institute iMdea networks

SISO/MIMO Matlab Gen/Decoder Model

Jesus Lacruz, Rafael Ruiz

Developing the Science of Networks

General Notes

- The functions are not optimized for fast execution but for verification of hardware / signal processing functions.
- The format of the files to be written in the RFSoC and Vadatech FPGAs match their corresponding firmware (MIMORPH and mm-FLEX papers, respectively).
- Functions assume symbol period of 1.76GSPS and 2 samples per symbol Tx and Rx datapaths in the FPGA (3.52GHz DA/AD clocks).
- SISO decoder is IEEE 802.11ad standard compliant i.e. can decode frames from COTS devices (C-PHY and SC).
- MIMO decoder can include EDMG-STF, EDMG-CEF from IEEE 802.11ay and header, data payload from IEEE 802.11ad standard (i.e. is NOT fully standard compliant).
- RFSoC Tx memories (in its basic form) are implemented using BRAM. Then it allows to store a single (short) packet of 32K samples MAX.

Notes about the folder structure (SISO/MIMO)

- In general, we divide the decoder folder structure as follows:
  - GEN_DATA_FUNCTIONS: it store the functions used to generate the Tx frames.
  - GEN_DATA: it stores the frames generated using the generation functions from GEN_DATA_FUNCTIONS folder. In general it includes subfolder for each FPGA destination (VADATECH / RFSOC).
  - CAPTURED_DATA: contains the IQ samples files captured using any of the FPGA platforms, oscilloscope or from simulated channels (Matlab). It contains subfolders for each “source”.
  - CAPTURED_DATA_FUNCTIONS: contains the functions used to decode the frames captured from any of the supported sources.

MIMO Generation (some notes) PART1

- Top file: GEN_IQ_SAMPLES_MIMO.m
- %%options
  - CIRC_SHIFT_PREAMBLE: if true, allow to cyclic rotate the STF of the 2º, 3º and 4º stream a fixed amount of samples. It is used to avoid unintended BP (according to IEEE 802.11ay standard).
  - ORTHOGONAL_STF: if true, include orthogonal 128-length Golay sequences for each spatial stream (Trying to mimic the EDMG-STF of IEEE 802.11ay). It helps to achieve better symbols synchronization, CFO estimation). If false, it uses the same Ga128 sequence for all spatial streams (like the L-STF defined in the IEEE 802.11ay).
  - Header and payload is generated from SC IEEE 802.11ad schemes. Note that we didn't implement C-PHY schemes for MIMO. User can select the PSDU length, MCS and number of spatial streams (nSTREAMS). Same function can be used to generate 2 and 4 spatial streams.
- %% Oversampling and filtering
  - We use a fixed SRRC filter for pulse shaping. Our FPGA platforms currently support 3.52GHz AD/DA clock, then we use 2 samples per symbol baseband waveforms.

MIMO Generation (some notes) PART2

- Top file: GEN_IQ_SAMPLES_MIMO.m
- %%Quantization
  - Vadatech FPGA integrates 16-bit DAC and RFSoC RFIP expects 16-bit samples, then we quantize the IQ samples with 16-bits in the range [-1,1), displaying an alert if samples exceed the range (overflow).
- %% Plot and save signal
  - IQ samples are saved in .mat file for comparison purposes, using them with simulated channels, etc.
  - If SAVE_RFSOC options is selected, the function save independent IQ samples files for each spatial stream in the folder / name selected by the user.

MIMO Decoding (some notes) PART1

- The RFSoC save the captured files using the format <*name*>_rx<*m*>_I and <*name*>_rx<*m*>_Q, being <*name*> the filename selected by the user and *m* the corresponding spatial stream.
- Top decoder file: TOP_DECODER_MIMO.m
- The variable FILES contains the information of the IQ samples to be decoded.
  - FILES{jj,1} → is the RESULTS_FOLDER where the decoded frames will be saved
  - FILES{jj,2} → is the SUBFOLDER where the captured date is located (using the CAPTURED_DATA/<source>/ as reference.
  - FILES{jj,3} → is the FILENAME of the captured IQ samples (without the _rx<m>_I or _Q suffix.
  - FILES{jj,4} → is the signal <source> used to capture the frames. Options valid are MATLAB and RFSOC
- SIGNAL.nSTREAMS → number of reveived streams. Valid options 2 and 4. For proper functioning must match the option selected in the frame generation.
- SIGNAL.ORTHOGONAL_STF → for proper functioning must match the option selected in the frame generation.
- SIGNAL.nsps = 2 → Decoder designed to operate at a fixed 2 samples per symbol
- SIGNAL.PLOT_RESULTS → if true, the decoder generate plot at different points of the decoding process (for debugging purposes).
- SIGNAL.SAVE_RESULTS → enable saving the decoded data to a .mat file (which name match the one of the captured IQ samples.

MIMO Decoding (some notes) PART2

- CONFIG_DECODER allows to configure different parameters for some of the block in the decoder datapath (SSRC filter, SNR computation, …)
- By default, in the FPGA firmware packets are “detected” using a packet detection block (details in the MIMORPH paper). The output of this block is saved along with the IQ samples. If SIGNAL.FPGA_PD is enabled, then the decoder use the packet detector flag to separate the IQ samples into “packets”. If disabled, a packet detection block is implemented in software.
- SEPARATE_FRAMES_MIMO_<m>x<m>: this function read the IQ samples file, separate the frames and save each packet in a separate file.
- From this point until the end of the decoder file, the script processes each one of the “detected” frames and save the decoding info in the same file (if save data is enabled).

MIMO Decoding (some notes) PART3

- PROCESS_STF: process the STF of the detected frame:
  - SSRC filter
  - CFO estimation / correction
  - SNR estimation
  - Symbol synchronization / downsampling
  - Boundary detection (coarse synchronization)
- PROCESS_L_CEF: process the L-CEF of the detected packet. Note that since the same sequence is used for all the spatial streams, no MIMO channel is estimated.
  - CIR estimation
  - Fine synchronization
  - Separate the L-Header, EDMG-CEF and Payload
  - Equalize the L-Header.
- PROCESS_11ad_HEADER: process the L-header (which follows the IEEE 802.11ad SC structure), which is sent using all the spatial streams. It allows to detect the payload length, MCS used, scrambler, etc.
- MIMO_EQ: compute the MIMO channel estimation (using the EDMG_CEF samples) and use the estimated channel to perform freq. domain equalization and phase tracking.
- PROCESS_11ad_DATA: process the payload of the packet. Note that the payload follows the 11ad structure for each one of the spatial streams.
  - Demapping
  - LDPC decoding
  - Descrambling