -596,6 +597,11 @@ if OP.SAVE_CONFIG2MAT || OP.CONFIG2MAT_ONLY
596    597    save(matconfile, 'parameter');
597    598    end
598    599
600    600    + OP.SNDR_REF= xls_parameter(parameter, 'SNDR_REF',false,0); % controls whether SNDR_REF is computed. Heed special config sheet
601    601    + preset_txffe=xls_parameter(parameter, 'preset_txffe',true,[]); % controls whether SNDR_REF is computed. Heed special config
602    602    + if ~isempty(preset_txffe)
603    603    +     for ii=1:length(preset_txffe),param.preset(ii).txffe=preset_txffe(ii,:);end
604    604    + end
599    605
600    606    %% At the very end of Parameter reading, swap in the proper Tx and Rx values for package parameters based on pkg name
601    607    if ~isempty(param.PKG_NAME)
...    ...
```

Example Configuration Spreadsheet Clip 1

Table 93A-1 parameters			
Parameter	Setting	Units	Information
f_b	106.25	GBd	
f_min	0.05	GHz	
Delta_f	0.01	GHz	
R_0	50	Ohm	
PKG_NAME	PKG_HiR_CLASSB no_Rx		TX RX
z_p select	[2]		
L	4		
M	32		
filter and Eq			
f_r	0.55	*fb	58.4375 GHz
c(0)	0.55		[min:step:max]
c(-1)	0	[-0.34:0.02:0]	see if 0 is OK
c(-2)	0	[0:0.02:0.14]	see if 0 is OK
c(1)	0	[-0.2:0.02:0]	see if 0 is OK
N_b	1	UI	As/dfe1
b_max(1)	1		As/dfe2..N_b
b_max(2..N_b)	1		NA
b_min(1)	0		As/dfe2..N_b
b_min(2..N_b)	0		NA
g_DC	0	[-20:1:0]	see if 0 is OK
f_z	1000000000.00	GHz	
f_p1	1000000000.00	GHz	
f_p2	1000000000.00	GHz	[min:step:max]
g_DC_HP	0		see if 0 is OK
f_HP_PZ	1.00E-09	GHz	
Butterworth	0		
Bessel_Thomson	1		SNDR uses BT
C_d	[0.4e-4 0.9e-4 1.1e-4 ; 0 0 0]	nF	[TX RX]
L_s	[0.13 0.15 0.14; 0 0 0]	nH	[TX RX]
C_b	[0.3e-4 0]	nF	[TX RX]

I/O control		
DIAGNOSTICS	1	logical
DISPLAY_WINDOW	1	logical
CSV_REPORT	0	logical
RESULT_DIR	.\results\SNR_MFEXT_{date}\	
SAVE_FIGURES	0	logical
Port Order	[1 2 3 4]	[1 3 2 4]
RUNTAG	SNR_MDFEXT	[1 2 3 4]
COM_CONTRIBUTION	0	logical
Bread_crumbs	0	
TDR and ERL options		
TDR	1	logical
ERL	1	logical
ERL_ONLY	0	ns
TR_TDR	0.005	
N	1600	logical
TDR_Butterworth	1	
beta_x	0	
rho_x	0.618	
TDR_W_TXPKG	1	UI
N_bx	0	
fixture delay time	[0 0]	
Tukey_Window	1	
Z_t	46.25	ohm
Noise, jitter		UI
sigma_RJ	0	UI
A_DD	0	
eta_0	1.00E-100	V^2/GHz
SNR_TX	33.5	dB
R_LM	1	

Example Configuration Spreadsheet Clip 2

Floating Tap Control			SAVE_CONFIG2MAT	0	
N_bg	0	0 1 2 or 3 groups	Receiver testing		
N_bf	4	taps per group	RX_CALIBRATION	0	logical
N_f	50	UI span for floating taps	Sigma BBN step	5.00E-03	V
bmaxg	0.05	max DFE value for floating taps	ICN amd FOM ILD parameters		
B_float_RSS_MAX	0.1	rss tail tap limit (not used)	f_v	0.524	55.64705882
N_tail_start	101	(UI) start of tail taps limit	f_f	0.524	55.64705882
Filter: Rx FFE			f_n	0.524	55.64705882
ffe_pre_tap_len	5	UI	f_2	67.000	GHz
ffe_post_tap_len	100	UI	f_1	0.010	GHz
ffe_main_cursor_min	1		A_ft	0.600	V
ffe_pre_tap1_max	0.7		A_nt	0.600	V
ffe_post_tap1_max	0.7		Parameter Setting		
ffe_tapn_max	0.7		board_tl_gamma0_a1_a2	[0 5.95e-4 2.6e-05]	1.4 db/in @ 53.125G
Operational			board_tl_tau	5.790E-03	ns/mm
ERL Pass threshold	7.3	dB	board_Z_c	92.5	Ohm
COM Pass threshold	3	db	z_bp (TX)	250	mm
DER_0	2.00E-05		z_bp (NEXT)	0	mm
T_r	0.00400	ns	z_bp (FEXT)	250	mm
FORCE_TR	1	logical	z_bp (RX)	0	mm
PMD_type	C2C		C_0	[0 0]	nF
EW	0		C_1	[0 0]	nF
MLSD	0	logical	Include PCB		
ts_anchor	1		Impulse response truncation threshold	0.0001	need for to increase percision
sample_adjustment	0		Force PDF bin size	0.00001	
Local Search	0	set to 2 for speed up			
D_p	4				
N_p	400				
N_v	400				
sndr_Ref	1				
preset_txffe	[0 0 0 1 0 ; 0 0 0 0.5 0 ; 0 0 -0.075 0.75 0 ; 0 0.05 -0.20 0.75 0 ; -0.025 0.075 -0.25 0.65 0 ; 0 0 0.7 0]				

I/O Changes

- ❑ Changes to config

- Keyword D_p and N_p can be specified but default to SNDR_REF
- 4 and 400 respectively
- Keyword SNDR_REF
 - Set to 0 disables the SNDR_REF computation
 - Set to 1 enables the SNDR_REF computation
- Keyword preset_txffe [{preset 1}; {preset 2}; {preset 3};... {preset n}]
 - Example:

sndr_Ref	1
preset_txffe	[0 0 0 1 0 ; 0 0 0 0.5 0; 0 0 -0.075 0.75 0; 0 0.05 -0.20 0.75 0;-0.025 0.075 -0.25 0.65 0;0 0 0 0.7 0]

☐ Usage requires adopting the configuration spreadsheet to applicable Section or Annex.

- ❑ Changes to output

- `output_args.SNDR_ref` ... 33.4990 27.4784 30.3223 29.4339 27.4648 30.4010

❑ Usage note: Please ignore the reported COM values as are only looking at the SNDR_REF output.

 Download beta test code

- [https://opensource.ieee.org/richard.mellitz/com_code/-/raw/SNDR Ref/release/com_ieee8023_4p9p0_beta/SNDR Ref 01.m?ref_type=heads&inline=false](https://opensource.ieee.org/richard.mellitz/com_code/-/raw/SNDR%20Ref/release/com_ieee8023_4p9p0_beta/SNDR%20Ref%2001.m?ref_type=heads&inline=false)

Thank You!