Apparent Channel Bandwidth
Commit request 4p16_1

Richard Mellitz, Samtec

Adam Gregory, Samtec

IEEE 802.3 Channel Operating Margin (COM) Open Source Ad hoc, August 11, 2026

Contents

- Proposal: Apparent Channel Bandwidth
- Method
- Added Functions in Branch
- I/O
- Backup – Git Details

Proposal: Apparent Channel Bandwidth

- Add computation and report for apparent channel bandwidth
- Ref: 400 GBPL Electrical Objective Kaleidoscope: Reach, Loss, Cutoff, Noise, and COM, 802.3 400GPL Study Group Electronic Teleconference Meeting July 30, 2026

Method: First Persistent CICP Deviation

Define cumulative inverse-channel penalty (CICP):

$$CICP(f) = \frac{\int_0^f |H(\nu)|^{-2}\, d\nu}{\int_0^{F_{max}} |H(\nu)|^{-2}\, d\nu}$$

- *CICP* : a fraction of total inverse-channel penalty accumulated below *f*
- *f* : evaluation frequency
- F_{max}: Upper analysis frequency used for normalization, typically the maximum frequency of the available S-parameter data
- *H (f):* the insertion loss

Inspired by inverse-channel weighting used in Salz-style equalization analysis.

Fit dB of CICP

$$CICP_{\text{fit}} = a_0 + a_1\sqrt{f} + a_2 f + a_3 f^2$$

- Using best adaptive fit

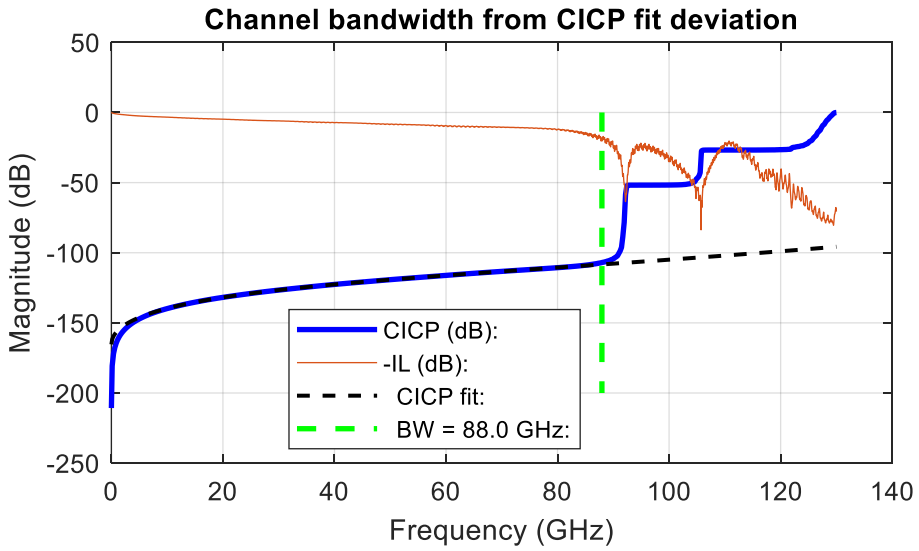
Derivative of smoothed rms fit residual

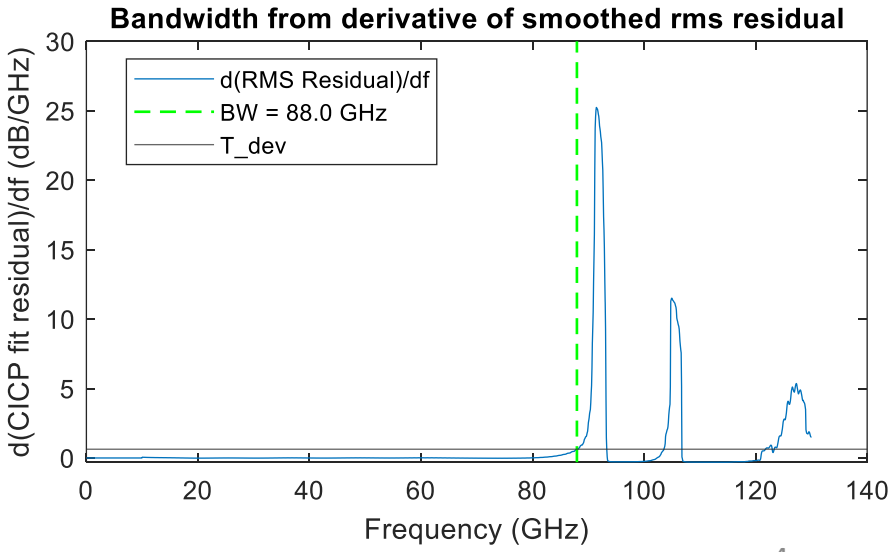
$$D(f) = \frac{d\sigma_{\text{rms}}(f)}{df},\ \sigma_{\text{rms}}(f) = \sqrt{\overline{r^2}},\ r(f) = CICP_{\text{dB}}(f) - CICP_{\text{fit}}(f)$$

Determine frequency of first persistent departure using threshold T_{dev}

$$f_{\text{dev}} = \min\{f\ :\ D(\nu) \geq T_{\text{dev}} \text{ for all } \nu \in [f, f + \Delta f]\}$$

- Δf : persistence window over which the deviation must be sustained.





Added Functions in Branch: aparent_channel_bandwidth

New

- ❑ get_ACBW.m
  - get_CICP_fit_sweep.m
  - get_BW_from_CICP_residual

Modified

- ❑ FD_Processing.m
  - Added get_ACBW of Keyword is set
  - Added ACBW to losses plot
- ❑ read_ParamConfigFile.m
  - Added new keywords
- ❑ get_ILN.m
  - Added fit alpha to function output

I/O

New Configuration variables

- **ACBW (0/1)**
  - **1 enables the computation and output report of apparent channel bandwidth**
  - **Default: 0**
- **T_dev (dB/GHz)**
  - **Threshold for derivative of smoothed rms fit residual (D(f)**
  - **Lower number moves BW lower**
  - **Default: 1**

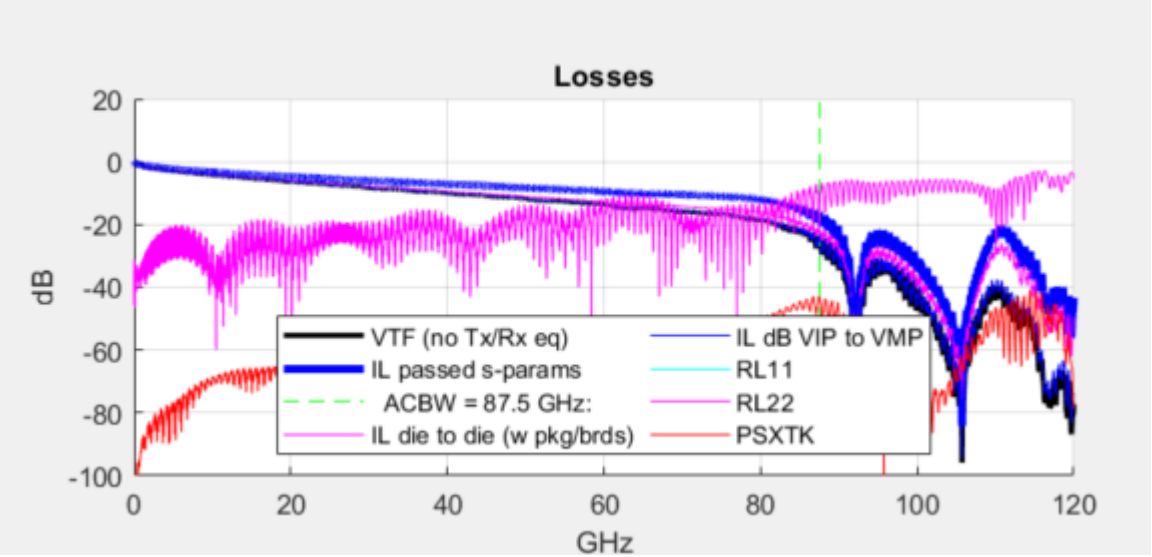
New output if ACBW=1

- **ACBW_GHz**

Configuration sheet example

ACBW	1	logical
T_dev	1	db/GHz

Added xline for ACBW



Thank You!

Backup Details

Changes: FD_Processing.m

src/FD_Processing.m +12 −1 Viewed

```
@@ -189,6 +189,11 @@ for i=1:param.number_of_s4p_files
189 189             output_args.SCMR_FD_CD_ch_dB=SCMR_FD_CD_ch_dB;
190 190             output_args.SCMR_FD_DC_ch_dB=SCMR_FD_DC_ch_dB;
191 191         end
    192 +         %% determine apparent channel bandwidth
    193 +         if OP.ACBW
    194 +             [chdata(i).Bch_GHz,chdata(i).CICP_db,CICP_fit_chdata(i).db,chdata(i).CICP_residual,chdata(i).CICP_alpha] =
        get_ACBW(chdata(i).sdd21f,chdata(i).faxis/1e9,OP,param); % param
    195 +             output_args.ACBW_GHz=chdata(i).Bch_GHz;
    196 +         end
192 197         %%
193 198         [ILD_magft chdata(i).fit_f2_ild] = get_ILN(chdata(i).sdd21f(index_f1:index_f2_ild), chdata(i).faxis(index_f1:index_f2_ild));
194 199         % find fitted loss values by interpolation using full data, no indexing - Adee 2022-08-28

@@ -296,8 +301,14 @@ for i=1:param.number_of_s4p_files
296 301             title('Losses')
297 302             plot(faxis_GHz, 20*log10(abs(squeeze(chdata(i).sdd21f))), 'b', 'LineWidth', 3, 'Disp','IL passed s-params')
298 303             hold on
    304 +             if OP.ACBW
    305 +                 if ~isnan(chdata(i).Bch_GHz)
    306 +                     plot([chdata(i).Bch_GHz chdata(i).Bch_GHz],[min(20*log10(abs(squeeze(chdata(i).sdd21f)))) 20], 'g--', 'LineWidth', .5, ...
    307 +                         'DisplayName', sprintf('  ACBW = %.1f GHz: %s', chdata(i).Bch_GHz))
    308 +                 end
    309 +             end
299 310             plot(faxis_GHz, 20*log10(abs(squeeze(chdata(i).sdd21p_nodie))), 'm-', 'Disp','IL die to die (w pkg/brds)')
300     -           plot(faxis_GHz, 20*log10(abs(squeeze(chdata(i).sdd21p))), 'b-', 'Disp','IL dB VIP to VMP')
    311 +           plot(faxis_GHz, 20*log10(abs(squeeze(chdata(i).sdd21p))), 'b-', 'Disp','IL dB VIP to VMP')
301 312             ylim(get(gca, 'ylim'));
302 313             plot(faxis_GHz, 20*log10(abs(squeeze(chdata(i).sdd11))),'c','Disp','RL11')
303 314             plot(faxis_GHz, 20*log10(abs(squeeze(chdata(i).sdd22))),'m','Disp','RL22')
```

Changes: read_ParamConfigFile.m & get_ILN.m

src/get_ILN.m +1 −1 Viewed

```
-  function [ILN, efit]= get_ILN(sdd21,faxis_f2)
+  function [ILN, efit,alpha]= get_ILN(sdd21,faxis_f2)
   %% License Notice
   %
   % Copyright 2025 802-COM Authors
```

src/read_ParamConfigFile.m +6 −1 Viewed

```
@@ -453,6 +453,9 @@ param.Txnskew=xls_parameter(parameter, 'Txnskew', true, 0 ); % Tx n skew in ps
   param.Rxpskew=xls_parameter(parameter, 'Rxpskew', true, 0 );  % Rx p skew in ps
   param.Rxnskew=xls_parameter(parameter, 'Rxnskew', true, 0 );  % Rx n skew in ps

+  %% apparent channel bandwith parameters
+  param.T_dev= xls_parameter(parameter, 'T_dev', false, 1); % dB/GHz d(CICP residual)/df threshold for apperant channel bandwidth
+
   %%
   OP.TIMESTAMP= xls_parameter(parameter, 'TIMESTAMP', false,0); % Debug feature: a time stamp string is appended to saved file names
   OP.WRITE_CSV_TRANSPOSED= xls_parameter(parameter, 'WRITE_CSV_TRANSPOSED', false,0); % Debug feature: Writes the csv file one field per row
@@ -624,7 +627,7 @@ if OP.RX_CALIBRATION
           OP.RX_CALIBRATION = 0;
       end
   end
-
+  OP.ACBW = xls_parameter(parameter, 'ACBW', false, 0); % report apperant channel bandwidth
   % Trigger some errors on bad parameter combinations
   if OP.PSDRXCAL
       if OP.RX_CALIBRATION
@@ -648,6 +651,8 @@ if ~isempty(preset_txffe)
       for ii=1:length(preset_txffe),param.preset(ii).txffe=preset_txffe(ii,:);end
   end

+
+
   %% At the very end of Parameter reading, swap in the proper Tx and Rx values for package parameters based on pkg name
   if ~isempty(param.PKG_NAME)
       if length(param.PKG_NAME) == 1
```

[Bch_GHz,CICP_db,CICP_fit_db,CICP_residual,alpha,diff_residual] = get_ACBW(Hch,fGHz,OP,param)

```
%% get_ACBW
% Estimates Adaptive Channel Bandwidth (ACBW) from the channel transfer
% function using a Cumulative Inverse-Channel Penalty (CICP) analysis.
%
% The procedure:
% 1. Computes the normalized CICP response from the channel transfer
% function.
% 2. Determines an adaptive CICP reference fit using a frequency-range
% sweep and residual flatness criterion.
% 3. Forms the residual:
% CICP_residual = CICP_db - CICP_fit_db
% 4. Identifies channel bandwidth from the first significant increase
% in residual slope, indicating the onset of increased equalization
% burden.
%
% Interpretation:
% ACBW represents the frequency at which the channel begins to depart
% significantly from its low-frequency CICP trend, indicating the onset
% of rapidly increasing equalization demand.
```

[CICP_residual, CICP_fit_db, alpha, idx_fit,idx_valid,f_upper] =...
get_CICP_fit_sweep(CICP_db, fGHz, OP, varargin)

```
%% get_CICP_fit_sweep
% Performs an adaptive CICP fit-range sweep to identify a stable
% low-frequency CICP trend and corresponding residual.
%
% The routine repeatedly fits the CICP response over candidate frequency
% ranges and evaluates each fit using a residual figure-of-merit (FOM).
% The fit range whose residual exhibits the smallest overall tilt is
% selected as the preferred CICP reference trend.
%
% The selected fit is subsequently used to generate:
% CICP_residual = CICP_db - CICP_fit_db
% which can be used for channel bandwidth determination and other
% residual-based channel analyses.
```

[Bch_GHz, idx_BW , diff_residual] = ...
get_BW_from_CICP_residual(fGHz, residual_db, idx_fit, T_dev, OP, varargin)

```
%% get_BW_from_CICP_residual
% Finds channel bandwidth from the onset of increasing CICP-fit residual.
%
% Intended residual input:
% residual_db = CICP_db - CICP_fit_db;
% Definition:
%
% Bch = first frequency within the valid search region where the
% slope of a smoothed CICP residual exceeds a specified
% threshold (T_dev).
%
% The underlying assumption is that increasing residual slope indicates
% the onset of significant deviation from the low-frequency CICP trend,
% which corresponds to the frequency range where additional equalization
% begins to be required.
```