COM Implementation example of receiver interference tolerance specified in 178A.1.7.5 (RiTol) Commit Request 4p9_5

Richard Mellitz, Samtec

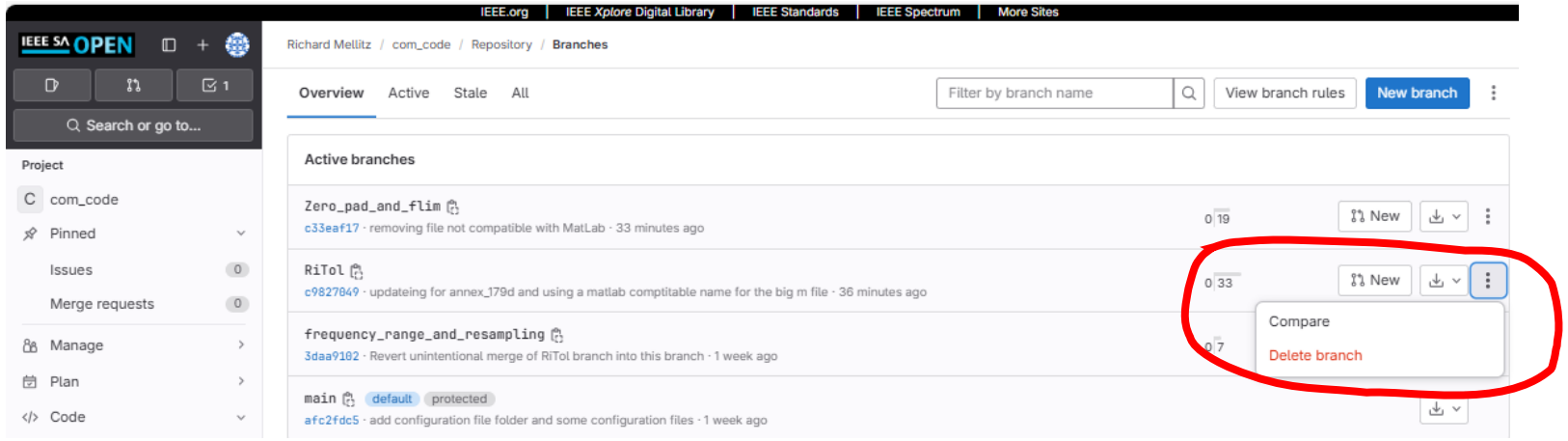
Adam Gregory, Samtec

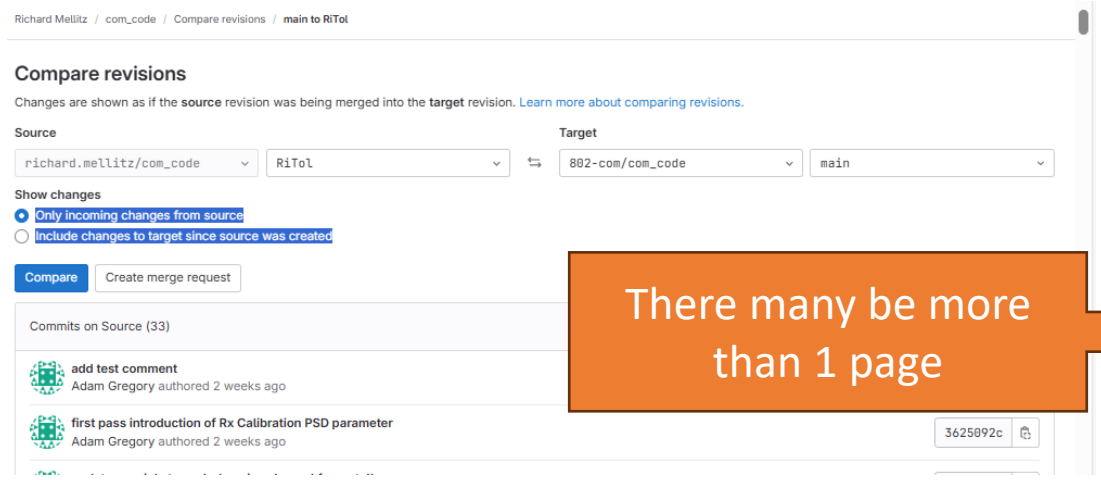
IEEE 802.3 Channel Operating Margin (COM) Open Source Project Ad Hoc,
New Orleans, LA May 2025 Interim

Purpose for Commit Request

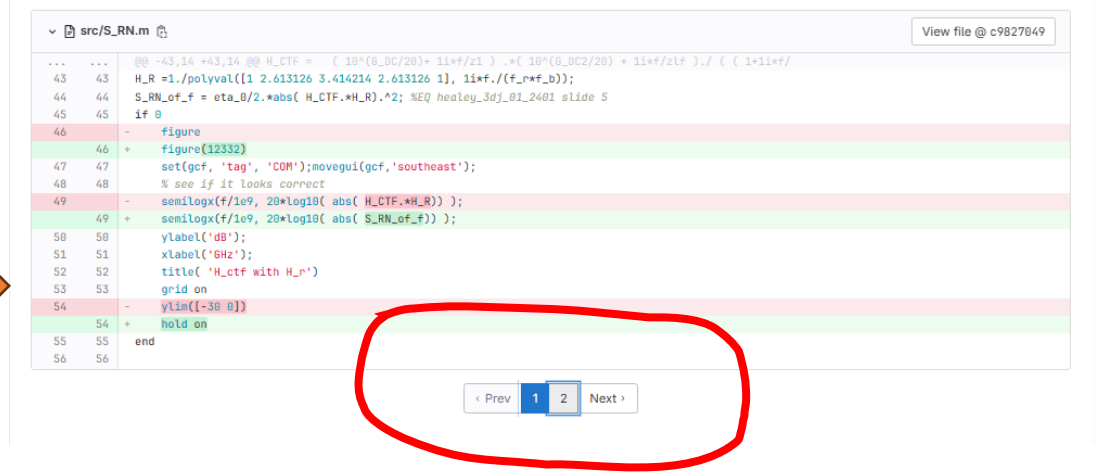
- ❑ COM Implementation example of receiver interference tolerance specified in 178A.1.7.5 (RiTol)
- ❑ Limit configuration templates provided
  - Values and settings in the spread sheet may not be aligned with a standard or ongoing work. Please check before using.
- ❑ Branch: RiTOL
- ❑ Branch test code (will disappear after merge)
  - https://opensource.ieee.org/richard.mellitz/com_code/-/raw/RiTol/release/com_ieee8023_4p9p1_RiTOl.m?ref_type=heads&inline=false

Checking changes of a branch to the last release





There many be more than 1 page



Operation

- The keyword RX_calibration and usages with RxFFE turns on the calibration operations.
  - Prior version without RxFFE operate as in pervious versions
- Injected noise power spectral density included in get_PDSs
- 2 new functions
  - S_in
    - Create injected noise PSD
  - N_s
    - Creates noise source
- Changes to 17 other functions to modify dialog, CLI, and looping..
- Operation is similar to prior versions.
  - i.e. looping until 3 dB COM is reached.
- The noise sigma BBN step is a starting step size as before
- The change is that crosstalk channels can be consider and entered and an added channel may be cascaded for running CR COM

N_s.m

- Creates noise source to be used in S_in

src/N_s.m 0 → 100644 View file @ c9827049

```
+ % 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
+ %%
+ % computes Ns(f) PSD on rx compliance injected noise base on sigma_ns
+ % Ns_of_f the single-sided power spectral density at the noise generator output in V^2/Hz
+ % hence the incomming f is > 0
+ f_b=param.fb; %align variable with standard varaible usage
+ f_hp=param.f_hp; %align variable with standard varaible usage
+ inq=find( f<=f_b/2,1,'last');
+ RIT_REF_PTR=lower(OP.RIT_REF_PTR);
+ Ns_of_f=zeros(1,length(f));
+ switch RIT_REF_PTR
+     case 'clause_178'
+         Ns_of_f(1:inq)=ones(1,inq)*2*sigma_ns^2/f_b;% descrived in 178.9.3.4.2 d1.5
+     case 'clause_179'
+         if f_hp <= 0
+             error('Parameter f_hp must be greater than 0 when using Clause 179');
+         end
+         beta=1-(2*f_hp/f_b)*atan(f_b/(2*f_hp));
+         Ns_of_f(1:inq)=(2*sigma_ns^2)/(beta*f_b)* (f(1:inq)./f_hp).^2 ./ (1+(f(1:inq)/f_hp).^2);% equation 179-23 d1.5
+     case 'annex_176d'
+         if f_hp <= 0
+             error('Parameter f_hp must be greater than 0 when using Clause 179');
+         end
+         beta=1-(2*f_hp/f_b)*atan(f_b/(2*f_hp));
+         Ns_of_f(1:inq)=(2*sigma_ns^2)/(beta*f_b)* (f(1:inq)./f_hp).^2 ./ (1+(f(1:inq)/f_hp).^2);% equation 179-23 d1.5
+     otherwise
+         error('unsuported RIT Reference Pointer "%s" ', RIT_REF_PTR);
+ end
+ if 0
+     figure(99)
+     set(gcf, 'tag', 'COM');movegui(gcf,'southeast');
+     % see if it looks correct
+     plot(f/1e9, Ns_of_f) ;
+     ylabel('pwr');
+     xlabel('GHz');
+     title( 'N_s')
+     grid on
+     hold on
+
+ end
```

S_IN

- Created interference nosie PSD

src/S_IN.m 0 → 100644 View file @ c9827049

```
+ % 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
+ %%
+ % computes PSD on rx compliance injected noise
+ Hn21=noise_path;% as function of f. the VTF of the passed noise channel with rx package added.
+ p1=param.CTLE_fp1(1);
+ z1=param.CTLE_fz(1);
+ p2=param.CTLE_fp2(1);
+ zlf=param.f_HP(1);
+ plf=param.f_HP(1);
+ f_b=param.fb;
+ f_r=param.f_r;
+ sigma_ns=param.sigma_ns+eps(0);
+ H_CTF =   ( 10^(G_DC/20)+ 1i*f/z1 ) .*( 10^(G_DC2/20) + 1i*f/zlf )./ ( ( 1+1i*f/p1) .* ( 1+1i*f/p2)  .* ( 1+1i*f/plf));
+ H_R =1./polyval([1 2.613126 3.414214 2.613126 1], 1i*f./(f_r*f_b));
+ Ns_of_f = N_s(f,param, sigma_ns,OP);
+ S_IN_of_f = Ns_of_f./2.*abs( Hn21.*H_CTF.*H_R).^2; %
+ if 0
+     delta_f=f(2)-f(1);
+     S_IN_of_f_rms=sqrt(sum(S_IN_of_f)* delta_f);
+     figure(12332)
+     set(gcf, 'tag', 'COM');movegui(gcf,'southeast');
+     % see if it looks correct
+     semilogx(f/1e9, 20*log10( abs( S_IN_of_f)) );
+     ylabel('dB');
+     xlabel('GHz');
+     title( 'S_IN_of_f including H_ctf with H_r')
+     grid on
+     hold off
+
+ end
+
```

get_PSDs

- Adding S_IN to noise PSD
- When not performing an Rx Calibration S_IN is zero.

src/get_PSDs.m — View file @ c9827649

```
@@ -78,9 +78,33 @@ if OP.NO_TXFFE % to speed up loop find sn only first time ctle is case
         S_rn= [real( S_rn(1)),  S_rn(2:end-1), real( S_rn(end)), conj( S_rn(end-1:-1:2))];
         result.S_rn=S_rn;
         result.S_rn_rms = sqrt(sum(result.S_rn)* delta_f);
+
+        %% compute S_in ( eq 178A-24 d0.2 )
+        % S_in = 0 when Rx Calibration is not performed
+        if OP.PSDRXCAL
+            % interpolate to fvec
+            result.H_noise=H_interp(chdata(end).sdd21p,chdata(1).faxis,fvec,f_b);
+            % Units are V^2/Hz
+            S_IN_of_f=S_IN(fvec,result.H_noise,G_DC,G_DC2,param,OP);
+            % Convert single-sided frequency response to conjugate-symmetric
+            inn_psd=[real(S_IN_of_f(1)), S_IN_of_f(2:end-1), real(S_IN_of_f(end)), conj(S_IN_of_f(end-1:-1:2))];
+            inn_rms = sqrt(sum(inn_psd)* delta_f);
+            S_in = sum(reshape(inn_psd, num_ui, M).');
+            S_in=S_in(1:num_ui/2+1);
+            S_in= [real( S_in(1)),  S_in(2:end-1), real( S_in(end)), conj( S_in(end-1:-1:2))];
+            result.S_in=S_in;
+            result.S_in_rms = sqrt(sum(result.S_in)* delta_f);
+        else
+            result.S_in=0;
+            result.S_in_rms =0;
+        end
     else
-        result.S_rn=result.S_rn.*H_rxffe_2;
-        result.S_rn_rms = sqrt(sum(result.S_rn)* delta_f);
+       %% compute S_rn
+        result.S_rn=result.S_rn.*H_rxffe_2;
+        result.S_rn_rms = sqrt(sum(result.S_rn)* delta_f);
+        %% compute S_in
+        result.S_in=result.S_in.*H_rxffe_2;
+        result.S_in_rms = sqrt(sum(result.S_in)* delta_f);
     end

 else % find noise for item that set have tx ffe for each loop
@@ -88,8 +112,13 @@ else % find noise for item that set have tx ffe for each loop
     %% Crosstalk power spectral density
     if ~OP.COMPUTE_COM % result.S_xn and result.S_xn_rms were found in optimizes_fom and passed in with the variable result
         result.S_xn=0;
-        if length(chdata)~=1
-            for xchan=2:length(chdata)
+        if OP.PSDRXCAL
+            num_channel_files = length(chdata) - 1;
+        else
+            num_channel_files = length(chdata);
+        end
+        if num_channel_files~=1
+            for xchan=2:num_channel_files
                 pulse_ctle=filter(ones(1,M),1,chdata(xchan).ctle_imp_response(:).');
                 pulse_ctle=[ pulse_ctle(1:floor(length(pulse_ctle)/M)*M) ];
                 hk(xchan).k=chdata(xchan).pulse_response_w_CFT_TXFFE_noRxFFE.';
@@ -193,7 +222,7 @@ else % find noise for item that set have tx ffe for each loop
         hext_txffe=filter(txffe,1,hext);
         sig_aftert_ctle_pdf = get_pdf_from_sampled_signal(hext_txffe,param.levels,OP.BinSize);
         noise_after_ctle_pdf = sig_aftert_ctle_pdf;
-        sigma_noise = sqrt(result.S_rn_rms^2+result.S_xn_rms^2++result.S_tn_rms^2++result.S_rj_rms^2);
+        sigma_noise = sqrt(result.S_in_rms.^2+result.S_rn_rms^2+result.S_xn_rms^2+result.S_tn_rms^2+result.S_rj_rms^2);
         noise_after_ctle_pdf.y = 1/(sqrt(2*pi)*sigma_noise)*exp(-noise_after_ctle_pdf.x.^2/(2*sigma_noise^2))*OP.BinSize;
         sig_noise_after_ctle_pdf= conv_fct(sig_aftert_ctle_pdf,noise_after_ctle_pdf);
         sig_noise_after_ctle_cdf = cumsum(sig_noise_after_ctle_pdf.y);
@@ -225,7 +254,7 @@ else % find noise for item that set have tx ffe for each loop
         %  result.S_n
     end
     result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn+ result.S_qn;
-    result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn+ result.S_qn+result.S_in;
     result.S_n_rms = sqrt(sum(result.S_n)* delta_f);

     %%
@@ -265,7 +294,7 @@ else % find noise for item that set have tx ffe for each loop
         result.S_isi_rms = sqrt(sum(result.S_isi)* delta_f);
         %%
         result.S_G=result.S_tn+ result.S_rj_jn  + result.S_rn; % eq 178A-30
-        result.S_G=result.S_tn+ result.S_rj_jn  + result.S_rn + result.S_in; % eq 178A-30
         result.S_G_rms = sqrt(sum(result.S_G)* delta_f);
         result.Sn_rho=result.S_isi +result.S_n; % need to include xtalk and isi
         result.Sn_rho_rms = sqrt(sum(result.Sn_rho)* delta_f);
```

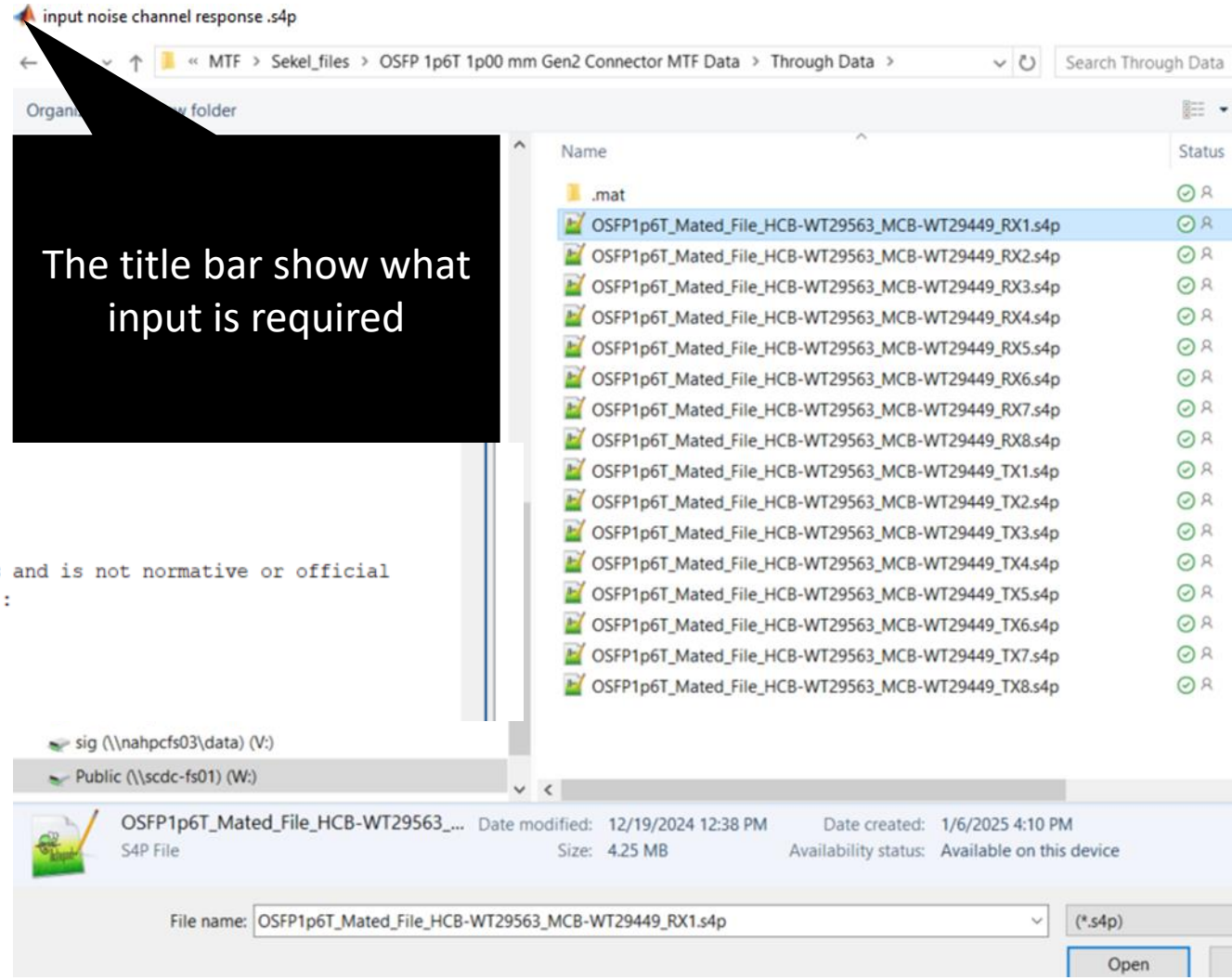
com_ieee8023_

- 39 changed files
  - Review in GIT
- Code complete and initially test
- Configuration sheet need contributions.
- An example provided
- New keywords and interactive dialog and command line syntax

New Dialog

- Question added as to whether want to proceed with a Rx calibration.
- The file request window is the entry for the noise channel

```
>> com_ieee8023_
This is NOT an official IEEE document.
Revision:  This is a computation example for exploring COM and ERL
 for projects like IEEE P802.3bj/b/bs/cd/ck/dj with some exploratory extensions and is not normative or official
Enter config XLS file or return will just pop a window to ask for the XLS file]:
How many FEXT channels are to be entered? [return means no FEXT] 1
How many NEXT channels are to be entered? [return means no NEXT] 1
Do you want to continue with Rx calibration (Y/N) ? [return means yes]
```



COM CLI call adds a noise path

```
com_ieee8023_('C:\Users\richardm\OneDrive - Samtec\COM\802-
COM\Richard_Mellitz_com_code\com_code\config_templates\RiTol_Calitbration\200G\178\config_com_dj_200G_RxTol_CAKR_178_PKGB_05_23_2025.xlsx'...
,1, 1,...
'W:\userdata\channels\IEEE802.3df\sekel_3dj_02_2503\OSFP1p6T_Mated_File_HCB-WT29563_MCB-WT29449_RX1.s4p',...
'W:\userdata\channels\IEEE802.3df\mammenga_3dj_02_2501\FEXT\HCB_vic_TX1_HCB_aggr_RX1_FEXT.s4p', ...
'W:\userdata\channels\IEEE802.3df\mammenga_3dj_02_2501\NEXT\HCB_vic_TX1_HCB_aggr_RX1_NEXT.s4p',...
'W:\userdata\channels\IEEE802.3df\sekel_3dj_02_2503\OSFP1p6T_Mated_File_HCB-WT29563_MCB-WT29449_RX1.s4p');
```

Noise Path

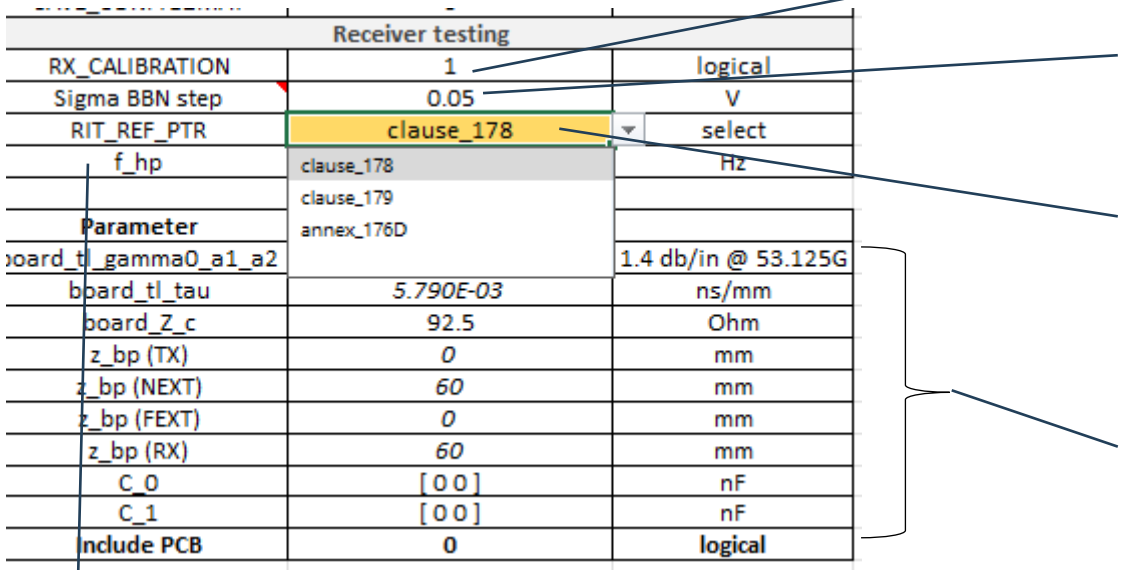
New Keywords

- RIT_REF_PTR – controls noise filter in N_in
  - clause_179 (default)
  - clause_178
  - annex_176d

Output indicates what RMS voltage to set noise source too

```
WC All cases  FAIL ... COM = 2.993 dB
WC All cases DER = 1.208e-04 at COM threshold
WC All cases:  PASS ... ERL = 11.079 dB (12.642 dB, 11.079 dB)
 LOOP with [sigma_bn] = [0.015234] performed with COM = 2.9929
Set Tx calibration noise (sigma_hp) rms voltage to 15.2344 mV.
 redo string is: eval(['My_var_0 = ' getappdata(0,'cmd_str')])
>>
```

Spreadsheet snippets



Receiver testing		
RX_CALIBRATION	1	logical
Sigma BBN step	0.05	V
RIT_REF_PTR	clause_178	select
f_hp		Hz
Parameter		
oard_tl_gamma0_a1_a2		1.4 db/in @ 53.125G
board_tl_tau	*5.790E-03*	ns/mm
board_Z_c	92.5	Ohm
z_bp (TX)	*0*	mm
z_bp (NEXT)	*60*	mm
z_bp (FEXT)	*0*	mm
z_bp (RX)	*60*	mm
C_0	[0 0]	nF
C_1	[0 0]	nF
Include PCB	**0**	**logical**

clause_178
clause_179
annex_176D

Switch for Rx calibration

Starting noise step size

If Rx FFE is used there are 3 Annex choices. 179 is the default. Only used when RxFFE is enabled

This section needs to be filled in from the referring section

f_hp must be set if using Clause 176 or 179. Default value of 0 will result in errors

Thank You!