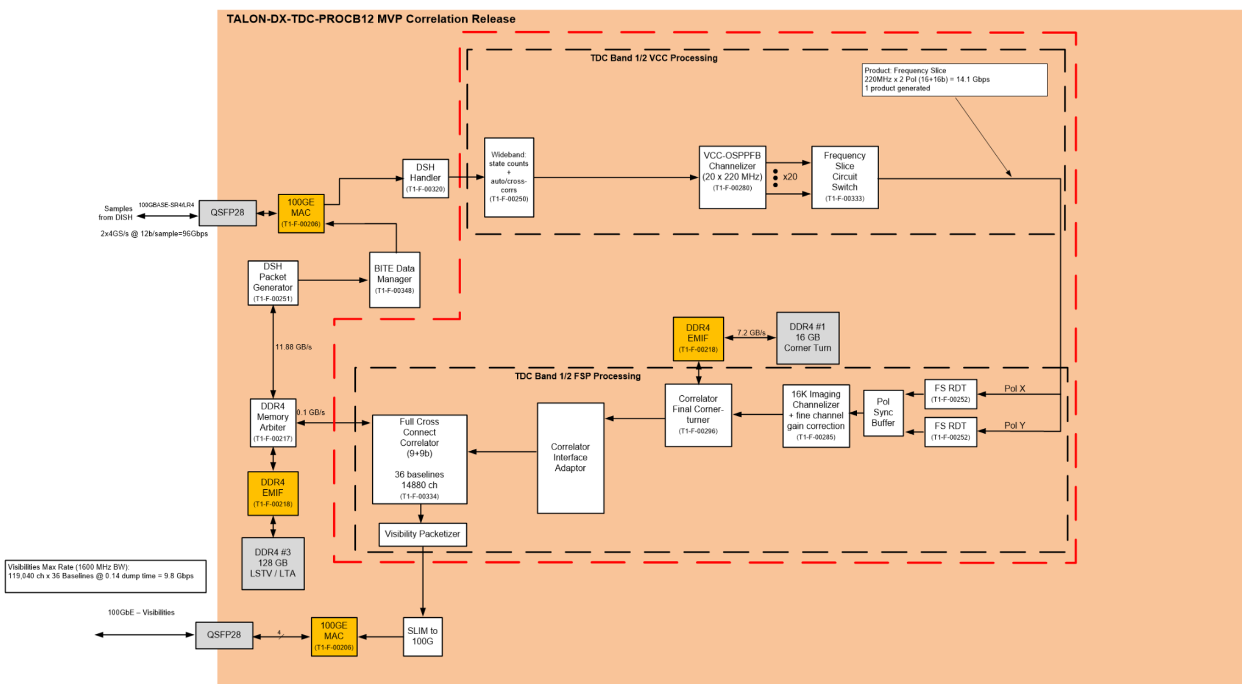
TDC MVP1 Build 1 Test Plan

7 December 2021 – Stephen Harrison

This document describes the tests to be run on TDC MVP1 Build 1. The intent is that these basic single-antenna tests can be adapted to future TDC builds. A block diagram of TDC MVP1 Build 1 FPGA design is shown below. The Built-In Test Environment (BITE) is used to generate stimulus for the data path. A single 200 MHz frequency slice at a time may be selected for processing via the circuit switch. This data is then passed through the resampler, fine channelizer, corner turn, and correlator to produce visibility packets containing the integrated auto- and cross-power from a single antenna.



Test 0.1

This test confirms basic software control of the datapath, and that visibility packets are correctly collected, interpreted and assembled into complete spectra.

BITE Configuration

- Noise input, independent between X and Y pols.
- Unique filter shapes for X and Y pols.
- Sample Rate: Nominal

DUT Configuration

- No resampling (1:1).
- No delay tracking.
- Minimum integration time.

Test Procedure

- Generate BITE test vector into DDR memory.
- For each frequency slice 1 to 9:
    - Select frequency slice via circuit switch.
    - Play back BITE test vector.
    - Collect and interpret visibility packets.
- Assemble the collected visibilities into complete spectra: XX*, YY*, XY*

Expected Results

- XX* and YY* spectra matches the configured FIR filter response in BITE.

Test 0.2

This test confirms the expected behavior of integrating independent noise in the correlator.

BITE Configuration

- Gaussian noise input, same deviation, independent between X and Y pols.
- No bandpass filter.
- Sample Rate: Nominal

DUT Configuration

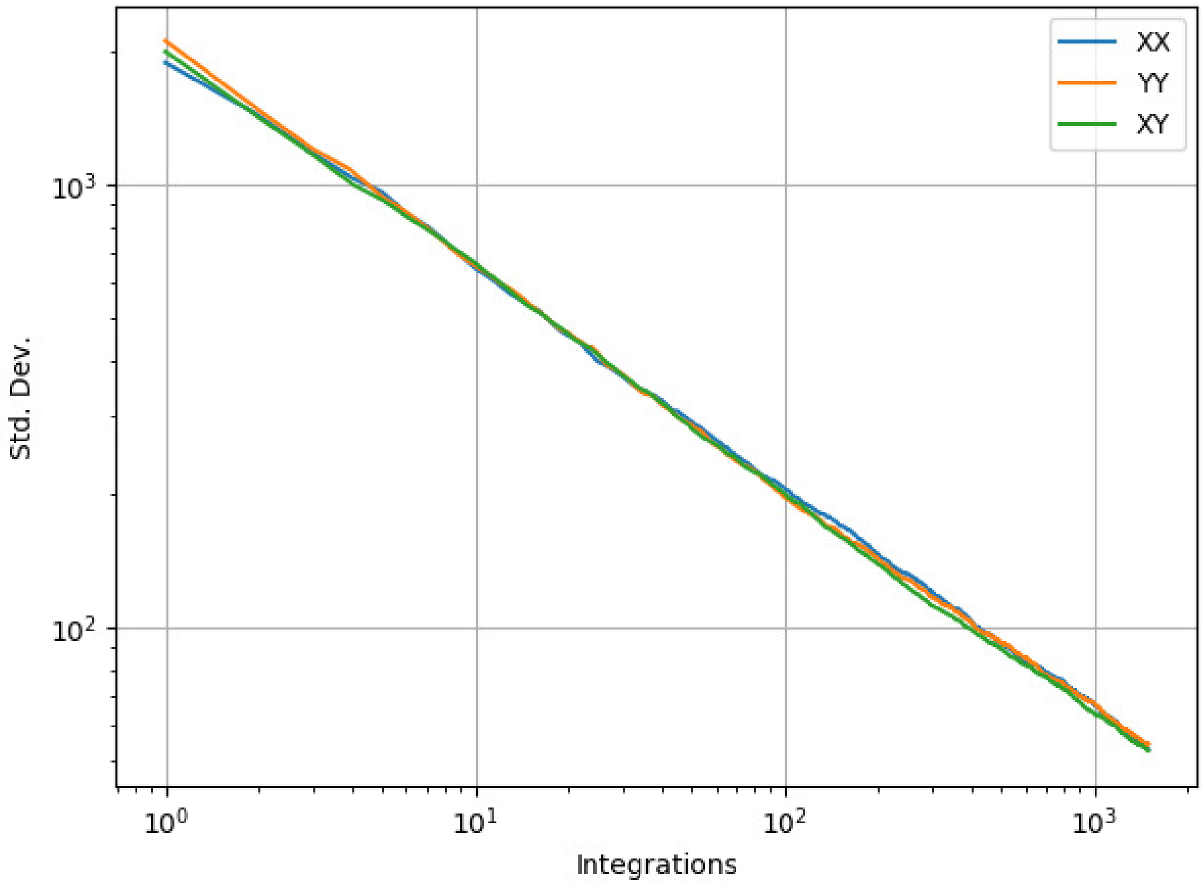
- As per Test 0.1.

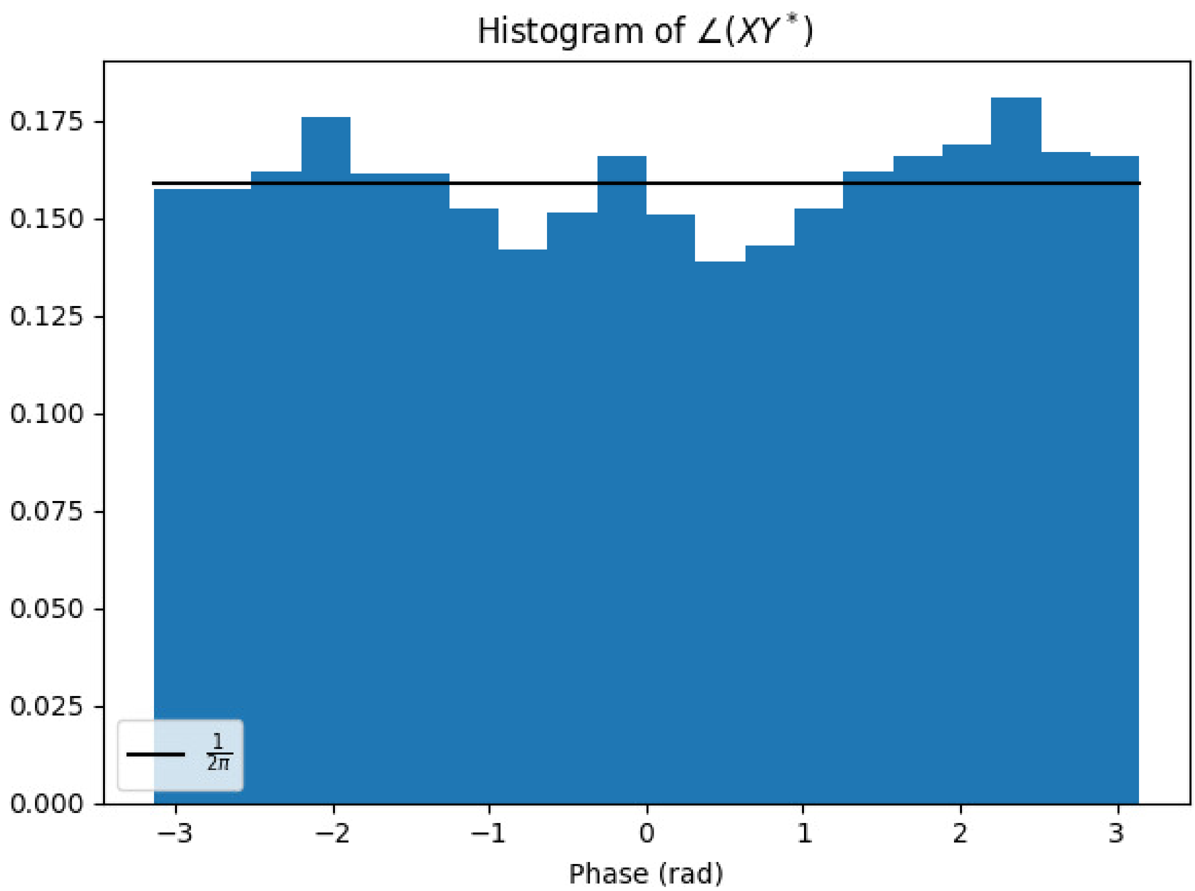
Test Procedure

- Generate BITE test vector into DDR memory.
- For (choose one) frequency slice in 1 to 9:
    - Select frequency slice via circuit switch.
    - Play back BITE test vector.
    - Collect and interpret visibility packets.

Expected Results

- Distribution of ∠XY* approaches U(-π, π).
- Std. Dev. of $\frac{1}{N}\sum_{N}$(XX*, YY*, XY*) tracks 1/$\sqrt{N}$. On a log-log plot Std. Dev. should go down one order of magnitude for every 2 orders of magnitude in integrations.
- |XY*|/sqrt(XX*)sqrt(YY*) approaches 0.





Test 0.3

This test confirms the expected behavior of integrating correlated noise in the correlator with no delay.

BITE Configuration

- Noise input, 50% correlated between X and Y pols.
- No bandpass filter.
- Sample Rate: Nominal

DUT Configuration

- As per Test 0.1.

Test Procedure

- As per Test 0.2

Expected Results

- ∠XY* approaches 0 (i.e. in phase).
- |XY*|/sqrt(XX*)sqrt(YY*) approaches 0.5.
- Repeat with coupling -0.5 – Confirm:
    - ∠XY* approaches 180.
    - |XY*|/sqrt(XX*)sqrt(YY*) approaches 0.5.
- Check other coupling values if there is time.

Test 0.4

This test confirms the expected behavior of integrating correlated noise in the correlator with simple delay.

BITE Configuration

- Noise input, same amplitude, 50% correlated between X and Y pols.
- Enable clock delay in polarization coupler.
- No bandpass filter.
- Sample Rate: Nominal

DUT Configuration

- As per Test 0.1.

Test Procedure

- As per Test 0.2

Expected Results

- ∠XY* phase slope as expected for the chosen frequency slice.
- |XY*|/sqrt(XX*)sqrt(YY*) approaches 0.5. Check other coupling values if there's time.

Test 0.5

This test confirms coarse and fine channel gains.

BITE Configuration

- As per Test 0.2.

DUT Configuration

- As per Test 0.2.

Test Procedure

- Generate BITE test vector into DDR memory.
- For (choose one) frequency slice in 1 to 9:
    - Select frequency slice via circuit switch.
    - Set fine channel gains to vary linearly from -3 to +3 dB
    - For (choose one) pol in X,Y:
        - For frequency slice gain in (0, -3, -6, -9, -12 dB):
            - Set frequency slice gain on selected pol.
            - Play back BITE test vector.
            - Collect and interpret visibility packets.

Expected Results

- Fine channel gain ramp as expected.
- Relative frequency slice gains as expected.

Test 0.6

This test confirms correlator integration times.

BITE Configuration

- As per Test 0.2.

DUT Configuration

- As per Test 0.2, except integration time as per test procedure.

Test Procedure

- Generate BITE test vector into DDR memory.
- For (choose one) frequency slice in 1 to 9:
    - For integration time in (minimum, 2x minimum, 4x minimum, 8x minimum)
        - Select frequency slice via circuit switch.
        - Play back BITE test vector.
        - Collect and interpret visibility packets.

Expected Results

- Offline summation of correlator dumps at different integration times produces the same result as doing longer integrations in the correlator.

Test 1.1

This test confirms the software control and expected behavior of the resampler.

BITE Configuration

- Single tone input, no noise.
- Sample rate as per k value under test.

DUT Configuration

- Resampling as per k value.
- No delay tracking.

Test Procedure

- For k in 1, some intermediate values, 2009, 2010, 2222:
  - Configure BITE for sample rate 3.96e9 + 1800k Hz.
  - For N test tones:
    - Select a target frequency such that it aligns with the center of a fine channel at the nominal rate, near the center of a coarse channel.
    - Select the appropriate frequency slice via the circuit switch.
    - Configure RDT for no resampling (1:1).
    - Play back BITE test vector.
    - Collect and interpret visibility packets.
    - Configure RDT for resampling (BITE rate)/18:220,200,960.
    - Play back BITE test vector.
    - Collect and interpret visibility packets.

Expected Results

- Test tone appears in the desired channel with resampling enabled and does not without resampling.
- Confirm resampling artifacts are limited to acceptable level (compare with non-resampled version).

Test 2.1

This test confirms the expected behavior of delay tracking.

BITE Configuration

- Noise input, identical for both X and Y.
- Sample rate: Nominal

DUT Configuration

- 1:1 Resampling.
- Delay configured as per test procedure.

Test Procedure

- For delay setpoint in (Table below):
  - For polarization in X, Y:
    - Configure RDT delay step for polarization under test, aligned with integration boundaries.
    - Configure RDT delay setpoint for other polarization.
    - Collect and interpret visibility packets.

Delay Setpoint	**Delay Steps**
-1 ms	-1ms + [0..100]*100ps
0 ms	0ms + [-50..50]*100ps
1 ms	1ms + [-100..0]*100ps

Expected Results

- Phase ramp in ∠XY* steps as expected for each delay step, for each polarization (tested independently).

Test 3.1

This test confirms the fine channel response over a limited range. The nature of BITE is that it would take a long time to comprehensively sweep the entire frequency range.

BITE Configuration

- Single tone input, no noise.
- Sample rate: Nominal

DUT Configuration

- 1:1 Resampling.
- No Delay Tracking.

Test Procedure

- Choose a target fine channel, near the center of a coarse channel.
- Select the appropriate coarse channel via the circuit switch.
- Choose frequency steps such that 101 steps covers 4 channels, centered on the target frequency.
- For frequency step in 1..N:
    - Configure the tone frequency
    - Generate the BITE test vector
    - Playback the BITE test vector
    - Collect and interpret visibility packets.

Expected Results

- Plotting (XX, YY) vs. input frequency traces out the fine channel response.
- Attenuation of the target channel meets the design spec at the center of adjacent channels.
- Passband ripple in the channel meets the design spec.
- Power sum of the 4 channels is flat i.e. no scalloping loss at midpoint between channels.

Test 3.2

This test confirms the fine channel response over a limited range, as it crosses the coarse channel boundary. The nature of BITE is that it would take a long time to comprehensively sweep the entire frequency range.

BITE Configuration

- Single tone input, no noise.
- Sample rate as per k value.

DUT Configuration

- Resampling as per k value.
- No Delay Tracking.
- Minimum Integration time.

Test Procedure

- For k in 1, 2222:
  - Choose a target fine channel, such that the upper edge of the channel aligns with the coarse channel boundary.
  - Choose frequency steps such that 101 steps covers 4 channels, centered on the target frequency
  - For frequency step in 1..N:
    - Configure the tone frequency
    - Generate the BITE test vector
    - For frequency step in M, M+1:
      - Select the appropriate frequency slice via the circuit switch.
      - Playback the BITE test vector
      - Collect and interpret visibility packets.

Expected Results

- Similar to Test 3.1 processing, except 2 fine channels are from the upper edge of slice M, and 2 fine channels are from the lower edge of slice M+1.
  - Plotting XX, YY vs. input frequency traces out the fine channel response.
  - Attenuation of the target channel meets the design spec at the center of adjacent channels.
  - Passband ripple in the channel meets the design spec.
  - Power sum of the 4 channels is flat i.e. no scalloping loss at midpoint between channels.
- Resampling artifacts are at an acceptable level.

Test 4.1

This test confirms the expected behavior of delay in BITE. The expected results should be similar to Test 2.

BITE Configuration

- Noise input, identical for both X and Y.
- Sample rate: Nominal

- Delay as per test procedure.

DUT Configuration

- 1:1 Resampling.
- No Delay

Test Procedure

- For delay setpoint in (Table as per Test 2.1):
  - For polarization in X, Y:
    - Configure BITE RDT delay step for polarization under test, aligned with integration boundaries.
    - Configure RDT delay setpoint for other polarization.
    - Collect and interpret visibility packets.

Expected Results

- Phase ramp in ∠XY* steps as expected for each delay step.

Test 5.1

This test confirms that the delay tracking works end-to-end.

BITE Configuration

- Noise input, identical for both X and Y.
- Sample rate: Nominal
- Delay as per test procedure.

DUT Configuration

- 1:1 Resampling.
- Delay as per test procedure.

Test Procedure

- For polarization in X, Y:
  - For BITE: Generate a non-trivial sky model that has enough of a delay to detect a difference between polarizations, and varies slow enough that it will not average out over 1 integration period.
  - Generate the BITE test vector
  - With RDT configured for no delay tracking:

        - Playback the BITE test vector
        - Collect and interpret visibility packets.
    - With RDT configured with matching delay correction model:
        - Playback the BITE test vector
        - Collect and interpret visibility packets.

Expected Results

- With no delay tracking:
    - ∠XY* tracks the model input to BITE.
- With delay tracking:
    - ∠XY* is 0.
    - |XY*|/sqrt(XX*)sqrt(YY*) approaches 1.

Test 5.2

This test confirms that the resampling and delay tracking together works end-to-end.

BITE Configuration

- Noise input, identical for both X and Y (try 50% correlated if there's time).
- Sample rate: k value as per test procedure.
- Delay as per test procedure.

DUT Configuration

- Resampling with k value as per test procedure.
- Delay as per test procedure.

Test Procedure

- For k in 1, some intermediate value, 2009, 2010, 2222:
    - For polarization in X, Y:
        - For BITE: Generate a non-trivial sky model that has enough of a delay to detect a difference between polarizations, and varies slow enough that it will not average out over 1 integration period.
        - For RDT: Generate the matching delay correction model.
        - Generate the BITE test vector
        - Playback the BITE test vector
        - Collect and interpret visibility packets.

Expected Results

- ∠XY* is zero.
- |XY*|/sqrt(XX*)sqrt(YY*) approaches 1 (0.5 if there's time).

Test 6.1

This test confirms basic flagging operation.

BITE Configuration

- Noise input, independent between X and Y pols.
- No bandpass filter.
- Sample Rate: Nominal
- Saturation flagging with period = 5 integrations, duty cycle = 1 integration (minus channelizer behavior).

DUT Configuration

- As per Test 0.1.

Test Procedure

- Generate BITE test vector into DDR memory.
- For (choose one) frequency slice in 1 to 9:
    - Select frequency slice via circuit switch.
    - Play back BITE test vector.
    - Collect and interpret visibility packets.

Expected Results

- Output flagging results as expected.

Test 6.2

This test confirms the time centroid index behaviour.

BITE Configuration

- Noise input, independent between X and Y pols.
- No bandpass filter.
- Sample Rate: Nominal

- Saturation flagging with period = 1.25 integrations, duty cycle = 0.25 integration (minus channelizer behavior).

DUT Configuration

- As per Test 0.1.

Test Procedure

- Generate BITE test vector into DDR memory.
- For (choose one) frequency slice in 1 to 9:
    - Select frequency slice via circuit switch.
    - Play back BITE test vector.
    - Collect and interpret visibility packets.

Expected Results

- Output flagging fraction as expected.
- Output time centroid index as expected.

Test 6.3

This test confirms flagging faster than 1 integration, but slow enough to prevent the fine channelizer from always being in the flagging state.

BITE Configuration

- Noise input, independent between X and Y pols.
- No bandpass filter.
- Sample Rate: Nominal
- Saturation flagging with period = 10 fine channelizer frames, duty cycle as per test procedure

DUT Configuration

- As per Test 0.1.

Test Procedure

- Generate BITE test vector into DDR memory.
- For (choose one) frequency slice in 1 to 9:
    - Select frequency slice via circuit switch.
    - For duty cycle in 10%, 20%, 30%, 40%:

        - Play back BITE test vector.
        - Collect and interpret visibility packets.

Expected Results

- Output flagging fraction as expected: 60%, 70%, 80%, 90% as it will take 5 frames for the coarse channelizer to clear.

Test 6.4

This test confirms per-channel flagging.

BITE Configuration

- Tone input.
- No bandpass filter.
- Sample Rate: Nominal

DUT Configuration

- As per Test 0.1.
- Fine channel gains as per test procedure.

Test Procedure

- Choose a target frequency.
- Configure the BITE tone generator.
- For one pol, configure the gain of this target frequency such that this channel will be flagged at the output of the 16k channelizer.
- Generate BITE test vector into DDR memory.
- Select the appropriate frequency slice via circuit switch.
- Play back BITE test vector.
- Collect and interpret visibility packets.

Expected Results

- Confirm channel is flagged in output visibilities.

Test 7.1

This test confirms longer runs (>> BITE vector length) can be executed.

BITE Configuration

- Noise input, identical for both X and Y (other values of correlation if there's time).
- Sample rate: k value as per test procedure.
- Delay as per test procedure.

DUT Configuration

- Resampling with k value as per test procedure.
- Delay as per test procedure.

Test Procedure

- For some k, for both pols:
    - Ensure that the delay models are continuous over the BITE playback boundary (e.g. Sinusoidal)
    - For BITE: Configure sky delay model.
    - For RDT: Configure delay correction model.
    - Generate the BITE test vector
    - For N playbacks of the BITE test vector:
        - Collect and interpret visibility packets.

Expected Results

- $\angle XY^*$ is zero.
- |XY*|/sqrt(XX*)sqrt(YY*) approaches 1.