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_2505
  - 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

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

SNDR_ref (Cont'd)

- Implements slide 6 of mellitz_3dj_02_2505
- $SNDR^{(ref)}(c_i) = 10log_{10}(\sigma_{ts}^2/\sigma_{tn}^2)$
  - 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

```
+ end
+
+ ss =@(a) sum(abs(a(1:length(a))).^2);
+ SNR_TX=param.SNDR(1);
+ M=param.samples_per_ui;
+ PR_noFFE=PR_Ref;
+ ipeak=find(PR_noFFE==max(PR_noFFE),1,'fir');
+ istart=mod(ipeak,M)+1;
+ iend=floor((length(PR_noFFE)/M))*M;
+ PR_noFFE_sampled=PR_noFFE(istart:M:iend);
+ for ipst=1:length(param.preset)
+     sigma_tn_sqrd=ss(PR_noFFE_sampled);
+     PR_FFE=FFE( param.preset(ipst).txffe  , 3, M, PR_noFFE );
+     ipeak=find(PR_FFE==max(PR_FFE),1,'fir');
+     hss= PR_FFE(-param.D_p*M+ipeak:M:param.N_p*M+ipeak);
+     numUI=floor(length(PR_FFE)/M)-1;
+     sigma_iL(ipst)=sqrt(ss(hss));
+     sigma_ts_sqrd(ipst)=ss(hss)*10^(SNR_TX/10);
+     sigma_tn_sqrd(ipst)=ss(PR_noFFE_sampled);
+     rat(ipst)=10*log10(ss(hss)/ss(PR_noFFE_sampled));
+     SNDR_ref(ipst)=10*log10(sigma_ts_sqrd(ipst)/sigma_tn_sqrd(ipst));
+ end
+ results.SNDR_ref=SNDR_ref;
+ results.SNDR_ref_p1=SNDR_ref(1);
+ results.SNDR_ref_p2=SNDR_ref(2);
+ results.SNDR_ref_p3=SNDR_ref(3);
+ results.SNDR_ref_p4=SNDR_ref(4);
+ results.SNDR_ref_p5=SNDR_ref(5);
+ results.SNDR_ref_p6=SNDR_ref(6);
+ results.sigma_iL=sigma_iL(1);
+ 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 +12 −2 View file @ 58a81f68

```
@@ -694,7 +694,11 @@ while (OP.RX_CALIBRATION==1 || DO_ONCE==true)
          end
          [msg] = end_display_control(msg,param,OP,output_args,COM,min_ERL,ERL,
VEO_mV,VEC_dB,threshold_DER,OP.DISPLAY_WINDOW); % {} forces no ERL print

-
+         % only for SNDR_REF
+         if OP.SNDR_REF
+             SNDR_REF_results=SNDR_ref(chdata(1).uneq_pulse_response,param);
+             output_args.SNDR_ref=SNDR_REF_results.SNDR_ref;
+         end
          %% Output Args
          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
@@ -728,7 +732,13 @@ while (OP.RX_CALIBRATION==1 || DO_ONCE==true)
              fprintf(' Die to die loss = %g dB \n',output_args.IL_db_die_to_die_at_Fnq)
              fprintf('run time = %g min \n',output_args.rtmin)
          end
-
+        if  OP.SNDR_REF
+            fprintf(' SNDR ref s = [')
+            for ii=1:length(output_args.SNDR_ref)
+             fprintf(' %g ',output_args.SNDR_ref(ii))
+            end
+            fprintf(' ] db \n')
+        end
          if nargout==0
              fprintf('<strong>--- Testcase %d results ---</strong>\n', package_testcase_i);
              disp(output_args)
```

read_ParamConfigFile

- D_p and N_p are required for the SNDR_REF calculation
- The 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       - param.D_p = xls_parameter(parameter, 'D_p',true, 4 ); % number of precursor UI's used to compute Vf Default to 10
     161  + param.D_p = xls_parameter(parameter, 'D_p',true, 4 ); % number of precursor UI's used to compute Vf Default to 4
     162  + param.N_p = xls_parameter(parameter, 'N_p',true, 400 ); % number of precursor UI's used to compute Vf Default to  400
162  163  param.N_bx = xls_parameter(parameter, 'N_bx', true, param.ndfe ); % Used for ERL to Compensate for a number of Ui assoicated
163  164  % support for floating taps
164  165  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(matcongfile ,'parameter');
597  598  end
598  599
     600  + OP.SNDR_REF= xls_parameter(parameter,'SNDR_REF',false,0); % controls whether SNDR_REF is computeted. Heed special config sheet
     601  + preset_txffe=xls_parameter(parameter,'preset_txffe',true,[]); % controls whether SNDR_REF is computeted. Heed special config
     602  + if ~isempty(preset_txffe)
     603  +     for ii=1:length(preset_txffe),param.preset(ii).txffe=preset_txffe(ii,:);end
     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/dffe1
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			
N_bg	**0**	0 1 2 or 3 groups	
N_bf	4	taps per group	
N_f	50	UI span for floating taps	
bmaxg	0.05	max DFE value for floating taps	
B_float_RSS_MAX	0.1	rss tail tap limit (not used)	
N_tail_start	101	(UI) start of tail taps limit	
Filter: Rx FFE			
ffe_pre_tap_len	**5**	UI	
ffe_post_tap_len	100	UI	
ffe_main_cursor_min	1		
ffe_pre_tap1_max	0.7		
ffe_post_tap1_max	0.7		
ffe_tapn_max	0.7		
Operational			
ERL Pass threshold	7.3	dB	
COM Pass threshold	3	db	
DER_0	2.00E-05		
T_r	0.00400	ns	
FORCE_TR	1	logical	
PMD_type	C2C		
EW	0		
MLSD	0	logical	
ts_anchor	1		
sample_adjustment	0		
Local Search	0	set to 2 for speed up	

SAVE_CONFIG2MAT	0	
Receiver testing		
RX_CALIBRATION	0	logical
Sigma BBN step	5.00E-03	V
ICN amd FOM ILD parameters		
f_v	0.524	55.64705882
f_f	0.524	55.64705882
f_n	0.524	55.64705882
f_2	67.000	GHz
f_1	0.010	GHz
A_ft	0.600	V
A_nt	0.600	V
Parameter	**Setting**	
board_tl_gamma0_a1_a2	*[0 5.95e-4 2.6e-05]*	1.4 db/in @ 53.125G
board_tl_tau	*5.790E-03*	ns/mm
board_Z_c	92.5	Ohm
z_bp (TX)	*250*	mm
z_bp (NEXT)	*0*	mm
z_bp (FEXT)	*250*	mm
z_bp (RX)	*0*	mm
C_0	[0 0]	nF
C_1	[0 0]	nF
Include PCB	**1**	**logical**

Impulse response truncation threshold	0.0001	need for to increase percision
Force PDF bin size	0.00001	

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 0.7 0]**	

I/O Changes

- Changes to config
  - Keywork 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

Thank You!