# OpenCPI
# Rx App Guide

Version 1.3

***Revision History***

| Revision | Description of Change | Date |
|---|---|---|
| v1.1 | Initial Release | 3/2017 |
| v1.2 | Updated for OpenCPI Release 1.2 | 8/2017 |
| v1.3 | Updated for OpenCPI Release 1.3 | 1/2018 |
| v1.3.1 | Updated for OpenCPI Release 1.3.1, including FMCOMMS2/3 support | 3/2018 |

# Table of Contents

# 1 Document Scope

This document describes the ANGRYPVIPER Receive demo application or "Rx App". It includes a description of the RX App application and instructions on how to setup the various supported hardware platforms, build and execution of the application.

# 2 Supported Hardware Setups

This app is supported on the following hardware configurations:

- Zedboard/FMCOMMS2
- Zedboard/FMCOMMS3
- x86/ML605/FMCOMMS2 in FMC LPC slot
- x86/ML605/FMCOMMS2 in FMC HPC slot
- x86/ML605/FMCOMMS3 in FMC LPC slot
- x86/ML605/FMCOMMS3 in FMC HPC slot
- Matchstiq-Z1
- Zedboard/Zipper/MyriadRF
- x86/Stratix IV GX development kit (230 Edition)/Zipper/MyriadRF in HSMC A slot
- x86/Stratix IV GX development kit (230 Edition)/Zipper/MyriadRF in HSMC B slot
- x86/ML605/Zipper/MyriadRF in FMC LPC slot
- x86/ML605/Zipper/MyriadRF in FMC HPC slot

# 3 Description

A block diagram of the RX app (for Stratix IV GX230 / Zipper on HSMC B specifically) can be seen in Figures 1 and 2. Complex samples from the ADC are ingested into the FPGA, processed, potentially timestamped, and written to file.

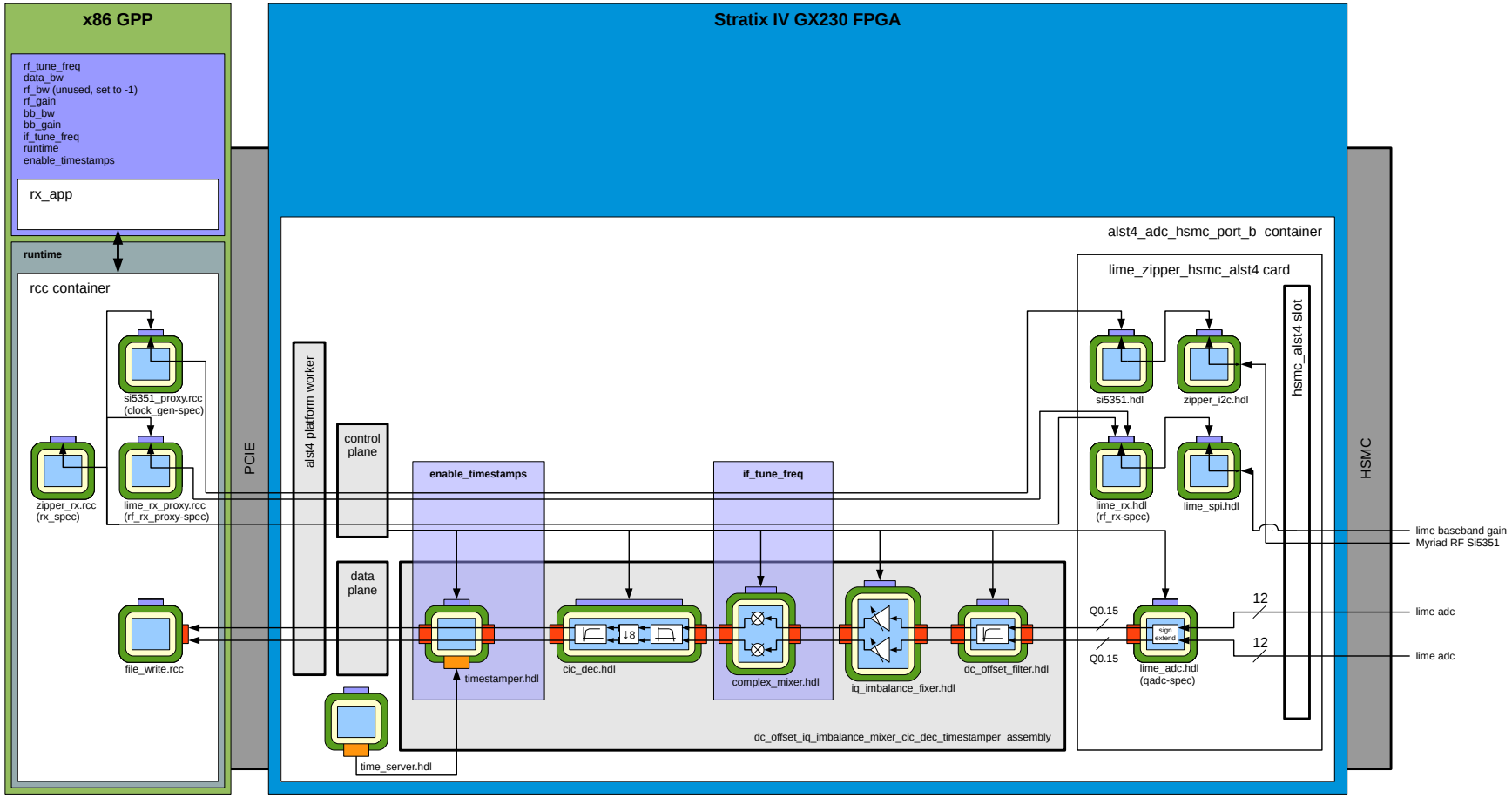



Figure 1: RX App Block Diagram for Stratix IV GX230 with Zipper on HSMC B (1/2)

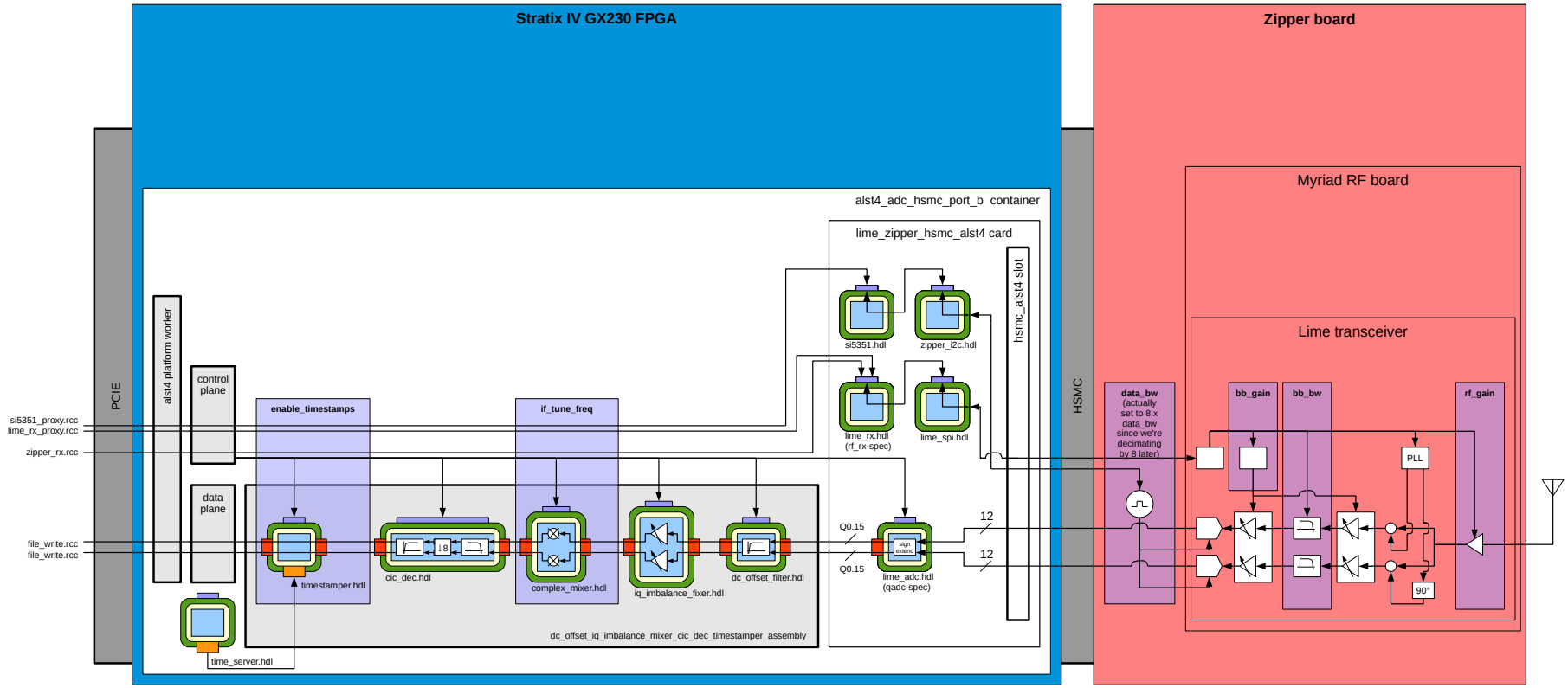



Figure 2: RX App Block Diagram for Stratix IV GX230 with Zipper on HSMC B (2/2)

# 4 Building the Application

## 4.1 Common Worker Dependencies

The following workers, sorted by component library name, must be built prior to building RX app, regardless of intended deployment hardware. See Appendix A for the parameter configurations used in the application, and see the individual component datasheets for more information.

- ocpi.core
  - file_write.rcc
- ocpi.assets.util_comps
  - timestamper.hdl
- ocpi.assets.dsp_comps
  - cic_dec.hdl
  - complex_mixer.hdl
  - iq_imbalance_fixer.hdl
  - dc_offset_filter.hdl

## 4.2 Hardware-Specific Worker Dependencies

The following workers, sorted by component library name, must be built prior to building RX app. See Appendix A for the parameter configurations used in the application, and see the individual component datasheets for more information and build instructions.

**Matchstiq-Z1**

- ocpi.assets.devices
  - lime_adc.hdl
  - lime_rx_proxy.rcc
  - lime_rx.hdl
  - lime_spi.hdl
  - pca9535.hdl
  - si5338_proxy.rcc
  - si5338.hdl
  - tmp100_proxy.rcc
  - tmp100.hdl
- ocpi.assets.platforms.matchstiq_z1.devices
  - matchstiq_z1_rx.rcc
  - matchstiq_z1_avr_proxy.rcc
  - matchstiq_z1_avr.hdl
  - matchstiq_z1_pca9535_proxy.rcc
  - matchstiq_z1_i2c.hdl

**Zipper/MyriadRF Transceiver Card**

- ocpi.assets.devices
  - lime_adc.hdl
  - lime_rx_proxy.rcc
  - lime_rx.hdl
  - lime_spi.hdl
  - si5351_proxy.rcc
  - si5351.hdl
- ocpi.assets.cards
  - zipper_rx.rcc

**FMCOMMS2/3 Transceiver Cards**

- ocpi.assets.devices
  - ad9361_adc.hdl
  - ad9361_adc_sub.hdl
  - ad9361_config.hdl
  - ad9361_config_proxy.rcc
  - ad9361_dac.hdl
  - ad9361_dac_sub.hdl
  - ad9361_data_sub.hdl
  - ad9361_spi.hdl
- ocpi.assets.cards
  - fmcomms_2_3_i2c.hdl
  - fmcomms_2_3_rx.rcc

Additionally, platform support for the Matchstiq-Z1/Zedboard/Stratix IV/ML605 must also be built prior to building the RX app. See the respective Platform Data Sheet for more information and build instructions.

## 4.3 HDL Assembly and HDL Container

The FPGA portion of the application consists of the dc_offset_iq_imbalance_mixer_cic_dec_timestamper HDL assembly and the appropriate Matchstiq-Z1/Zedboard/Stratix IV/ML605 HDL container file. The HDL assembly instances the signal processing and timestamping components. The HDL container has three primary functions in this application:

1) Connects HDL assembly input to the ADC hardware for gathering IQ data

2) Connects HDL assembly output to the processor for writing data to disk

3) Instances command and control SPI/I2C HDL Device Workers required to configure the RF front end

## 4.4 Performance and Resource Utilization

| Hardware Configuration | FPGA | Registers (typ) | LUTs (typ) | Fmax (typ) | Memory/Special Functions | Design Suite |
|---|---|---|---|---|---|---|
| Matchstiq-Z1 | XC7Z020-1-CLG484 | 9813 | 9918 | 100 MHz | DSP48E1s=15 (7%)<br>RAMB36E1s=21<br>BUFGs=2<br>BUFGCTRLs=2 | Vivado 2017.1 |
| ML605/FMCOMMS2/3 HPC | XC6VLX240T-1-FF1156 | 13,972 (4%) | 18,149 (12%) | 125 MHz | DSP48E1s=15 (1%)<br>BUFG/BUFGCTRLs=6 | ISE 14.7 |
| ML605/FMCOMMS2/3 LPC | XC6VLX240T-1-FF1156 | 13,972 (4%) | 18,103 (12%) | 125 MHz | DSP48E1s=15 (1%)<br>BUFG/BUFGCTRLs=6 | ISE 14.7 |
| ML605/Zipper HPC | XC6VLX240T-1-FF1156 | 13850 | 18325 | 125 MHz | DSP48E1s=15 (1%)<br>BUFG/BUFGCTRLs=7 | ISE 14.7 |
| ML605/Zipper LPC | XC6VLX240T-1-FF1156 | 13850 | 18404 | 125 MHz | DSP48E1s=15 (1%)<br>BUFG/BUFGCTRLs=7 | ISE 14.7 |
| Stratix IV/Zipper HSMCA | EP4SGX230K-C2-F40 | 41017 | 27147 | 125 MHz | DSP18x18s=26 (2%)<br>GXB Receiver PCSs=4<br>GXB Transmitter PCSs=4<br>GXB Receiver PMAs=4<br>GXB Transmitter PMAs=4<br>PLLs=1 | Quartus 15.1.0 |
| Stratix IV/Zipper HSMCB | EP4SGX230K-C2-F40 | 41017 | 27147 | 125 MHz | DSP18x18s=26 (2%)<br>GXB Receiver PCSs=4<br>GXB Transmitter PCSs=4<br>GXB Receiver PMAs=4<br>GXB Transmitter PMAs=4<br>PLLs=1 | Quartus 15.1.0 |
| Zedboard/FMCOMMS2/3 | XC7Z020-1-CLG484 | 8,882 (9%) | 8,865 (17%) | 100 MHz | DSP48E1s=15 (7%)<br>RAMB36E1s=18<br>BUFGs=1<br>BUFGCTRLs=2 | Vivado 2017.1 |
| Zedboard/Zipper | XC7Z020-1-CLG484 | 8759 | 8872 | 100 MHz | DSP48E1s=15 (7%)<br>RAMB36E1s=21<br>BUFGs=2<br>BUFGCTRLs=2 | Vivado 2017.1 |

## 4.5 Executable

The software portion of the application consists of a C++ program written using the OpenCPI C++ API, RCC endpoint proxy workers for command and control functionality, and the file_write.rcc RCC app worker for capturing data. For more implementation details on the endpoint proxy, see the matchstiq_z1_rx.rcc, zipper_rx.rcc, or fmcomms_2_3_rx.rcc component datasheets. The C++ program instantiates and OpenCPI application object using one of the application XML files: rx_fmcomms_2_app.xml, rx_fmcomms_3_app.xml, rx_matchstiq_z1_app.xml, rx_zipper_app.xml. Each of these files contain all of the property settings for the components in the application, except the configuration of the endpoint proxy for controlling RF hardware. These settings are passed on the command line to the approriate endpoint proxy worker and set using the ACI.

For optimal throughput, the file write RCC component writes directly to RAM via a Linux RAMdisk at runtime, and the application copies the data from RAM to a file in the application directory (odata/rx_app_raw.out) RX data capture is complete. The executable creates an additional shortened copy of this data (odata/rx_app_shortened.out) which omits some number of bytes from the beginning of the data. This is done because some of the components include feedback loops which require some setup time before functioning as desired.

# 5 Testing the Application

## 5.1 Sample Test Setup

To verify functionality of the application, a transmitter broadcasting a known signal is needed. Optionally, a spectrum analyzer to compare the transmitter output to the received data is a useful verification tool.

Figure 3 shows an example test setup for the RX app. It uses GNUradio and the Ettus N210 SDR to inject data into the Platform Under Test. GNUradio is available to download in the default CENTOS 7 repository, and a sample block diagram for transmitting random FSK data is included with this application (gnuradio/usrp_fsk.grc). The transmitter output is also split off to a spectrum analyzer.

A recommended alternative to using the Ettus N210 would be an arbitrary RF signal generator.

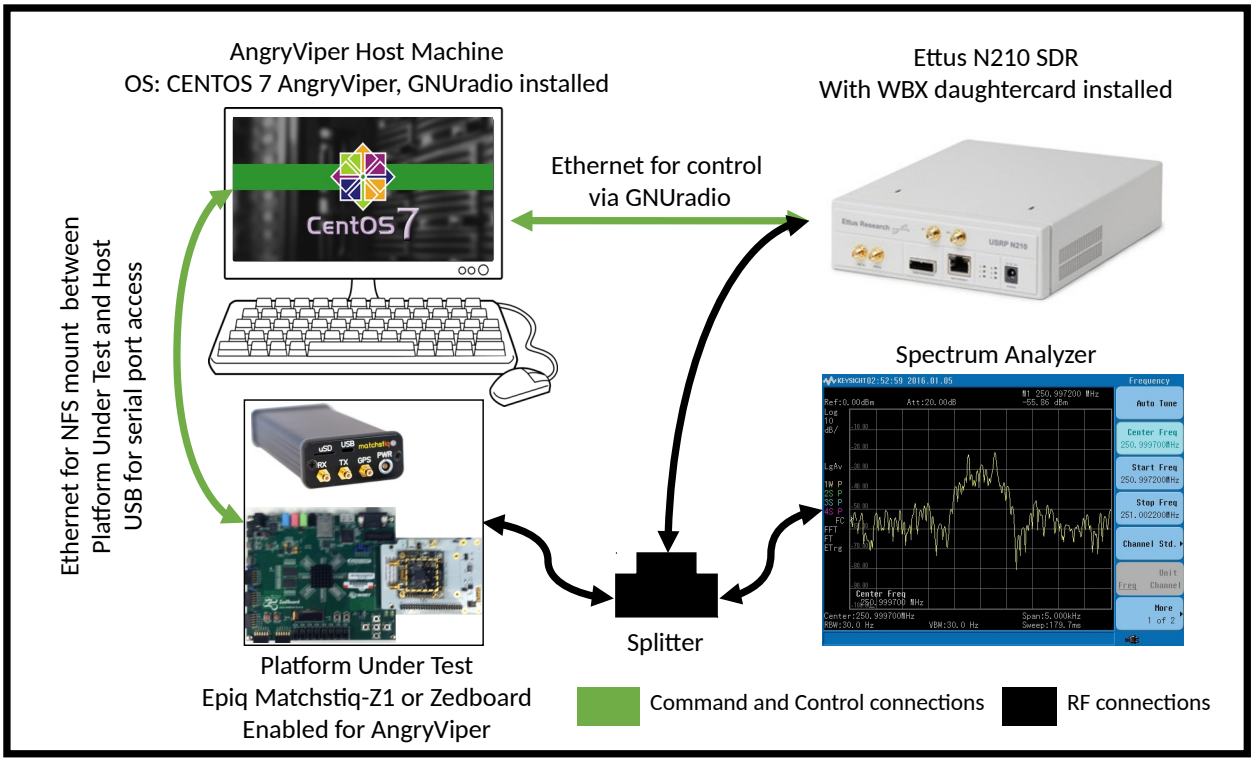



Figure 3: RX App Test Setup

## 5.2 make show

In order to test the application, `make show` can be run from the `applications/rx_app` directory. This provides instructions (for Zynq-Based Platforms) for setting `OCPI_LIBRARY_PATH` on the hardware platform and then running the application. Finally, it explains how to verify the output data on the development computer. The following sections provide further insight into these instructions.

## 5.3 Artifacts

Before running the application, the location of the required deployable artifacts must be specified in the OCPI_LIBRARY_PATH environment variable. Each RCC worker and the FPGA image exist as an artifact which should be included. Furthermore, artifacts differ depending on which mode the application is to be run in. Appendix B includes a list of the artifacts required for each platform and mode.

## 5.4 Arguments to executable

There are eleven arguments to the RX app executable. They primarily configure the RF front end of the Platform Unit Test using the matchstiq_z1_rx.rcc/zipper_rx.rcc components. Additionally, the application can be configured by setting properties in the application XML file: rx_app.xml. Descriptions of properties can be found in the individual component datasheets. Valid ranges for each argument can be printed out by running the executable with no arguments.

The arguments to the executable are summarized in the below table:

| Argument | Description |
|---|---|
| rf_tune_freq | RF (analog) tuning frequency in MHz |
| data_bw | Effective sample rate of the frontend ADC (as well as the data being written to file) in MS/s |
| rf_bw | Analog RF filter bandwidth in MHz |
| rf_gain | RF (analog) gain in dB |
| bb_bw | Analog filter bandwidth of the basebanded (downconverted) signal path in MHz |
| bb_gain | Gain (analog) of basebanded (downconverted) signal path in dB |
| if_tune_freq | Tuning frequency in MHz of the HDL mixer which mixes (downconverts) the digitized data stream |
| runtime | Runtime of app in seconds |
| enable_timestamps | Enable timestamp insertion in between messages |
| frontend | Only required for Zedboard or ML605, (FMCOMMS2 or FMCOMMS3 or zipper) |
| sma_channel | (optional) specify which PCB SMA is used when FMCOMMS2/3 is used (RX1A or RX2A) |

## 5.5 Library Path Requirements

Prior to running the application, the environment variable OCPI_LIBRARY_PATH must include the following directories:

- ocpi.core component RCC library location
- ocpi.assets bitstream directory location
- ocpi.assets.devices library location
- ocpi.assets component RCC library location

Matchstiq-Z1 additionally requires the following:

- ocpi.assets.platforms.matchstiq_z1.devices library location

Note that the Stratix IV GX230/Zipper/MyriadRF and ML605/Zipper/MyriadRF hardware setups require the intended slot-specific bitstream's file location to occur first in OCPI_LIBRARY_PATH. This is necessary because ocpirun's aritifact compatibility test can not currently differentiate between slot-connected device workers for multiple bitstreams that contain the same device worker, in the scenario where what differentiates the bitstreams is the device worker's slot connectivity. Examples of library paths that could be used can be seen below:

Examples of library paths that could be used can be seen below:

Note: All example paths are relative to the applications/rx_app/ directory.

**Recommended Library Path for Matchstiq-Z1 or Zedboard**

```
OCPI_LIBRARY_PATH=$OCPI_PROJECT_REGISTRY_DIR/ocpi.core/exports/lib:$OCPI_PROJECT\
_REGISTRY_DIR/ocpi.assets/exports/lib
```

**Recommended Library Path for Stratix IV GX230/Zipper in HSMC A**

```
OCPI_LIBRARY_PATH=$OCPI_PROJECT_REGISTRY_DIR/ocpi.core/exports/lib:$OCPI_PROJECT\
_REGISTRY_DIR/ocpi.assets/exports/lib:../../hdl/assemblies/dc_offset_iq_imbalanc\
e_mixer_cic_dec_timestamper/container-dc_offset_iq_imbalance_mixer_cic_dec_times\
tamper_alst4_alst4_zipper_hsmc_alst4_port_a_rx_cnt_1rx_0tx_thruasm_zipper_hsmc_a\
_alst4/
```

**Recommended Library Path for Stratix IV GX230/Zipper in HSMC B**

```
OCPI_LIBRARY_PATH=$OCPI_PROJECT_REGISTRY_DIR/ocpi.core/exports/lib:$OCPI_PROJECT\
_REGISTRY_DIR/ocpi.assets/exports/lib:../../hdl/assemblies/dc_offset_iq_imbalanc\
e_mixer_cic_dec_timestamper/container-dc_offset_iq_imbalance_mixer_cic_dec_times\
tamper_alst4_alst4_zipper_hsmc_alst4_port_b_rx_cnt_1rx_0tx_thruasm_zipper_hsmc_b\
_alst4/
```

**Recommended Library Path for ML605/Zipper in FMC HPC**

```
OCPI_LIBRARY_PATH=$OCPI_PROJECT_REGISTRY_DIR/ocpi.core/exports/lib:$OCPI_PROJECT\
_REGISTRY_DIR/ocpi.assets/exports/lib:../../hdl/assemblies/dc_offset_iq_imbalanc\
e_mixer_cic_dec_timestamper/container-dc_offset_iq_imbalance_mixer_cic_dec_times\
tamper_ml605_ml605_zipper_fmc_hpc_rx_cnt_1rx_0tx_thruasm_zipper_hpc_ml605/
```

**Recommended Library Path for ML605/Zipper in FMC LPC**

```
OCPI_LIBRARY_PATH=$OCPI_PROJECT_REGISTRY_DIR/ocpi.core/exports/lib:$OCPI_PROJECT\
_REGISTRY_DIR/ocpi.assets/exports/lib:../../hdl/assemblies/dc_offset_iq_imbalanc\
e_mixer_cic_dec_timestamper/container-dc_offset_iq_imbalance_mixer_cic_dec_times\
tamper_ml605_ml605_zipper_fmc_lpc_rx_cnt_1rx_0tx_thruasm_zipper_lpc_ml605/
```

**Recommended Library Path for ML605/FMCOMMS2/3 in FMC HPC**

```
OCPI_LIBRARY_PATH=$OCPI_PROJECT_REGISTRY_DIR/ocpi.core/exports/lib:$OCPI_PROJECT\
_REGISTRY_DIR/ocpi.assets/exports/lib:../../hdl/assemblies/dc_offset_iq_imbalanc\
e_mixer_cic_dec_timestamper/container-dc_offset_iq_imbalance_mixer_cic_dec_times\
tamper_ml605_cfg_1rx_0tx_fmcomms_2_3_hpc_lvds_cnt_1rx_0tx_thruasm_fmcomms_2_3_hp\
c_LVDS_ml605/
```

**Recommended Library Path for ML605/FMCOMMS2/3 in FMC LPC**

```
OCPI_LIBRARY_PATH=$OCPI_PROJECT_REGISTRY_DIR/ocpi.core/exports/lib:$OCPI_PROJECT\
_REGISTRY_DIR/ocpi.assets/exports/lib:../../hdl/assemblies/dc_offset_iq_imbalanc\
e_mixer_cic_dec_timestamper/container-dc_offset_iq_imbalance_mixer_cic_dec_times\
tamper_ml605_cfg_1rx_0tx_fmcomms_2_3_lpc_lvds_cnt_1rx_0tx_thruasm_fmcomms_2_3_lp\
c_LVDS_ml605/
```

## 5.6 Expected results

A python script is included with the application for plotting the received data in both the time and frequency domain. Using the test setup shown above and the default settings for the GNUradio FSK block diagram, run the application with the following arguments:

**FMCOMMS2/3**

```
# Usage is:
# ./target-linux-x13_3-arm/rx_app rf_tune_freq data_bw rf_bw rf_gain bb_bw bb_gain if_tune_freq runtime enable_timestamps
  ./target-linux-x13_3-arm/rx_app 1000         0.256   -1    12      1     -1      0.256        1       1
```

**Matchstiq-Z1**

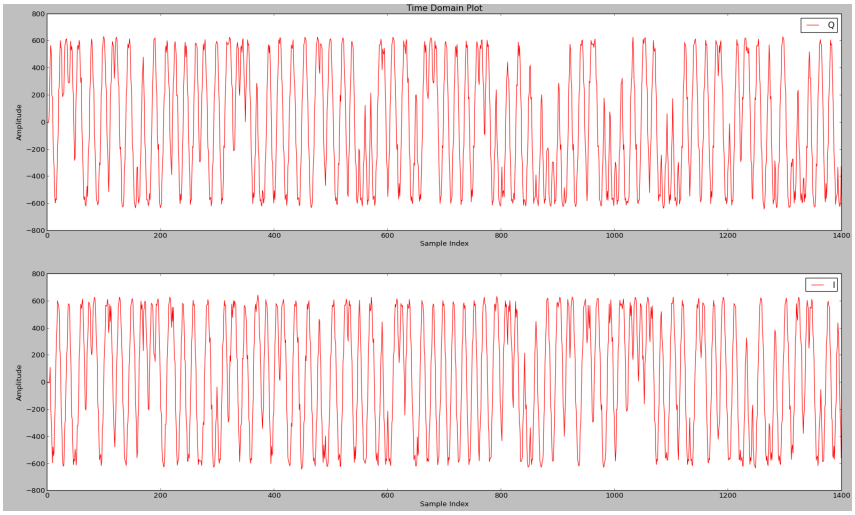
```
# Usage is:
# ./target-linux-x13_3-arm/rx_app rf_tune_freq data_bw rf_bw rf_gain bb_bw bb_gain if_tune_freq runtime enable_timestamps
  ./target-linux-x13_3-arm/rx_app 1000         0.256   400   10      0.75  51      0.256        1       1
```

**Zipper/Myriad RF card**

```
# Usage is:
# ./target-linux-c6-x86_64/rx_app rf_tune_freq data_bw rf_bw rf_gain bb_bw bb_gain if_tune_freq runtime enable_timestamps frontend
  ./target-linux-c6-x86_64/rx_app 1000         0.256   -1    6       0.75  51      0.256        1       1                 zipper
```

The output file can then be plotted with the python script with the following syntax and the output can be seen below:

```
python ./scripts/plotAndFftAndTime.py odata/rx_app_raw.out complex 18000 256000 16352
```




Alternatively, the shortened file can be plotted which will ignore potentially unwanted startup data:

```
python ./scripts/plotAndFftAndTime.py odata/rx_app_shortened.out complex 18000 256000 16352
```

The default sample rate for the GNUradio FSK block diagram is 512 kS/s. It is recommended that when using RX app with this input signal that a sample rate close to 512 kS/s be used. Higher sample rates are still valid, but may produce plots that look drastically different than those shown here.

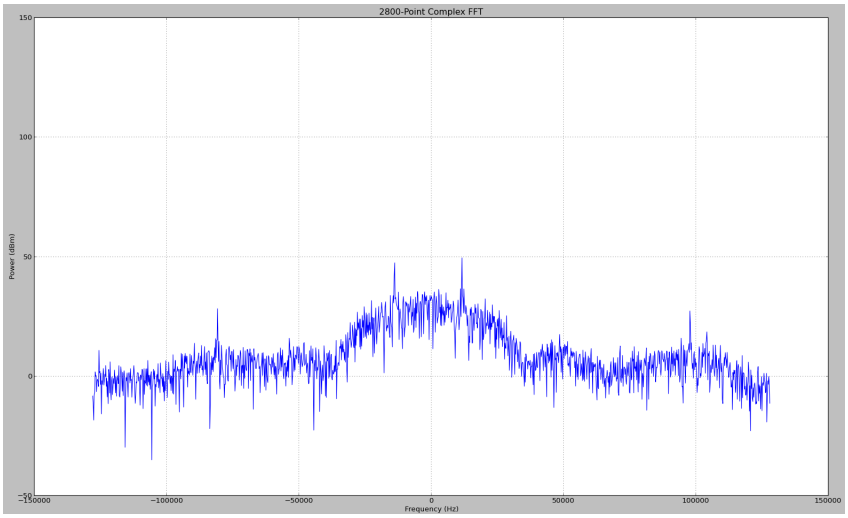



Figure 4: Output of RX app

Timestamps are embedded, optionally, in the output file, and in addition to plotting, the script parses out and prints the timestamps. An example output gathered using the syntax above:

```
Timestamp at index:  000000000 :  1.0728292 Seconds:  0x1 Fraction:  0x12a4eec4
Timestamp at index:  000008180 :  1.0887978 Seconds:  0x1 Fraction:  0x16bb73ba ('Delta:  0.0159686', 'Expected:, 0.0159688')
Timestamp at index:  000016360 :  1.1047664 Seconds:  0x1 Fraction:  0x1ad1f906 ('Delta:  0.0159686', 'Expected:, 0.0159688')
```

A small discrepancy (+/- 10) between Delta and Expected is typical. The difference is an artifact of the resolution of the fractional part of the timestamp applied in the timestamper HDL component. More information can be found in the timestamper component datasheet.

## 5.7 Using a RF Signal Generator

As mentioned earlier, an arbitrary RF signal generator can be used with RX app instead of the Ettus N210. Below is an example using a signal generator and a Matchstiq-Z1 or Zed/Zipper.

In this example, the signal generator is set to 1.001250 GHz with an amplitude of -60 dBm (Matchstiq-Z1) or -40 dBm (Zed/Zipper). The following parameters can be passed to the executable:

**FMCOMMS2/3**

```
# Usage is:
# ./target-linux-x13_3-arm/rx_app rf_tune_freq data_bw rf_bw rf_gain bb_bw bb_gain if_tune_freq runtime enable_timestamps
  ./target-linux-x13_3-arm/rx_app 1000         2.5     -1    12      2.5   -1      1            1       1
```

**Matchstiq-Z1**

```
# Usage is:
# ./target-linux-x13_3-arm/rx_app rf_tune_freq data_bw rf_bw rf_gain bb_bw bb_gain if_tune_freq runtime enable_timestamps frontend
  ./target-linux-x13_3-arm/rx_app 1000         2.5     400   10      1.25  51      1            1       1
```

**Zed/Zipper**

```
# Usage is:
# ./target-linux-x13_3-arm/rx_app rf_tune_freq data_bw rf_bw rf_gain bb_bw bb_gain if_tune_freq runtime enable_timestamps frontend
  ./target-linux-x13_3-arm/rx_app 1000         2.5     -1   6      1.25  51      1            1       1                 zipper
```

Here we plot output data:

```
python ./scripts/plotAndFftAndTime.py odata/rx_app_shortened.out complex 65536 2500000 16352
```

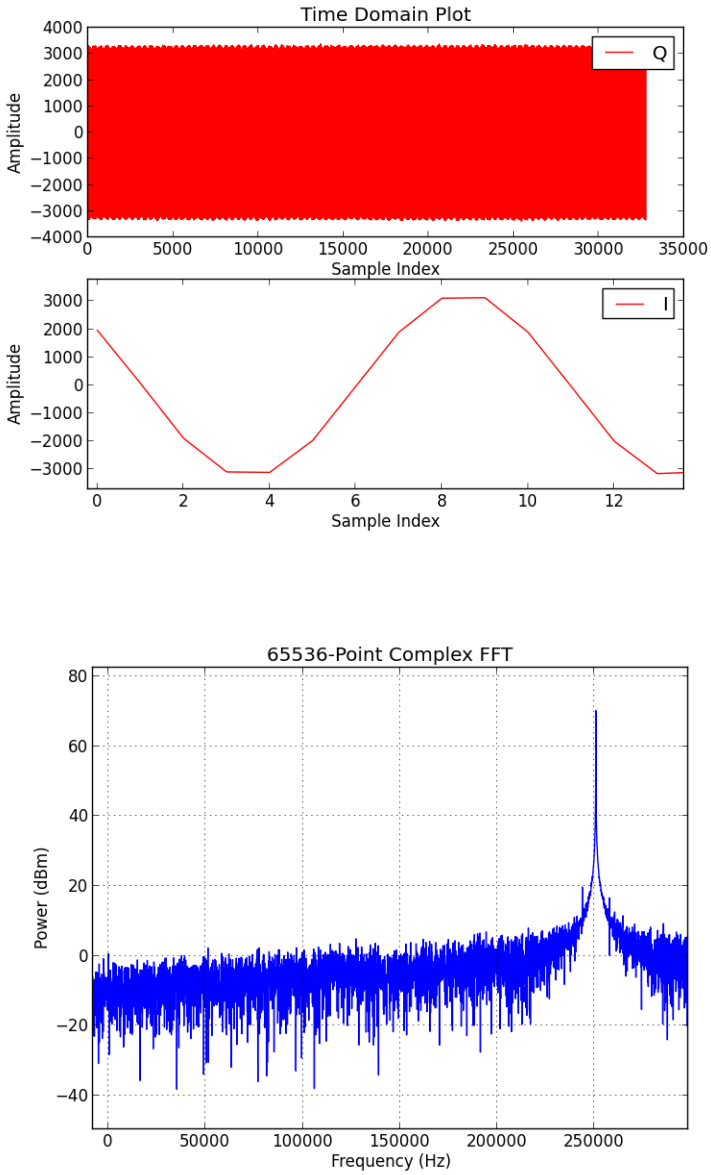



Figure 5: Output of RX app for Matchstiq-Z1

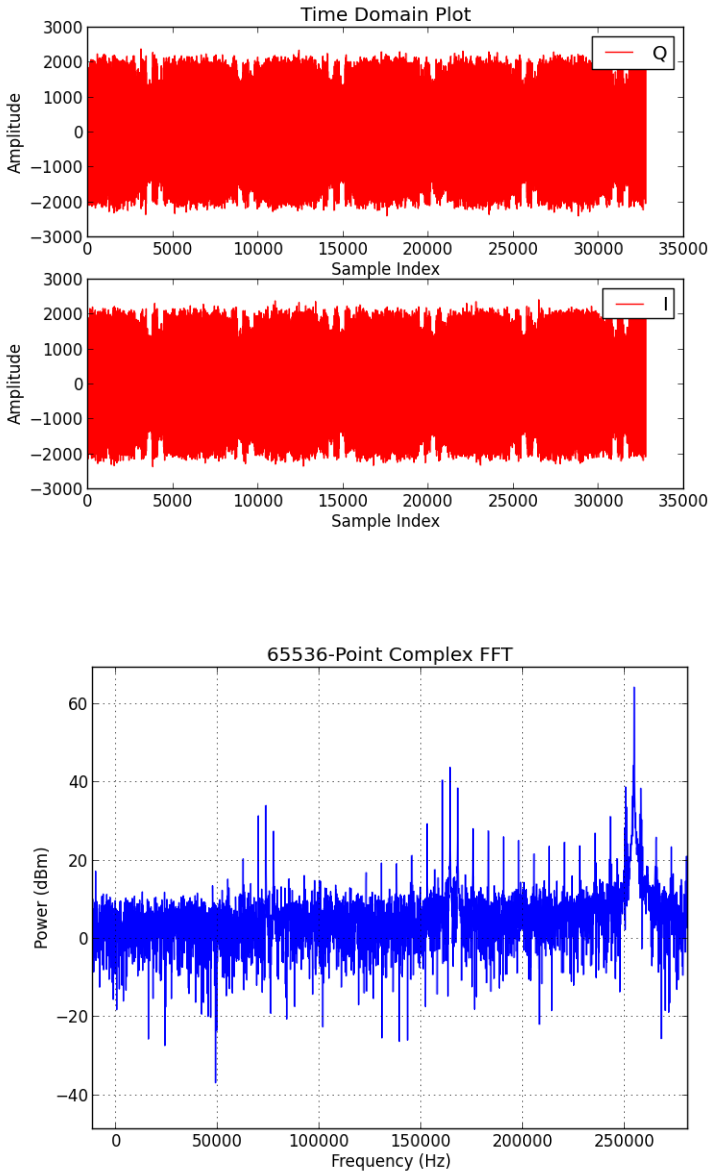



Figure 6: Output of RX app for Zed/Zipper

## 5.8 Known Issues

- For more information on known limitations when using the Zipper-related platforms (Zedboard, Stratix IV, ML605), see the document Myriad-RF_1_Zipper_Limitations included with this project.
- If the path `/var/volatile` does not exist or requires root permission to write to, you will need to modify the ACI and the application XML to use a different directory for writing data. This involves simply finding and replacing `/var/volatile` with a different directory in the `.cxx` and `.xml` files. Failing to make this change when necessary may result in a segmentation fault error at application runtime.
- On x86 host machines with more than one Stratix IV and/or ML605s plugged into PCIe slots, this app will assume that the first found Stratix IV/ML605 has a Zipper/MyriadRF plugged in. The first found Stratix IV/ML605 will be used during execution. While there are means to address this issue, they have not been implemented for the current release.

# 6 Appendix A: Worker Parameters

**Zedboard (with FMCOMMS2/3 card)**

- cic_dec.hdl
  - N = 3
  - M = 1
  - R = 8
  - DIN_WIDTH = 16
  - ACC_WIDTH = 25
  - DOUT_WIDTH = 16
- complex_mixer.hdl
  - NCO_DATA_WIDTH_p = 12
  - INPUT_DATA_WIDTH_p = 12
  - CORDIC_STAGES_p = 16
  - PEAK_MONITOR_p = true
- iq_imbalance_fixer.hdl
  - DATA_WIDTH_p = 16
  - ACC_PREC_p = 34
  - PEAK_MONITOR_p = true
- dc_offset_filter.hdl
  - DATA_WIDTH_p = 16
  - PEAK_MONITOR_p = true
- fmcomms_2_3_i2c.hdl
  - CP_CLK_FREQ_p = 100e6
  - FMC_GA1 = 0
  - FMC_GA0 = 0
- ad9361_spi.hdl
  - CP_CLK_FREQ_HZ_p = 100e6
- ad9361_data_sub.hdl
  - LVDS_p = true
  - DATA_CLK_Delay = 3
  - RX_Data_Delay = 0

**ML605 (with FMCOMMS2/3 card in FMC HPC slot)**

- cic_dec.hdl
  - N = 3
  - M = 1
  - R = 8
  - DIN_WIDTH = 16
  - ACC_WIDTH = 25
  - DOUT_WIDTH = 16
- complex_mixer.hdl
  - NCO_DATA_WIDTH_p = 12
  - INPUT_DATA_WIDTH_p = 12
  - CORDIC_STAGES_p = 16
  - PEAK_MONITOR_p = true
- iq_imbalance_fixer.hdl
  - DATA_WIDTH_p = 16
  - ACC_PREC_p = 34
  - PEAK_MONITOR_p = true
- dc_offset_filter.hdl
  - DATA_WIDTH_p = 16
  - PEAK_MONITOR_p = true
- fmcomms_2_3_i2c.hdl
  - CP_CLK_FREQ_p = 125e6
  - FMC_GA1 = 0
  - FMC_GA0 = 0
- ad9361_spi.hdl
  - CP_CLK_FREQ_HZ_p = 125e6
- ad9361_data_sub.hdl
  - LVDS_p = true
  - DATA_CLK_Delay = 2
  - RX_Data_Delay = 0

**ML605 (with FMCOMMS2/3 card in FMC LPC slot) Matchstiq-Z1**

- cic_dec.hdl
  - N = 3
  - M = 1
  - R = 8
  - DIN_WIDTH = 16
  - ACC_WIDTH = 25
  - DOUT_WIDTH = 16
- complex_mixer.hdl
  - NCO_DATA_WIDTH_p = 12
  - INPUT_DATA_WIDTH_p = 12
  - CORDIC_STAGES_p = 16
  - PEAK_MONITOR_p = true
- iq_imbalance_fixer.hdl
  - DATA_WIDTH_p = 16
  - ACC_PREC_p = 34
  - PEAK_MONITOR_p = true
- dc_offset_filter.hdl
  - DATA_WIDTH_p = 16
  - PEAK_MONITOR_p = true
- fmcomms_2_3_i2c.hdl
  - CP_CLK_FREQ_p = 125e6
  - FMC_GA1 = 1
  - FMC_GA0 = 0
- ad9361_spi.hdl
  - CP_CLK_FREQ_HZ_p = 125e6
- ad9361_data_sub.hdl
  - LVDS_p = true
  - DATA_CLK_Delay = 2
  - RX_Data_Delay = 0
- cic_dec.hdl
  - N = 3
  - M = 1
  - R = 8
  - DIN_WIDTH = 16
  - ACC_WIDTH = 25
  - DOUT_WIDTH = 16
- complex_mixer.hdl
  - NCO_DATA_WIDTH_p = 12
  - INPUT_DATA_WIDTH_p = 12
  - CORDIC_STAGES_p = 16
  - PEAK_MONITOR_p = true
- iq_imbalance_fixer.hdl
  - DATA_WIDTH_p = 16
  - ACC_PREC_p = 34
  - PEAK_MONITOR_p = true
- dc_offset_filter.hdl
  - DATA_WIDTH_p = 16
  - PEAK_MONITOR_p = true
- lime_adc.hdl
  - DRIVE_CLK_p = false
  - USE_CLK_IN_p = false
  - USE_CTL_CLK_p = false
  - USE_CLK_OUT_p = true
- si5338.hdl
  - CLKIN_PRESENT_p = true
  - CLKIN_FREQ_p = 3.072e7
  - XTAL_PRESENT_p = false
  - XTAL_FREQ_p = 0
  - OUTPUTS_PRESENT_p = 1,0,0,0
  - INTR_CONNECTED_p = false
- matchstiq_z1_i2c.hdl
  - NUSERS_p = 5
  - SLAVE_ADDRESS_p = 0x45,0x71,0x48,0x21,0x20
  - CLK_CNT_p = 199

**Zedboard (with Zipper/Myriad-RF card)**

- cic_dec.hdl
  - N = 3
  - M = 1
  - R = 8
  - DIN_WIDTH = 16
  - ACC_WIDTH = 25
  - DOUT_WIDTH = 16
- complex_mixer.hdl
  - NCO_DATA_WIDTH_p = 12
  - INPUT_DATA_WIDTH_p = 12
  - CORDIC_STAGES_p = 16
  - PEAK_MONITOR_p = true
- iq_imbalance_fixer.hdl
  - DATA_WIDTH_p = 16
  - ACC_PREC_p = 34
  - PEAK_MONITOR_p = true
- dc_offset_filter.hdl
  - DATA_WIDTH_p = 16
  - PEAK_MONITOR_p = true
- lime_adc.hdl
  - DRIVE_CLK_p = false
  - USE_CLK_IN_p = true
  - USE_CTL_CLK_p = false
  - USE_CLK_OUT_p = false
- si5351.hdl
  - CLKIN_PRESENT = true
  - CLKIN_FREQ = 3.072e7
  - XTAL_PRESENT = false
  - XTAL_FREQ = 0
  - VC_PRESENT = false
  - OUTPUTS_PRESENT = 0,0,1,1,1,1,0,0
  - OEB_MODE = low
  - INTR_CONNECTED = false
- zipper_i2c.hdl
  - NUSERS_p = 2

**Stratix IV GX230 (with Zipper/Myriad-RF card)**

- cic_dec.hdl
  - N = 3
  - M = 1
  - R = 8
  - DIN_WIDTH = 16
  - ACC_WIDTH = 25
  - DOUT_WIDTH = 16
- complex_mixer.hdl
  - NCO_DATA_WIDTH_p = 12
  - INPUT_DATA_WIDTH_p = 12
  - CORDIC_STAGES_p = 16
  - PEAK_MONITOR_p = true
- iq_imbalance_fixer.hdl
  - DATA_WIDTH_p = 16
  - ACC_PREC_p = 34
  - PEAK_MONITOR_p = true
- dc_offset_filter.hdl
  - DATA_WIDTH_p = 16
  - PEAK_MONITOR_p = true
- lime_adc.hdl
  - DRIVE_CLK_p = false
  - USE_CLK_IN_p = true
  - USE_CTL_CLK_p = false
  - USE_CLK_OUT_p = false
- si5351.hdl
  - CLKIN_PRESENT = true
  - CLKIN_FREQ = 3.072e7
  - XTAL_PRESENT = false
  - XTAL_FREQ = 0
  - VC_PRESENT = false
  - OUTPUTS_PRESENT = 0,0,1,1,1,1,0,0
  - OEB_MODE = low
  - INTR_CONNECTED = false
- zipper_i2c.hdl
  - NUSERS_p = 2

**ML605 (with Zipper/Myriad-RF card)**

- cic_dec.hdl
  - N = 3
  - M = 1
  - R = 8
  - DIN_WIDTH = 16
  - ACC_WIDTH = 25
  - DOUT_WIDTH = 16
- complex_mixer.hdl
  - NCO_DATA_WIDTH_p = 12
  - INPUT_DATA_WIDTH_p = 12
  - CORDIC_STAGES_p = 16
  - PEAK_MONITOR_p = true
- iq_imbalance_fixer.hdl
  - DATA_WIDTH_p = 16
  - ACC_PREC_p = 34
  - PEAK_MONITOR_p = true
- dc_offset_filter.hdl
  - DATA_WIDTH_p = 16
  - PEAK_MONITOR_p = true
- lime_adc.hdl
  - DRIVE_CLK_p = false
  - USE_CLK_IN_p = true
  - USE_CTL_CLK_p = false
  - USE_CLK_OUT_p = false
- si5351.hdl
  - CLKIN_PRESENT = true
  - CLKIN_FREQ = 3.072e7
  - XTAL_PRESENT = false
  - XTAL_FREQ = 0
  - VC_PRESENT = false
  - OUTPUTS_PRESENT = 0,0,1,1,1,1,0,0
  - OEB_MODE = low
  - INTR_CONNECTED = false
- zipper_i2c.hdl
  - NUSERS_p = 2

# 7 Appendix B: Artifacts

## 7.1 Zedboard/FMCOMMS2/3

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_zed_cfg_1rx_0tx _fmcomms_2_3_lpc_lvds_cnt_1rx_0tx_thruasm_fmcomms_2_3_lpc_LVDS_zed.bitz
- target-linux-x13_3-arm/file_write_s.so
- target-linux-x13_3-arm/fmcomms_2_3_rx_s.so
- target-linux-x13_3-arm/ad9361_config_proxy_s.so

## 7.2 ML605 FMCOMMS2/3 in FMC HPC

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_ml605_cfg_1rx_0tx _fmcomms_2_3_hpc_lvds_cnt_1rx_0tx_thruasm_fmcomms_2_3_hpc_LVDS_ml605.bitz
- target-linux-c7-x86_64/file_write_s.so
- target-linux-c7-x86_64/fmcomms_2_3_rx_s.so
- target-linux-c7-x86_64/ad9361_config_proxy_s.so

## 7.3 ML605 FMCOMMS2/3 in FMC LPC

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_ml605_cfg_1rx_0tx _fmcomms_2_3_lpc_lvds_cnt_1rx_0tx_thruasm_fmcomms_2_3_lpc_LVDS_ml605.bitz
- target-linux-c7-x86_64/file_write_s.so
- target-linux-c7-x86_64/fmcomms_2_3_rx_s.so
- target-linux-c7-x86_64/ad9361_config_proxy_s.so

## 7.4 Matchstiq-Z1

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_matchstiq_z1_matchstiq_z1_rx_cnt_1rx_0tx_thruasm_matchstiq_z1.bitz
- target-linux-x13_3-arm/file_write_s.so
- target-linux-x13_3-arm/matchstiq_z1_rx_s.so
- target-linux-x13_3-arm/lime_rx_proxy_s.so
- target-linux-x13_3-arm/si5338_proxy_s.so
- target-linux-x13_3-arm/matchstiq_z1_avr_proxy_s.so
- target-linux-x13_3-arm/tmp100_proxy_s.so
- target-linux-x13_3-arm/matchstiq_z1_pca9535_proxy_s.so

## 7.5 Zedboard/Zipper

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_zed_base_cnt_1rx_0tx_thruasm_zipper_lpc_zed.bitz
- target-linux-x13_3-arm/file_write_s.so
- target-linux-x13_3-arm/zipper_rx_s.so
- target-linux-x13_3-arm/lime_rx_proxy_s.so
- target-linux-x13_3-arm/si5351_proxy_s.so

## 7.6 Stratix IV/Zipper

For Zipper plugged into HSMC Port A:

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_alst4_alst4_zipper_hsmc_alst4_port_a _rx_cnt_1rx_0tx_thruasm_zipper_hsmc_a_alst4.bitz

For Zipper plugged into HSMC Port B:

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_alst4_alst4_zipper_hsmc_alst4_port_b _rx_cnt_1rx_0tx_thruasm_zipper_hsmc_b_alst4.bitz
- target-linux-c7-x86_64/file_write_s.so
- target-linux-c7-x86_64/zipper_rx_s.so
- target-linux-c7-x86_64/lime_rx_proxy_s.so
- target-linux-c7-x86_64/si5351_proxy_s.so

## 7.7 ML605/Zipper

For Zipper plugged into FMC HPC:

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_ml605_ml605_zipper_fmc_lpc_rx_cnt_1rx_0tx_thruasm_zipper_lpc_ml605.bitz

For Zipper plugged into FMC LPC:

- dc_offset_iq_imbalance_mixer_cic_dec_timestamper_ml605_ml605_zipper_fmc_hpc_rx_cnt_1rx_0tx_thruasm_zipper_hpc_ml605.bitz

- target-linux-c7-x86_64/file_write_s.so
- target-linux-c7-x86_64/zipper_rx_s.so
- target-linux-c7-x86_64/lime_rx_proxy_s.so
- target-linux-c7-x86_64/si5351_proxy_s.so