Full-Duplex Experiment Using ZedBoard with AD9361 SDR

Kelvin Kuang-Chi Lee
Tamkang University

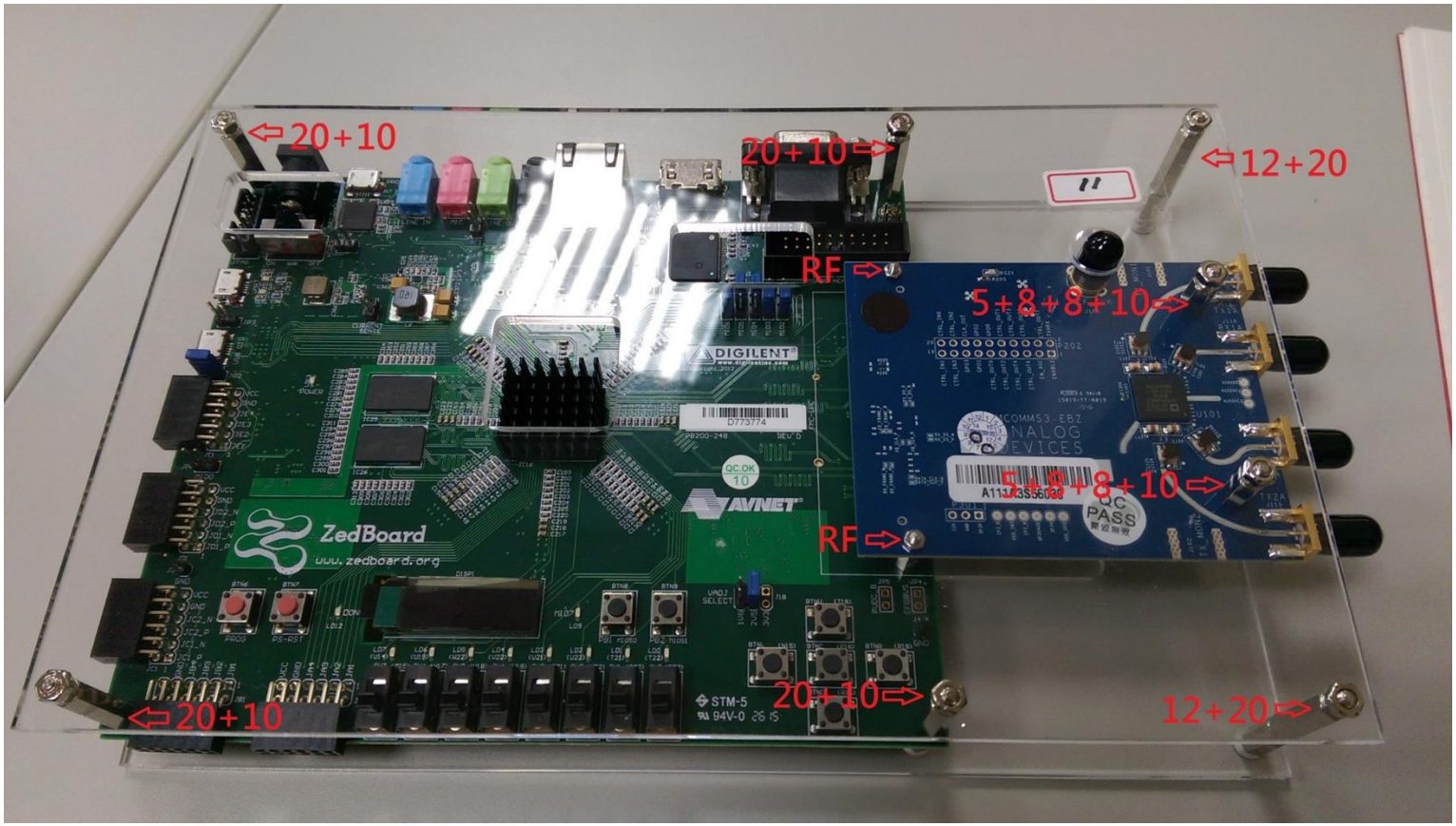
Overview of the Experiment

- Objective
  - Instrument a full-duplex (FDX) communication system with one half-duplex (HDX) transmitter (Tx), one FDX relay transceiver (Relay), and one FDX receiver (Rx)
- Key Elements:
  - Zedboard Zynq-7000 with AD9361
  - Omni-directional and directional antennas
  - Zynq QPSK transmitter/receivrer blocks
  - Zynq QPSK Relay self-interference cancellation (SIC) algorithms
    - SGD-based vs. RLS based SIC
  - Zadoff-Chu (ZC) sequence generation and synchronization

System Setup (1)

- SDR
  - HDX TX (ZedBoard + AD9361 + omni-directional antenna)
  - FDX Relay (ZedBoard + AD9361 + directional antennas)
  - HDX RX (ZedBoard + AD9361 + omni-directional antenna)
- SIC @ Relay SDR
- ZC reference sequence for synchronization

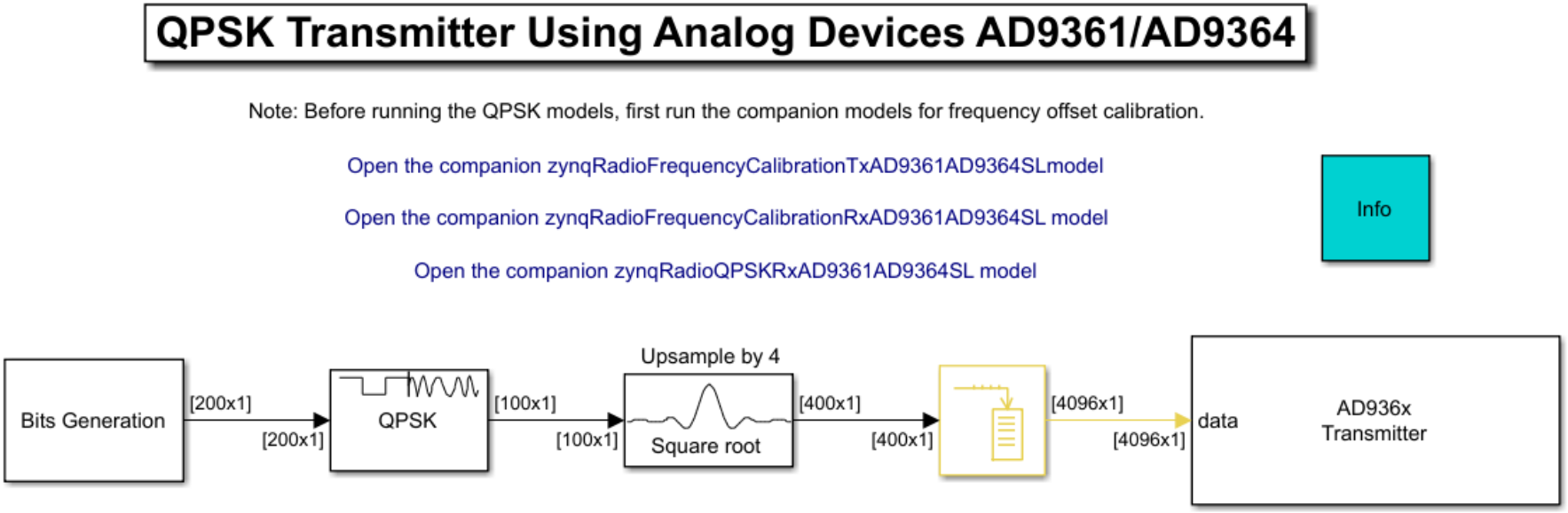
System Setup (2)



Lab-3: IB FDX Communications

- Tx SDR Setup
  - Setup ZedBoard + AD9361 as the HDX QPSK Tx SDR
  - Configure "AD9361 Transmitter" (i.e., IP address, carrier frequency = 5G Hz, antenna selection, baseband sample rate, gain, …)
  - Convert the "Hello World ###" message to baseband QPSK modulated signal including Baker/ZC sequence; transmit data to "Relay SDR" via the omni-directional antenna

QPSK Transmitter SDR (1)



QPSK Transmitter SDR (2)

Block Parameters: AD936x Transmitter

AD936x Transmitter

Transmit data using an AD936x.

Main | Filter | Advanced

Radio Connection

Radio IP address: 192.168.3.2

Info

Radio Properties

Source of center frequency: Dialog

Center frequency (Hz): 5e9+15500 <- Post Calibration result 5e+09

Source of gain: Dialog

Gain (dB): -10

Channel mapping: 1

Baseband sample rate (Hz): sdrqpsktx.RadioFrontEndSampleRate 520841

Data

Enable output port for underflow indicator

OK | Cancel | Help | Apply

Full-Duplex (FDX) Relay SDR Setup

- Configure AD9361 Tx SDR block
  - Center Frequency: 5 GHz
  - Baseband Sample Rate (i.e., data rate) = 520841 Hz
  - Set the Input Data Type: Complex Input
  - Enable Transmit Continuously (optional)

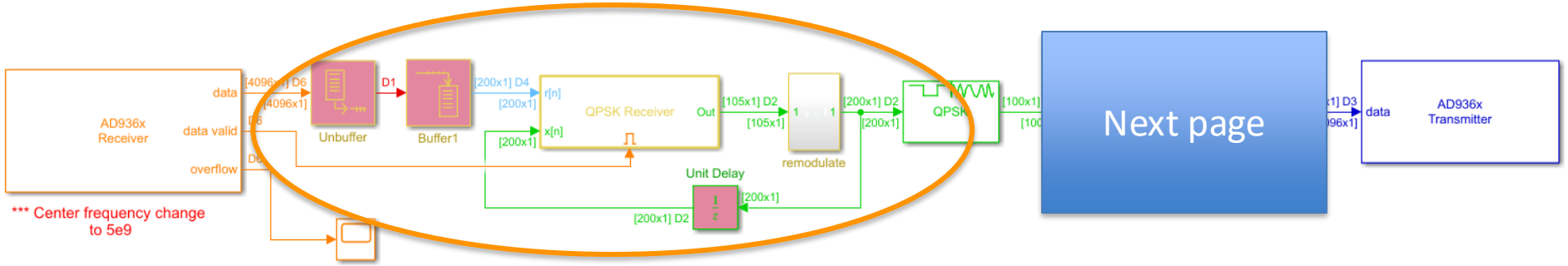
- Configure AD9361 Rx SDR block
  - Center Frequency: 5 GHz
  - Baseband Sample Rate (i.e., data rate) = 520841 MHz
  - Set the Output Data Type: Complex Output

QPSK Relay SDR (0)

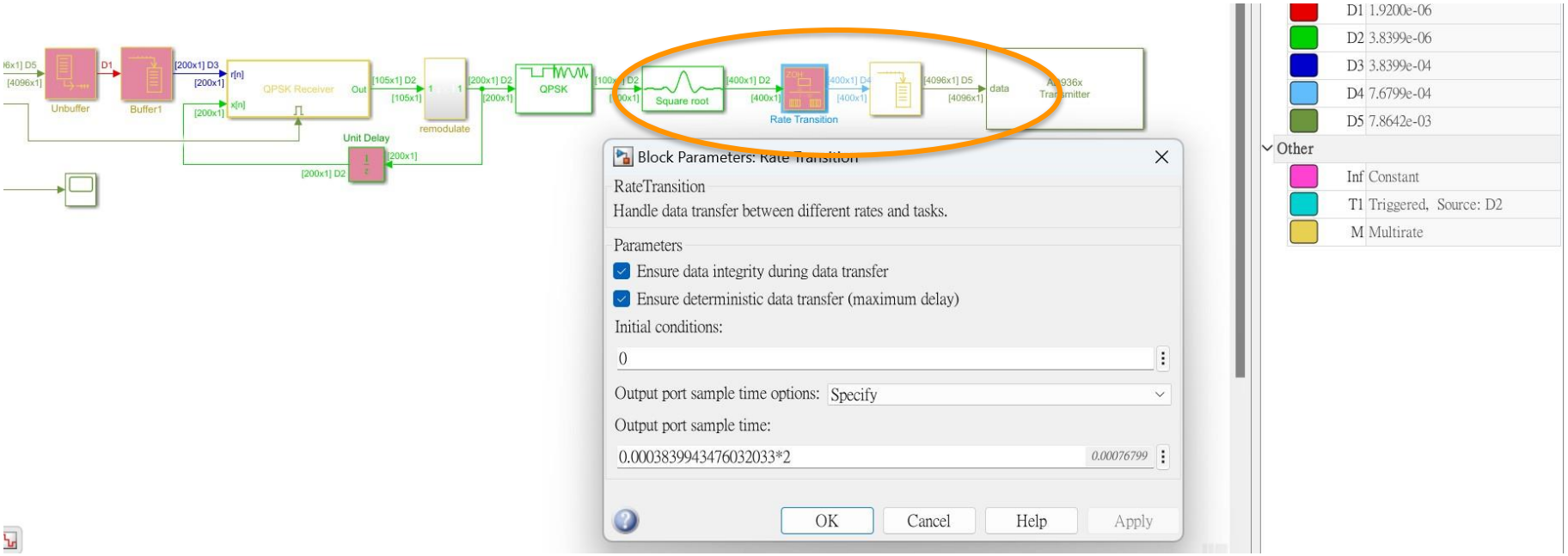
- FDX QPSK Relay SDR
  - QPSK Receiver: demodulates the received signal
  - Unbuffer: frame to sample conversion
  - Buffer: sample to frame conversion
  - remodulate: processes and forwards the received bits
  - QPSK modulator: convert bits to constellation

➢ Enable Sample Time Color Coding

➢ Debug > Information Overlays > Text and Colors

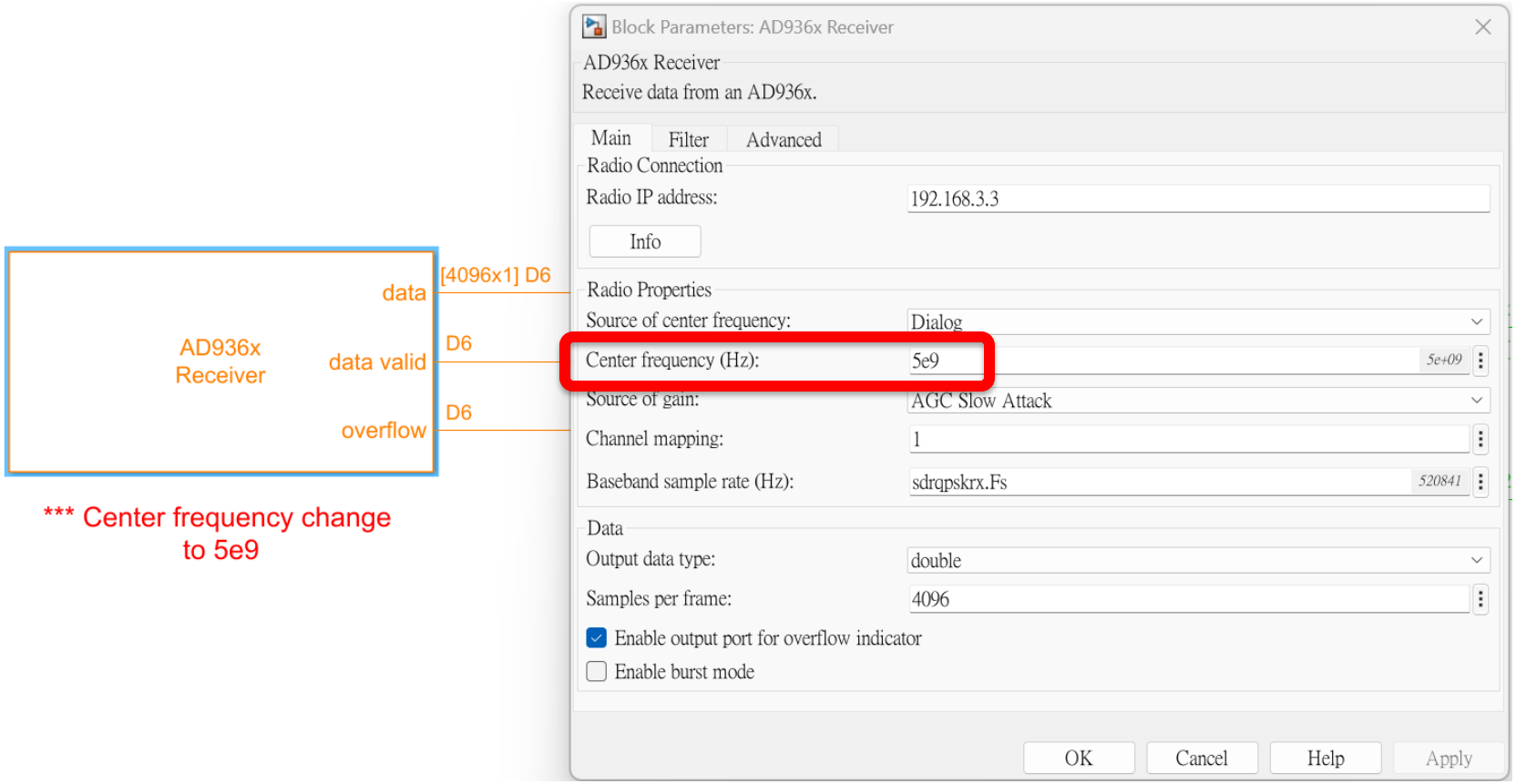


QPSK Relay SDR (1)



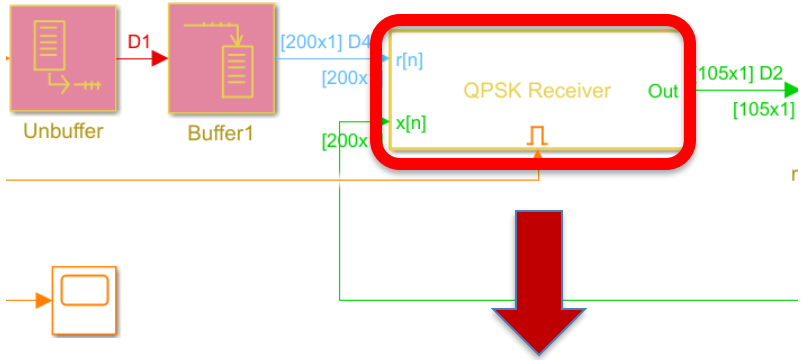
QPSK Relay SDR (2)

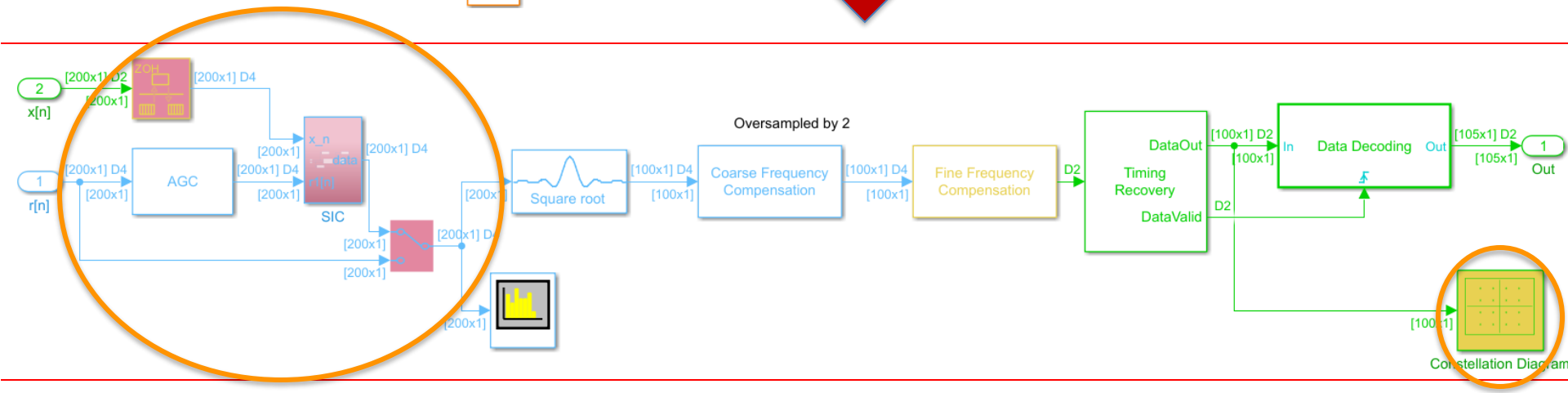
- AD9361 Receiver
  - Bandpass (RF) to baseband conversion



QPSK Relay SDR (4)

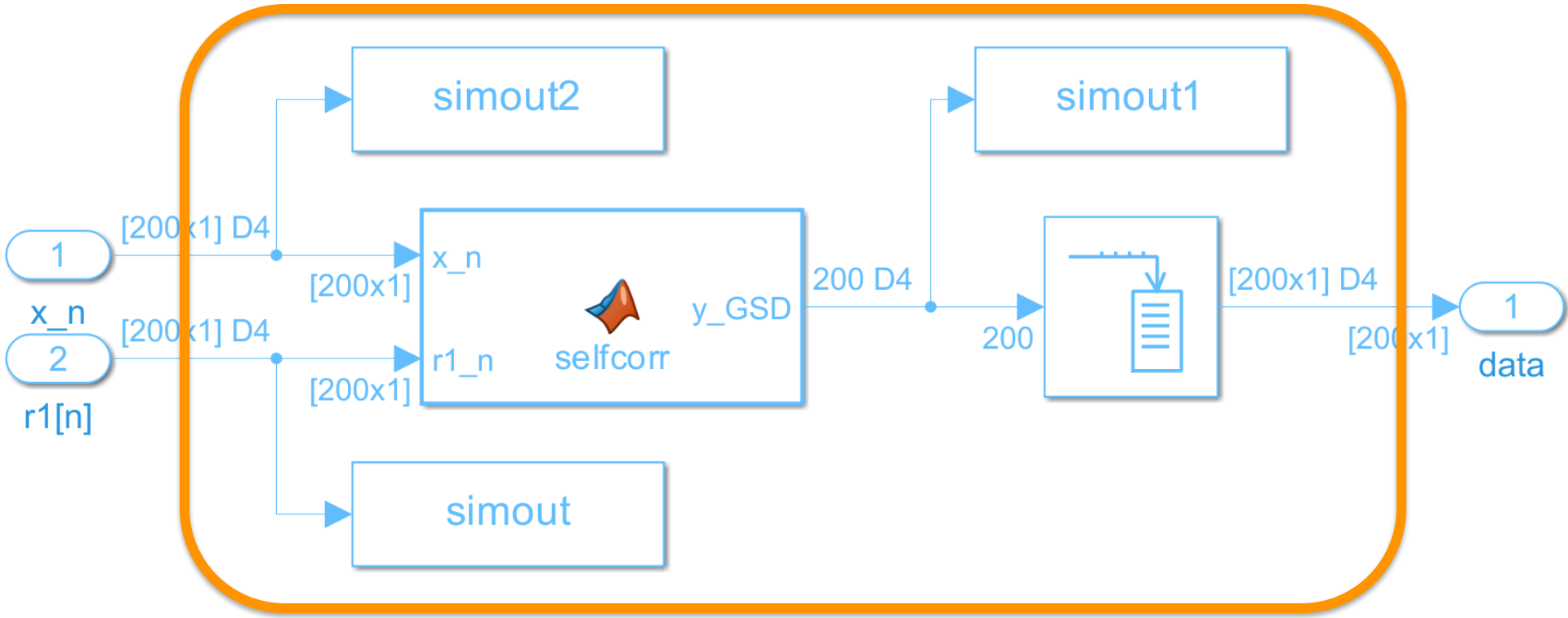
- QPSK Receiver
  - Perform QPSK demodulation





QPSK Relay SDR (6)

- QPSK Receiver
  - SIC (Self-Interference Cancellation)



Side note: Digital SIC Algorithms

- Stochastic Gradient Descent (SGD)
- Recursive Least Squares (RLS)

Table 1 SGD Algorithm for Iteratively Updating $\mathbf{a}_n$.

1: Initialization:

2: $\mathbf{a}_n = \mathbf{0}$;

3: Iteration loop:

4: $\mathbf{a}_n = \mathbf{a}_{n-1} + \mu(r[n] - \mathbf{a}_{n-1}^{*}\bar{\mathbf{x}}_n)\bar{\mathbf{x}}_n$.

Table 2 RLS Algorithm for Iteratively Updating $\mathbf{a}_n$.

1: Initialization:

2: $\mathbf{a}_n = \mathbf{0}$, $\mathbf{P}_0 = \varepsilon^{-1}\mathbf{I}_{L_a}$, $0 \le \varepsilon \le 1$;

3: Iteration loop:

4: $\xi_n = (1 + \lambda^{-1}\bar{\mathbf{x}}_n^{*}\mathbf{P}_{n-1}\bar{\mathbf{x}}_n)^{-1}\lambda^{-1}\mathbf{P}_{n-1}\bar{\mathbf{x}}_n$;

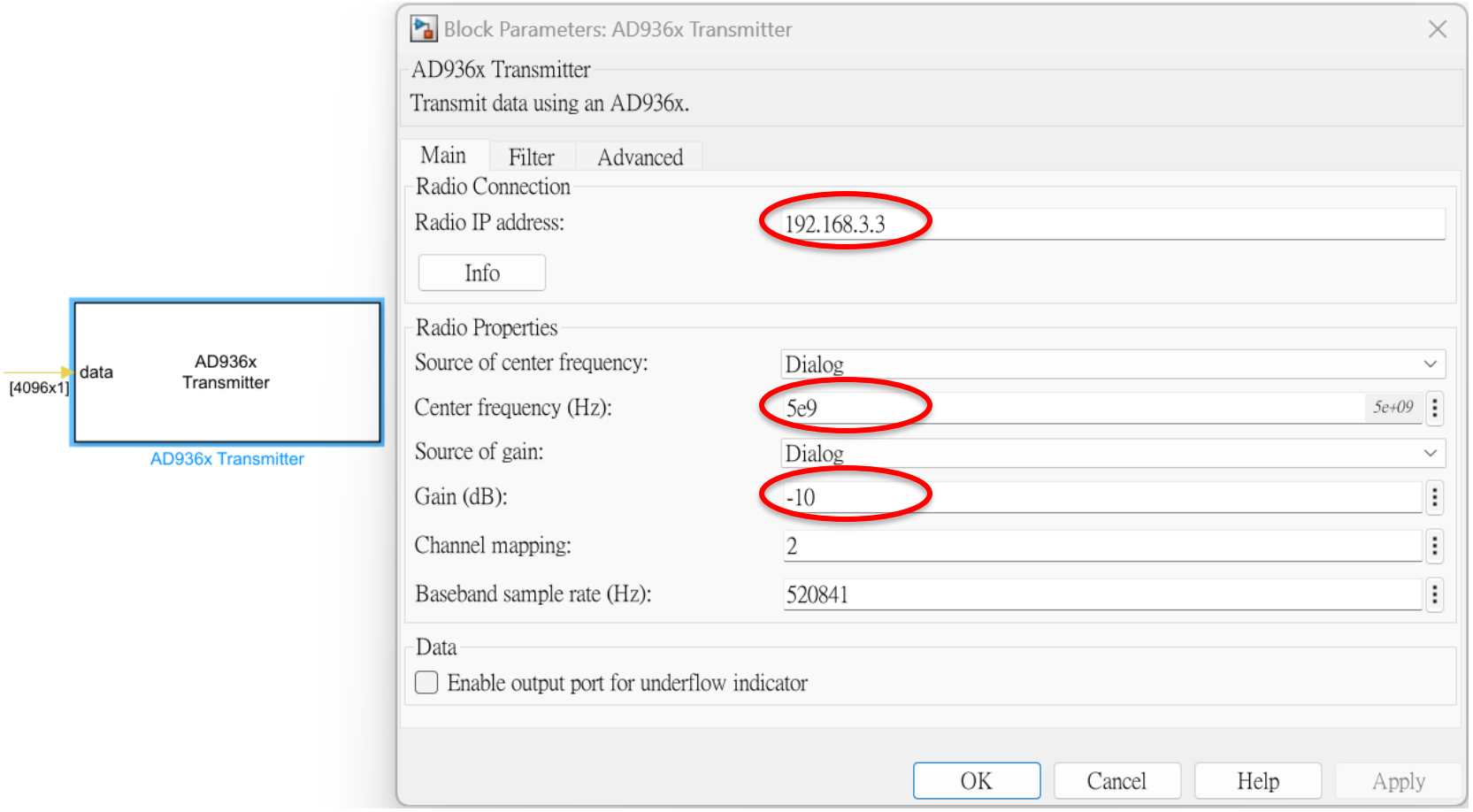
5: $\mathbf{a}_n = \mathbf{a}_{n-1} + \xi_n(r[n] - \mathbf{a}_{n-1}^{*}\bar{\mathbf{x}}_n)^{*}$;

6: $\mathbf{P}_n = \frac{1}{\lambda}(\mathbf{P}_{n-1} - \xi_n\bar{\mathbf{x}}_n^{*}\mathbf{P}_{n-1})$.

[1] M. L. Ku and S. A. Lo, “Full-duplex decode-and-forward relay systems utilizing software-defined radio: From prototyping, implementation to experiments”

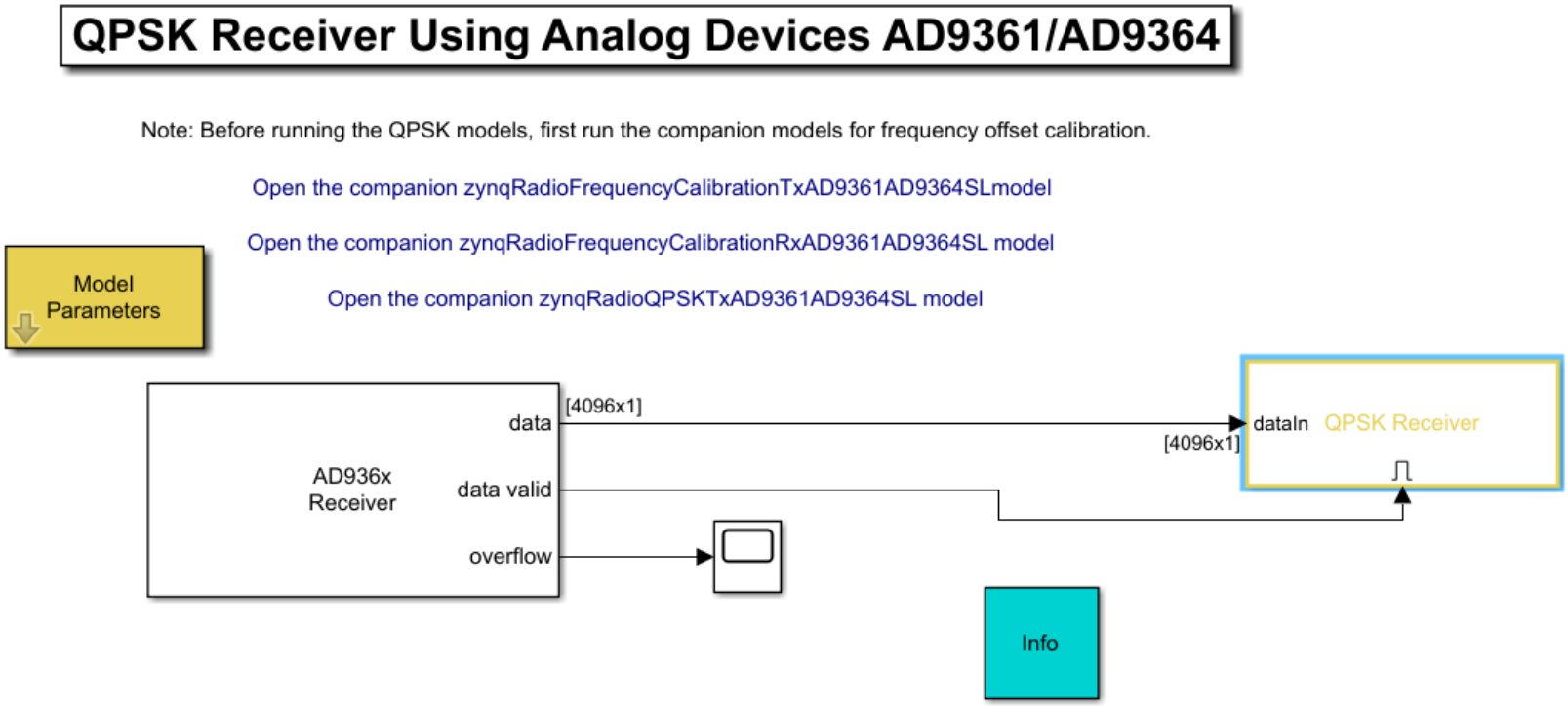
QPSK Relay SDR (9)

- AD9361 Transmitter



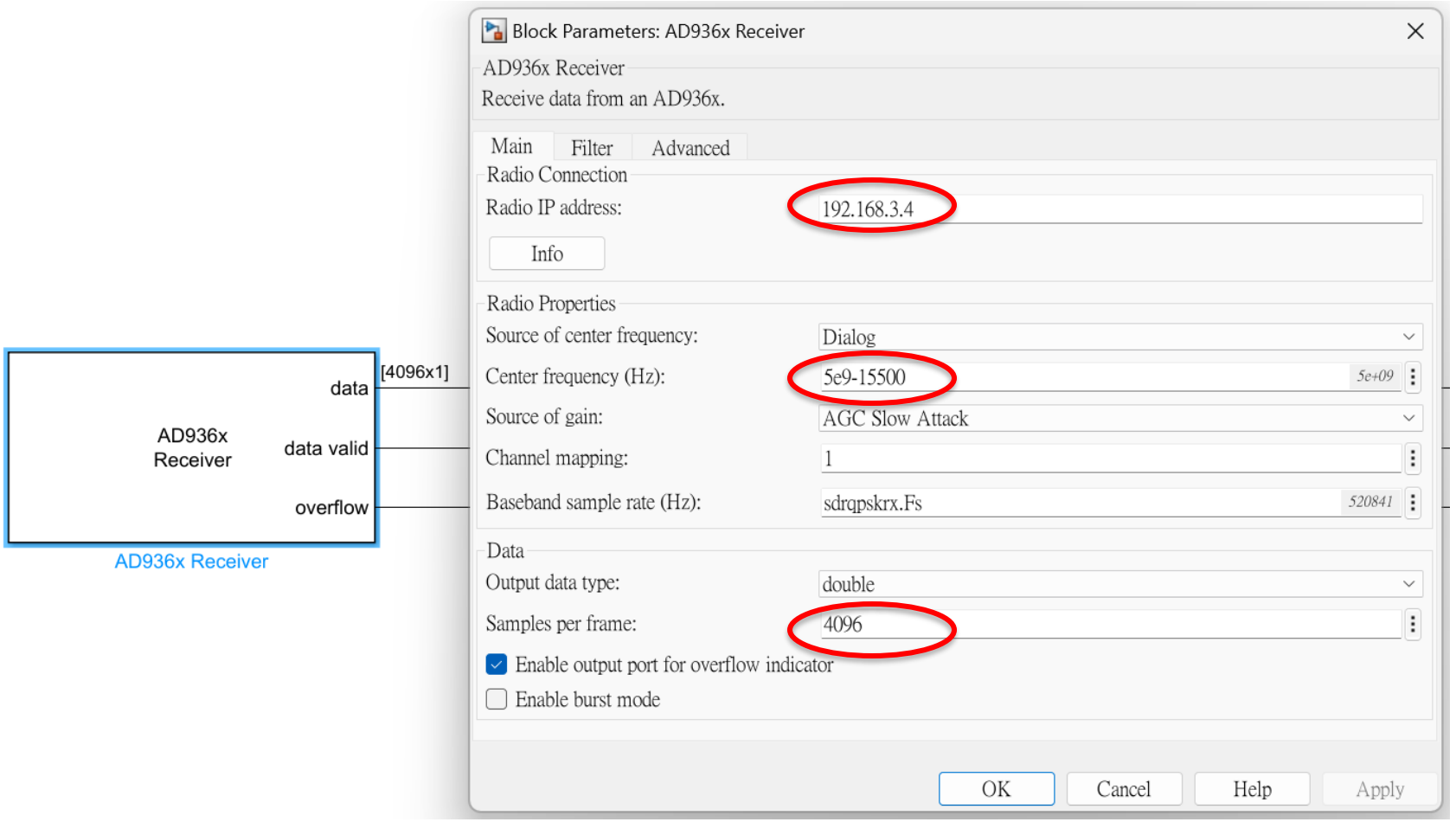
QPSK Receiver SDR (1)

- QPSK Receiver simulink model (HDX)



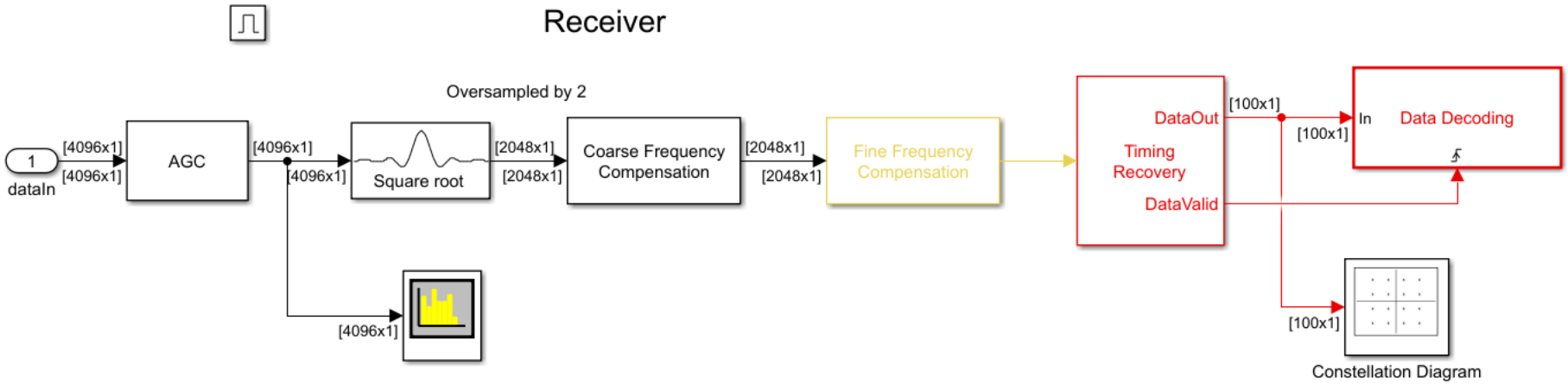
QPSK Receiver SDR (2)

- QPSK Receiver simulink model (HDX)



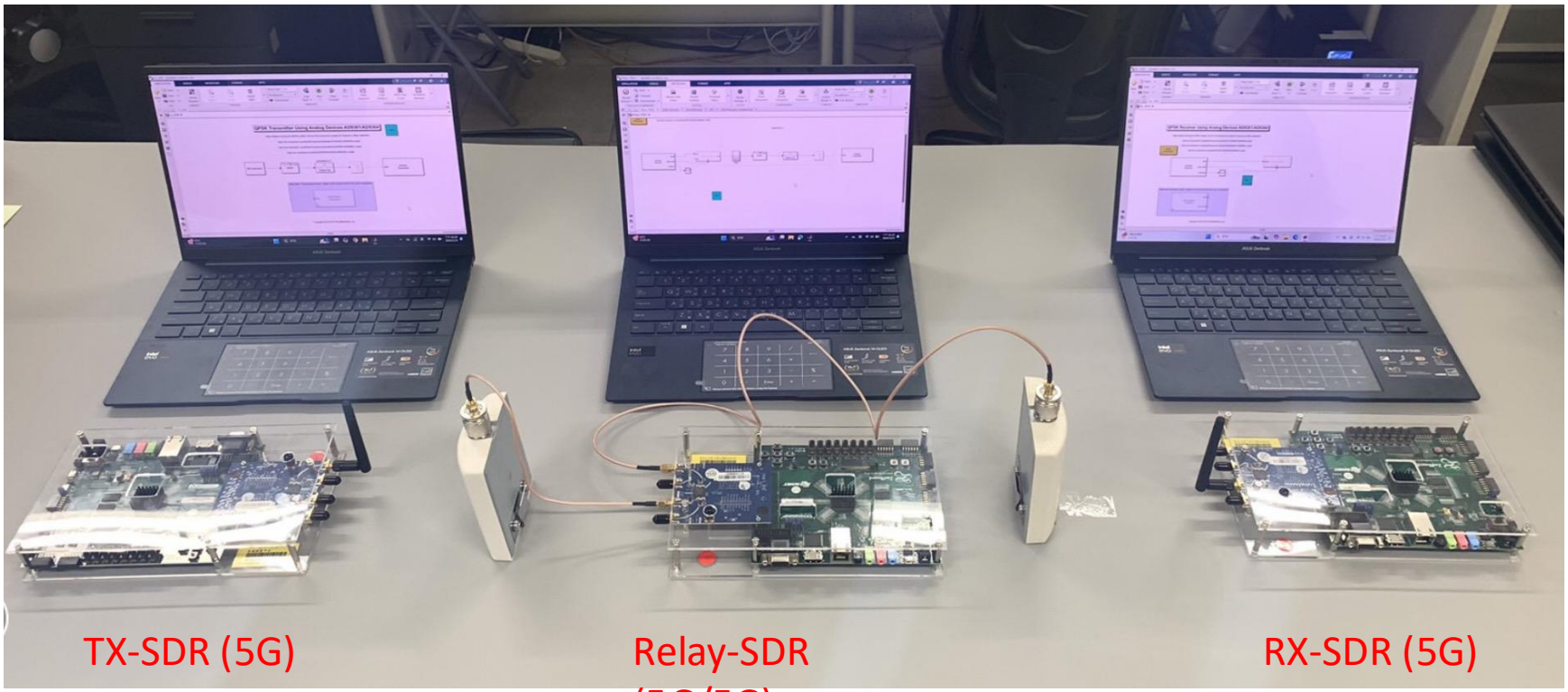
QPSK Receiver SDR (3)

- QPSK Receiver simulink model (HDX)



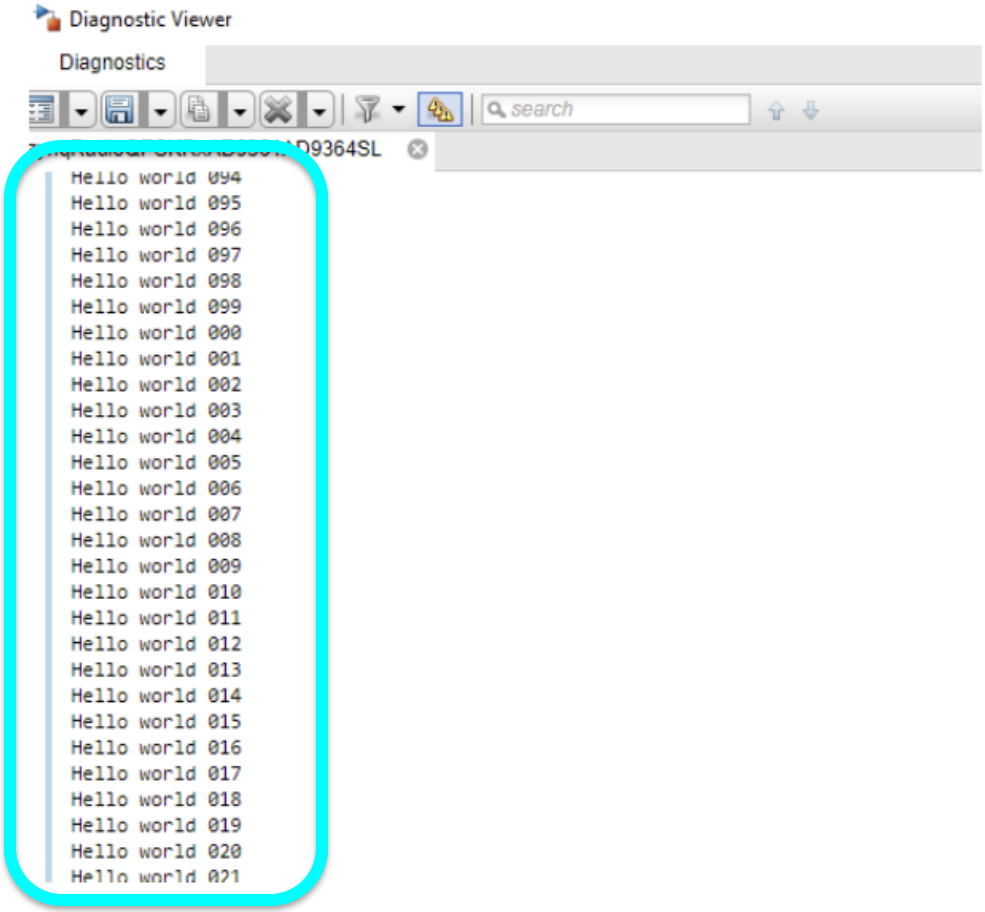
IB FDX QPSK OTA Test (1)

- Execute Tx SDR , Relay SDR and Rx SDR simultaneously.
  - using ‘sim’ mode to display the message in the command window
  - in simulink: Debug > Diagnostics >Diagnostic Viewer



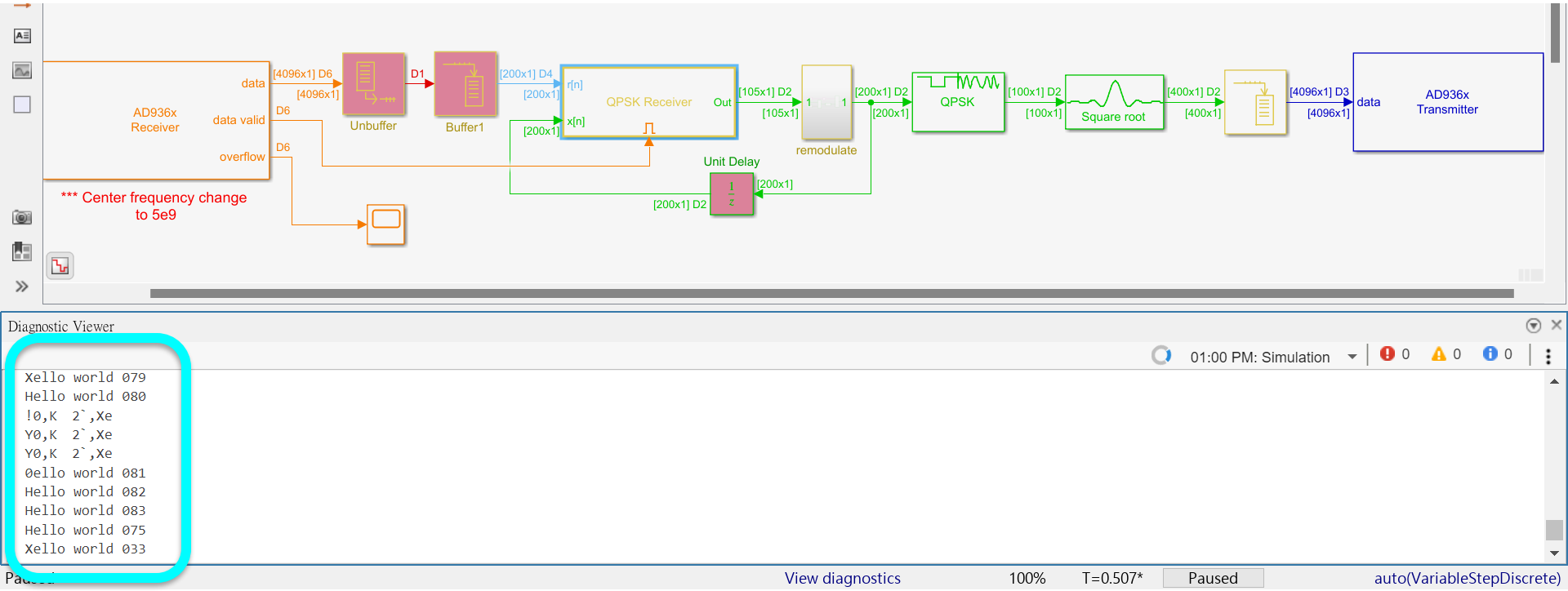
IB FDX QPSK OTA Test (2)

- OTA results appeared at Tx SDR

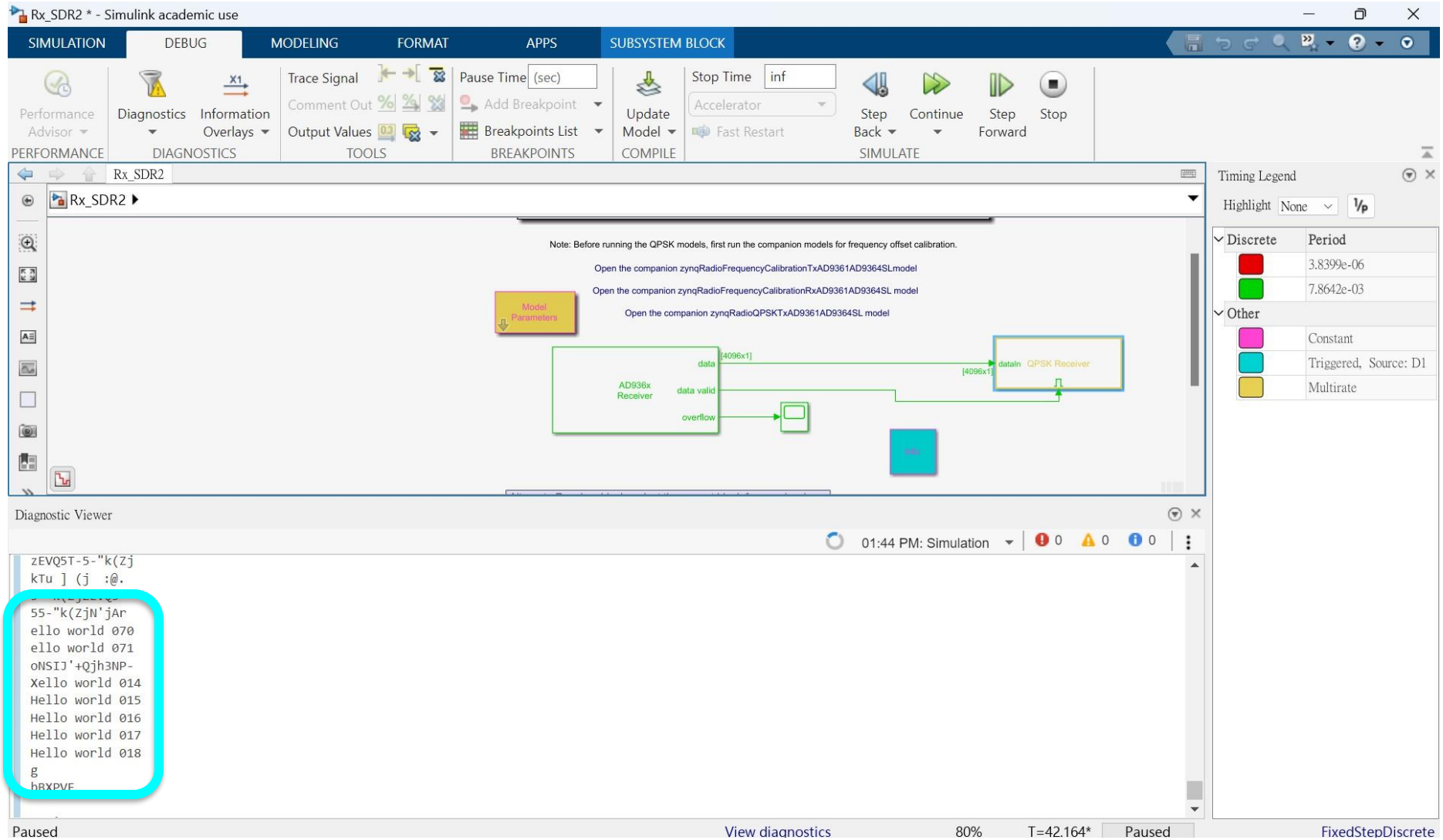


IB FDX QPSK OTA Test (3)

- OTA results appeared at Relay SDR

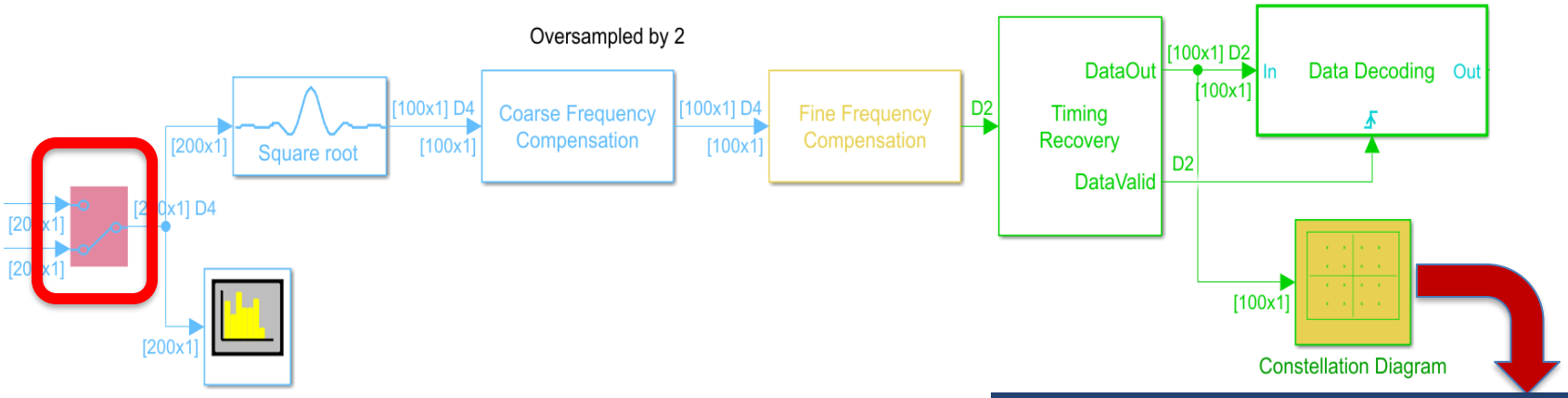


IB FDX QPSK OTA Test (4)



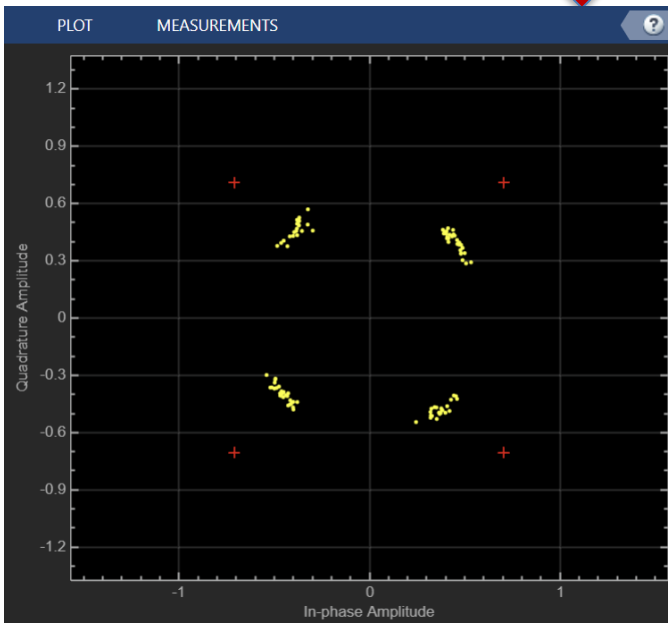
IB FDX QPSK OTA Test (5)

- Without SIC



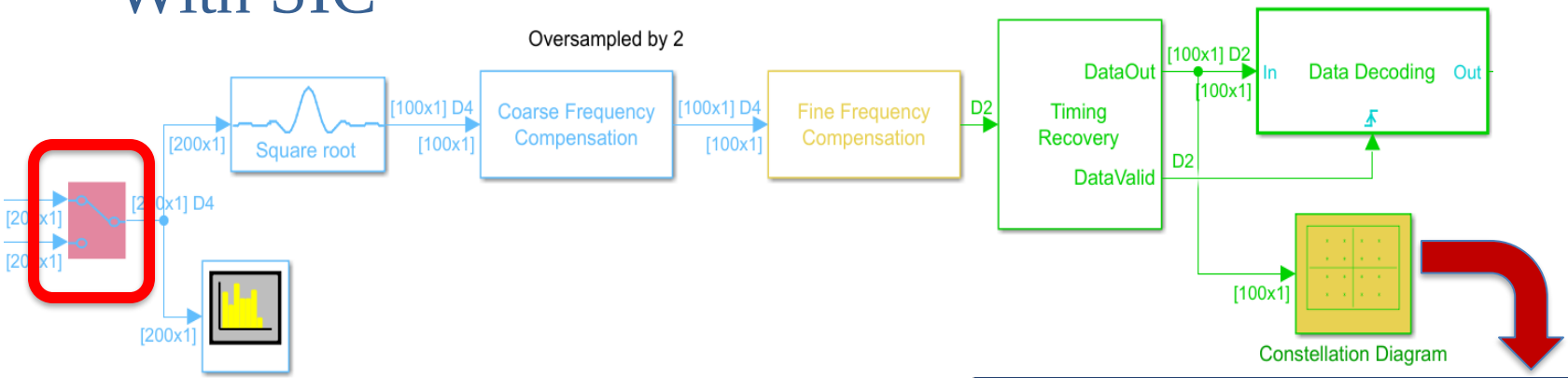
- Diagnostic Viewer @ Relay-SDR

```
  *li&fIYR6
M3$li&fIYR6
M3$li&fIYR6
MM3$li&fIYR6
MM3$liCm3  :]
hello world 085
Hello world 086
Hello wo_saE(1
~}C_0wl{>aoX
V}C 0wl{>aoX
```



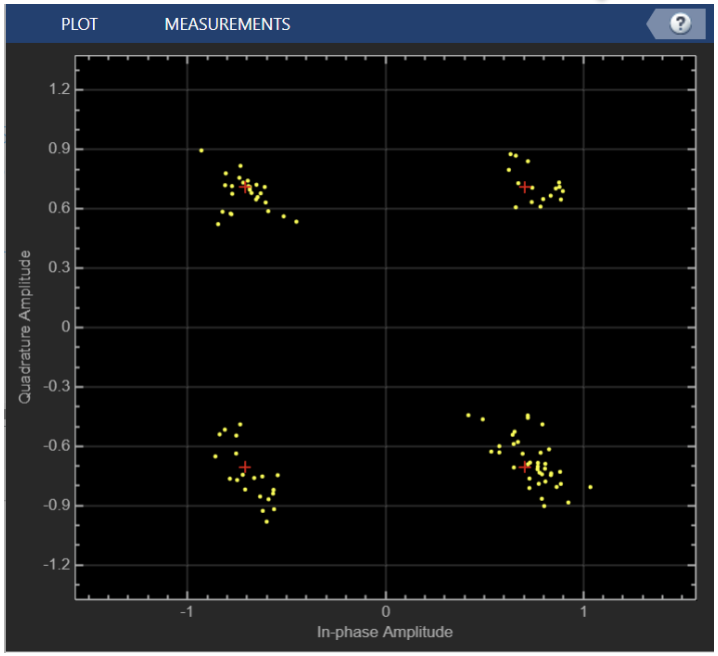
IB FDX QPSK OTA Test (6)

- With SIC



- Diagnostic Viewer @ Relay-SDR

```
Zmt[=o%[i6z-^K
v"+F%mdEu
Xello world 051
Hello world 052
Hello world 053
Hello world 054
Hello world 055
Hello world 056
Hello world 057
```



IB FDX QPSK OTA Test (7)

```
function y_GSD  = selfcorr(x_n, r1_n)
    if isempty(x_n)
        x_n = 0;
    end
    N = 200;
    La = 5;
    mu = 0.01;
    a_GSD = complex(zeros(La, 1));
    y_GSD = zeros(N, 1);
    for index = 1:N
        if index < La
            y_GSD(index) = r1_n(index);
        else
            disp('a_GSD:');
            disp(a_GSD);
            y_GSD(index) = r1_n(index)-a_GSD' * x_n(index:-1:index-La+1);
            a_GSD = a_GSD + mu*conj(y_GSD(index))*x_n(index:-1:index-La+1);
        end
    end
end
```

iagnostic Viewer

```
   0.0103 + 0.1122i
   0.0098 + 0.1084i
   0.0058 + 0.1060i
a_GSD:
   0.0133 + 0.1091i
   0.0129 + 0.1099i
   0.0139 + 0.1118i
   0.0134 + 0.1080i
   0.0094 + 0.1056i
```