

# **SigFPGA™**

## **Quick Start Guide**



797 North Grove Rd, Suite 101  
Richardson, TX 75081  
Phone: 972-671-9570  
[www.redrapids.com](http://www.redrapids.com)

Red Rapids reserves the right to alter product specifications or discontinue any product without notice. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment. This product is not designed, authorized, or warranted for use in a life-support system or other critical application.

All trademark and registered trademarks are the property of their respective owners.

Copyright © 2018, Red Rapids, Inc. All rights reserved.

## Table of Contents

1.0	Introduction.....	1
1.1	Model Number Options Decoder .....	1
1.2	Conventions .....	1
1.3	Distribution Disk (DSK-370-001-Rxx) .....	2
1.4	Revision History .....	2
2.0	Software Installation .....	3
3.0	Hardware Installation .....	4
3.1	XMC Hardware Installation.....	4
3.2	PCIe Hardware Installation.....	4
4.0	Front Panel Connectors .....	6
4.1	Model 371 .....	6
4.2	Model 372 .....	7
4.3	Model 373 .....	7
4.4	Model 374 .....	8
4.5	Model 376 .....	8
4.6	Model 377 .....	9
5.0	Hardware/Software Demonstration .....	10
5.1	Receiver Demonstration .....	10
5.2	Transmitter Demonstration .....	11
5.3	Transceiver Demonstration .....	12
6.0	Run-time Messages .....	13
6.1	Open Device .....	13
6.2	QDR II+ SRAM Test (Optional) .....	13
6.3	PCIe Performance Benchmark.....	13
6.4	Starting Channels.....	13
6.5	Start/Stop Processing.....	13
6.6	Retrieve Temperature and Power Status.....	14
6.7	Review Channel Status .....	14
6.8	Save Captured Data (RX Only) .....	14
7.0	Appendix A – Example Run-time Report.....	15

## List of Figures

Figure 4-1 Model 371 (a) XMC, (b) CCXMC, (c) PCIe Front Panel.....	6
Figure 4-2 Model 372 (a) XMC, (b) CCXMC, (c) PCIe Front Panel.....	7
Figure 4-3 Model 373 (a) XMC, (b) CCXMC, (c) PCIe Front Panel.....	7
Figure 4-4 Model 374 (a) XMC, (b) CCXMC, (c) PCIe Front Panel.....	8
Figure 4-5 Model 376 (a) XMC, (b) CCXMC, (c) PCIe Front Panel.....	8
Figure 4-6 Model 377 (a) XMC, (b) CCXMC, (c) PCIe Front Panel.....	9
Figure 5-1 Receiver Data Flow.....	10
Figure 5-2 Transmitter Data Flow.....	11

## 1.0 Introduction

The SigFPGA™ product family provides the ideal platform to rapidly field application specific signal acquisition and generation functions minus the expense of custom hardware development. The products share a common FPGA processing architecture and code base with different interface options tailored to a variety of market needs.

The latest product documentation and software is available for download from the Red Rapids web site ([www.redrapids.com](http://www.redrapids.com)).

### 1.1 Model Number Options Decoder

The SigFPGA™ product family offers several analog interface options coupled to a common digital signal processing architecture. Each product is assigned a six-digit model number (Model XXX-YYY) that uniquely identifies the specific hardware features of that unit. The first three digits are the primary designator used to convey information about the structure of the analog front-end. The last three digits define the build options selected by a customer to tailor the product to a specific application. Consult the SigFPGA™ Model Number Options Decoder manual (REF-007-000-Rxx) for a current list.

### 1.2 Conventions

This manual uses the following conventions:

- Hexadecimal numbers are prefixed by “0x” (e.g. 0x00058C).
- **Blue** font is used for names of directories, files and OS commands.
- **Green** font is used to designate source code.



Text in this format highlights useful or important information.



Text shown in this format is a warning. It describes a situation that could potentially damage your equipment. Please read each warning carefully.

The following are some of the acronyms used in this manual.

- **ADC** Analog to Digital Converter
- **API** Application Program Interface
- **CCXMC** Conduction Cooled Express Mezzanine Card
- **CLK** External Clock
- **DAC** Digital to Analog Converter
- **DEV#** Device Number #
- **GPIO** General Purpose I/O
- **LED** Light Emitting Diode
- **PCIe** Peripheral Component Interconnect Express
- **REF** External 10 MHz Clock Reference
- **RX** Receiver
- **TOD** Time of Day
- **TX** Transmitter
- **TOD** Time of Day
- **TRIG** External Trigger

- **XMC** Express Mezzanine Card

### 1.3 Distribution Disk (DSK-370-001-Rxx)

The demonstration software and SigFPGA™ API are distributed on Red Rapids disk archive number DSK-370-001-Rxx. The directory structure of the archive is outlined below:

[\](#)

The root directory contains three files that are used to compile the demonstration source code. The [VSbuild-32.bat](#) and [VSbuild-64.bat](#) batch files can be executed from a Visual Studio command prompt to build the 32-bit or 64-bit Windows executable. The [makefile](#) will compile either the 32-bit or 64-bit Linux to match the installation. The [sigfpga.c](#) main function is also located in the root directory.

[\include](#)

The [include](#) subdirectory contains the header file used to access the SigFPGA™ API.

[\linux\\_x86-32](#)

The [linux\\_x86-32](#) subdirectory is the repository for object and executable files generated when the [makefile](#) is executed from a 32-bit Linux installation. It is also the destination for output files created when the demonstration application is run.

[\linux\\_x86-64](#)

The [linux\\_x86-64](#) subdirectory is the repository for object and executable files generated when the [makefile](#) is executed from a 64-bit Linux installation. It is also the destination for output files created when the demonstration application is run.

[\src](#)

The [src](#) subdirectory contains source code to all SigFPGA™ API functions.

[\win\\_x86-32](#)

The [win\\_x86-32](#) subdirectory is the repository for object and executable files generated when the [VSbuild-32.bat](#) batch file is executed. It is also the destination for output files created when the demonstration application is run. This subdirectory includes a [Set\\_Path.bat](#) batch file that can be used to add the driver directory (DSK-806-902-Rxx) to the Windows search path.

[\win\\_x86-64](#)

The [win\\_x86-64](#) subdirectory is the repository for object and executable files generated when the [VSbuild-64.bat](#) batch file is executed. It is also the destination for output files created when the demonstration application is run. This subdirectory includes a [Set\\_Path.bat](#) batch file that can be used to add the driver directory (DSK-806-902-Rxx) to the Windows search path.

### 1.4 Revision History

Version	Date	Description
R00	2/1/2018	Initial release.

## 2.0 Software Installation

The SigFPGA™ software suite consists of three disk archives:

**DSK-806-902-Rxx:** Adapter Device Driver and API

**DSK-370-001-Rxx:** Demonstration Software and SigFPGA™ API

**DSK-806-066-Rxx:** Flash Programmer

The software is available for download from the Red Rapids website ([www.redrapids.com](http://www.redrapids.com)). Each archive is accompanied by a reference manual that describes the function and operation of the software distributed on that disk.

The distribution number of each archive is also the name of the root directory for that disk. All three archives must be extracted into the same working directory so that they share a common folder on the host. This is necessary for path names between the subdirectories to cross reference correctly.

The Adapter Device Driver needs to be installed on the host computer before any application will execute.

The demonstration software consists of the SigFPGA™ API and an application example that demonstrates all API functions. There is no special installation procedure required.

The Flash Programmer is a utility used to load firmware updates as they become available. All firmware revisions are available from the Red Rapids website ([www.redrapids.com](http://www.redrapids.com)).

---

## 3.0 Hardware Installation

Every SigFPGA™ product is available in both XMC and PCI Express (PCIe) half-length form factors. The hardware ships with a x8 physical and x8 electrical PCIe connector. It is possible to load alternate firmware that reduces the number of electrical connections if desired.

### 3.1 XMC Hardware Installation

The following instructions provide a general guide for mounting a XMC module to a baseboard or carrier host socket. Consult your baseboard documentation for product specific guidance.



This is a static sensitive electronic device; please follow standard ESD guidelines when installing the device.

**Load Drivers:** Load the appropriate software drivers prior to installing the hardware.

**Power Down:** Power to the host must be off during hardware installation. Permanent damage may result if the card is plugged into a hot socket.

**Identify Socket:** Find an open XMC socket on the host. The socket may be keyed for legacy PMC operation. The key consists of a metal post located at the centerline of the socket between the electrical connectors. The position of the post will indicate if the legacy PCI bus is designed for 3.3 Volt or 5 Volt signaling. The SigFPGA™ XMC product does not support legacy PCI operation, but the card is designed to accommodate a 3.3 Volt key for convenience.

**Insert Card:** Angle the XMC bezel into the host front panel cut-out. The EMI gasket around the bezel may offer resistance, be careful not to dislodge the gasket from the groove. Gently press the XMC down onto the host until the connectors mate.

**Secure Card:** Secure the XMC hardware with four mounting screws (provided). Be sure that the card does not bow as the screws are fastened.

**Check Obstructions:** Verify that the card is securely mounted in the XMC socket and not in contact with other components on the host.

**Boot Computer:** The host will detect the presence of new hardware the first time power is applied. Consult the driver installation instructions if prompted for additional information.



The expansion card can pick up electrical interference from other devices or directly through the host power supply. Try moving the card away from other devices or try a different host platform if you are experiencing interference.

### 3.2 PCIe Hardware Installation

The following instructions provide a general guide for PCIe expansion card installation. Consult your host documentation for product specific guidance.



This is a static sensitive electronic device; please follow standard ESD guidelines when installing the device.

**Load Drivers:** Load the appropriate software drivers prior to installing the hardware.

**Power Down:** Power to the host must be off during hardware installation. Permanent damage may result if the card is plugged into a hot socket.



**Identify Slot:** Find an open PCIe slot on the host backplane and verify that the socket is compatible with the card you are installing. The PCI Express specification defines each slot by a physical size and electrical size. It is not uncommon to find a socket electrically wired to fewer lanes than the physical size allows. The specific configuration is usually silkscreened on the motherboard.

The interoperability of PCIe cards and slots is summarized in Table 3-1. The SigFPGA™ card ships with a x8 physical connector.

**Table 3-1** PCIe Card Interoperability

Slot	x1	x4	x8	x16
Card				
x1	Required	Required	Required	Required
x4	No	Required	Allowed	Allowed
x8	No	No	Required	Allowed
x16	No	No	No	Required


**Remove Metal Insert:** The expansion slot opening will typically be covered by a metal insert held in with a single screw. The metal insert may have to be punched out of some cases. Unscrew or punch out the appropriate metal insert. It may help to align the expansion card over the slot to determine which insert to remove.

**Insert Card:** Align the expansion card edge connector with the PCIe slot making sure the bottom edge of the metal faceplate clears the edge of the backplane motherboard. Apply firm pressure to seat the card in the slot. You may need to rock the card slightly from front to back to get the unit seated properly. *Do not force the card or significantly flex the motherboard.* The expansion board should not require much force to insert provided everything is lined up correctly.

**Secure Card:** The top side of the faceplate should be flush or close to flush with the card retention bar. If the plate is not close to the bar, verify that the board is not canted in the slot. Secure the expansion card to the chassis by inserting a screw into the top of the metal faceplate.

**Check Obstructions:** Verify that the card is securely mounted in the PCIe slot and not in contact with other items inside the chassis.

**Boot Computer:** The host will detect the presence of new hardware the first time power is applied. Consult the driver installation instructions if prompted for additional information.



The expansion card can pick up electrical interference from other devices or directly through the host power supply. Try moving the card away from other devices or try a different host platform if you are experiencing interference.

## 4.0 Front Panel Connectors

The face of each SigFPGA™ product contains an array of coaxial and digital connectors to connect signals, clocks and external triggers. If a bezel or face plate is present, each connector will be marked with one of the following designators:

**RX#:** Analog receiver channel number # input.

**TX#:** Analog transmitter channel number # output.

**GPIO:** General purpose I/O connector for external triggers.

**REF/CLK:** External 10 MHz reference or sample clock input.

**TRIG:** Coaxial trigger input or 1 PPS timing input.

There are two light emitting diodes (LEDs) located on either side of the GPIO connector. They can be illuminated by writing to BAR2 address x0020 from software. Writing a one to bit 0 will illuminate the yellow LED and writing a one to bit 1 will illuminate the green LED.

The connector placement and LED location for each product is shown below. The three available form factors are illustrated for each product.

### 4.1 Model 371

The Model 371 front panel includes four coaxial connectors and one GPIO connector as shown in Figure 4-1. The coaxial receiver channel inputs are numbered RX1 and RX2. The remaining two coaxial connectors are the REF/CLK input and TRIG input.

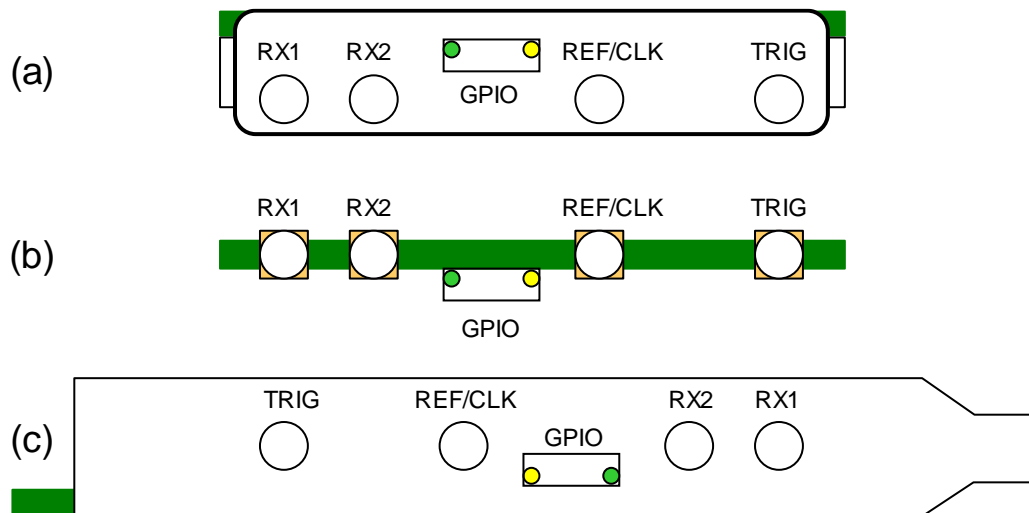


Figure 4-1 Model 371 (a) XMC, (b) CCXMC, (c) PCIe Front Panel

### 4.2 Model 372

The Model 372 front panel includes five coaxial connectors and one GPIO connector as shown in Figure 4-2. The coaxial receiver channel inputs are numbered RX1 and RX2, while the transmitter outputs are numbered TX3 and TX4. The fifth coaxial connector can either be used as a REF/CLK input or a TRIG input.

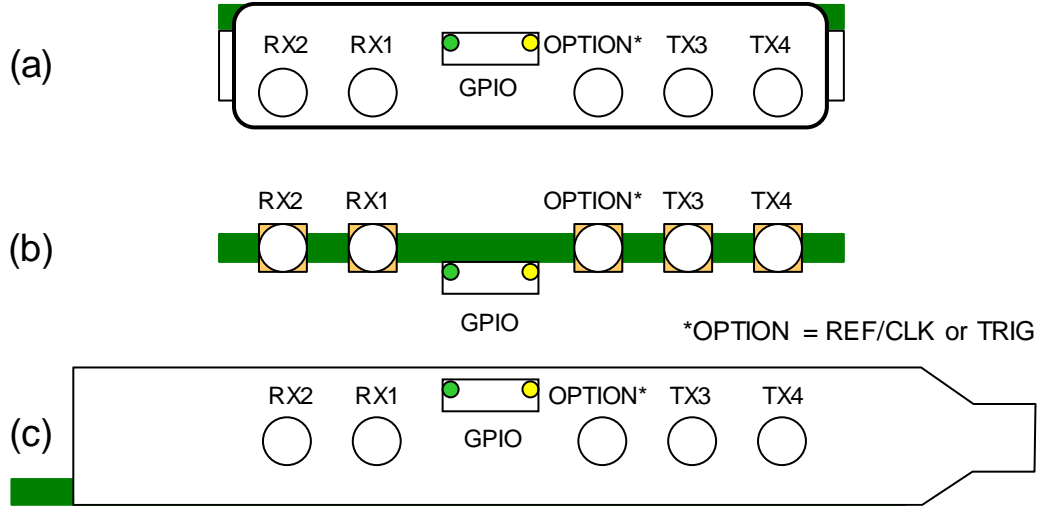


Figure 4-2 Model 372 (a) XMC, (b) CCXMC, (c) PCIe Front Panel

### 4.3 Model 373

The Model 373 front panel includes four coaxial connectors and one GPIO connector as shown in Figure 4-3. The coaxial receiver channel inputs are numbered RX1 and RX2. The remaining two coaxial connectors are the REF/CLK input and TRIG input.

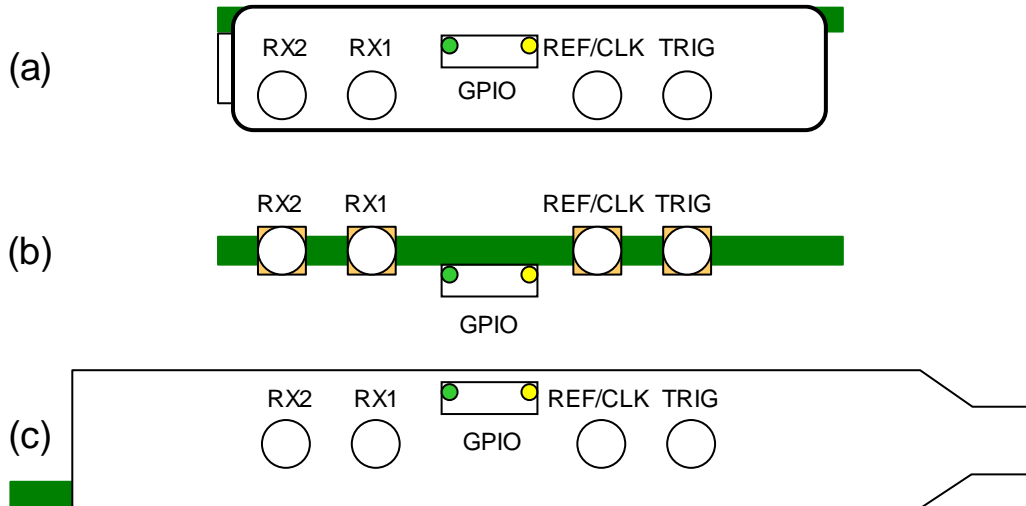


Figure 4-3 Model 373 (a) XMC, (b) CCXMC, (c) PCIe Front Panel

#### 4.4 Model 374

The Model 374 front panel includes four coaxial connectors and one GPIO connector as shown in Figure 4-4. The coaxial transmitter channel outputs are numbered TX1 and TX2. The remaining two coaxial connectors are the REF/CLK input and TRIG input.

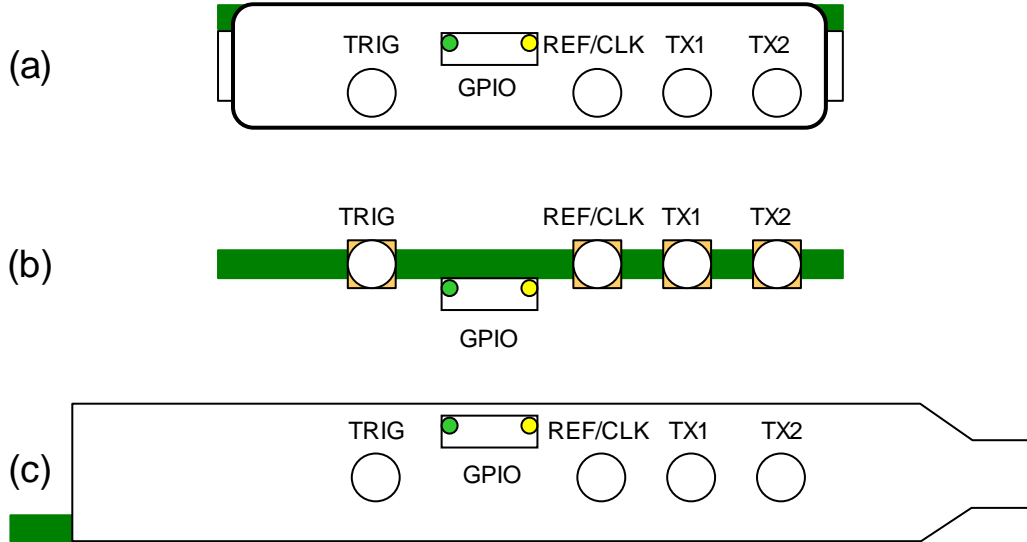


Figure 4-4 Model 374 (a) XMC, (b) CCXMC, (c) PCIe Front Panel

#### 4.5 Model 376

The Model 376 front panel includes four coaxial connectors and one GPIO connector as shown in Figure 4-5. The coaxial receiver channel inputs are numbered RX1 and RX2. The remaining two coaxial connectors are the REF/CLK input and TRIG input.

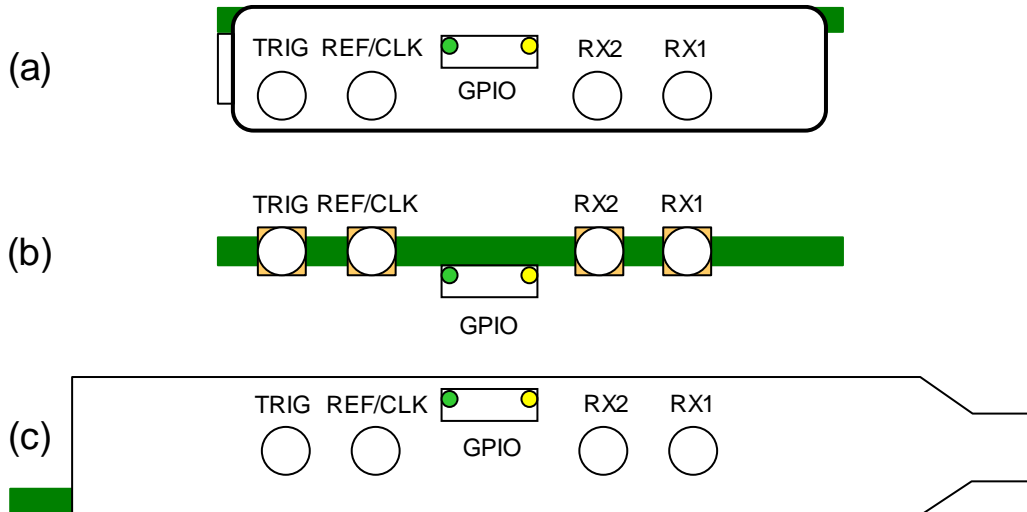


Figure 4-5 Model 376 (a) XMC, (b) CCXMC, (c) PCIe Front Panel

### 4.6 Model 377

The Model 377 front panel includes five coaxial connectors and one GPIO connector as shown in Figure 4-6. The coaxial receiver channel inputs are consecutively numbered RX1 through RX4. The fifth coaxial connector can either be used as a REF/CLK input or a TRIG input.

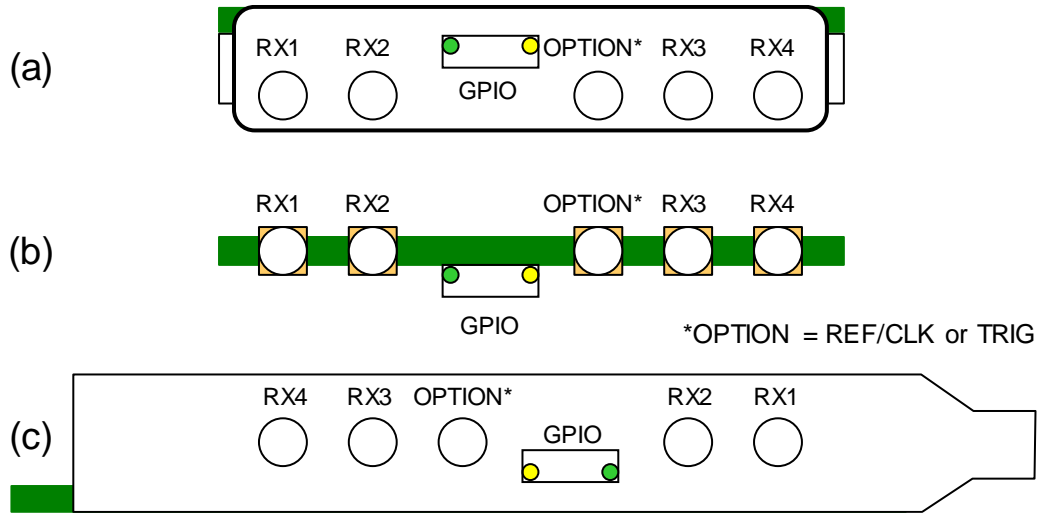


Figure 4-6 Model 377 (a) XMC, (b) CCXMC, (c) PCIe Front Panel

## 5.0 Hardware/Software Demonstration

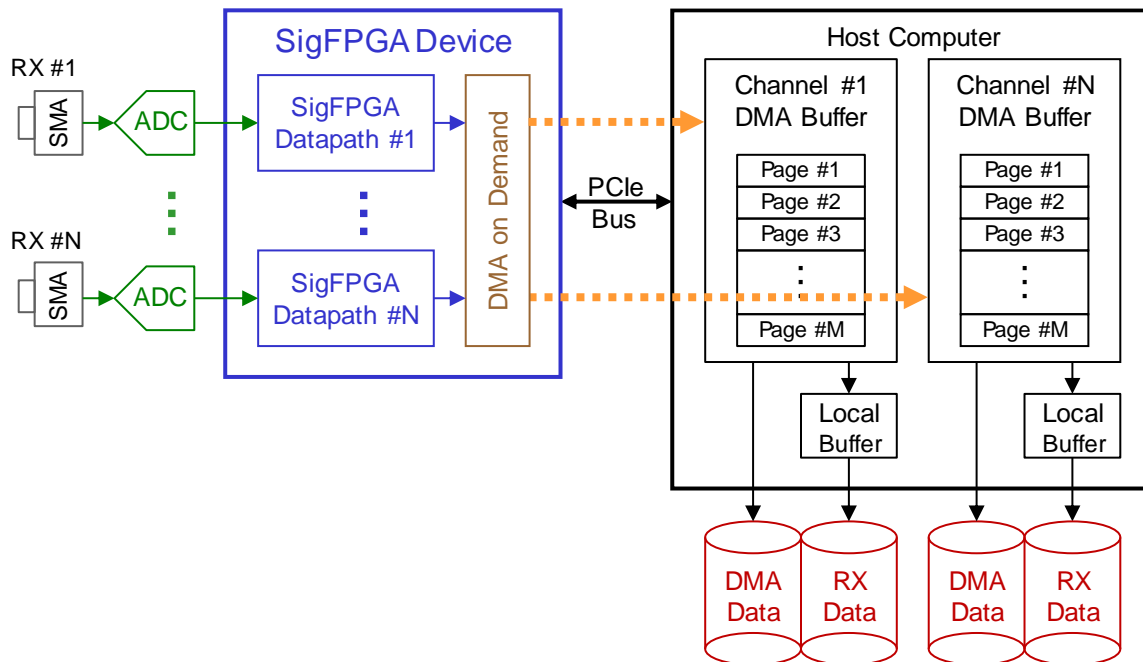
Each SigFPGA™ product is accompanied by a software application that demonstrates hardware functionality using calls to the supplied API. Products equipped with receiver channels will digitize an analog signal, transfer samples to host memory using DMA transactions, and store a snapshot of the collection to a text file. Products equipped with transmitter channels will retrieve digital samples from host memory via DMA transactions and produce an analog reproduction of the digital signal. The transmitter samples are precomputed to produce a sinusoid of frequency  $F_s/8$ .

The demonstration software is provided on software archive DSK-370-001-Rxx. The application can be run from a command prompt by executing the following instructions from the root directory.

32-bit Linux	64-bit Linux	32-bit Windows	64-bit Windows
> make > cd linux_x86-32 > ./sigfpga	> make > cd linux_x86-64 > ./sigfpga	> cd win_x86-32 > Set_Path > sigfpga	> cd win_x86-64 > Set_Path > sigfpga

### 5.1 Receiver Demonstration

Figure 5-1 illustrates the hardware path exercised by the receiver demonstration software. The application enables all available channels and issues a software trigger to initiate signal acquisition.



**Figure 5-1 Receiver Data Flow**

The SigFPGA™ datapath processes the number of samples requested by the channel configuration. All channels are configured identically for demonstration purposes.

Each channel continuously writes digital data samples to a dedicated DMA buffer in host memory. The DMA transactions are mastered by the SigFPGA™ device, no software intervention is required. Consult the DMA on Demand Operating Guide (REF-004-000-Rxx) for further details about DMA transactions. The application software continuously transfers the contents of each DMA buffer to another local memory buffer to demonstrate how a customer application would manage the incoming data.

The demonstration will create two output files per channel. The first set of files are named `dma#buf.txt`, where # is the channel number. This file contains the first page of sample data in the DMA buffer. The number of pages that are written to the file can be changed in the call to the `DMA Save()` API function, but must not be any greater than the total number of pages in the DMA buffer. The second set of files are named `rx#data.txt`, where # is the channel number. This file contains the first page of sample data stored in the local buffer. The number of pages written can be increased, but must not be any greater than the number of pages between DMA markers. The data in the `rx#data.txt` and `dma#buf.txt` files will only match if processing terminated while writing to a DMA page number between the first and second DMA marker.

No external equipment is required to run the demonstration code, but the receiver channels will only collect noise samples if there is no input signal present. Each SigFPGA™ product is assigned an analog Front End Reference Manual (REF-000-YYY-Rxx) that contains information about the expected input signal power.

## 5.2 Transmitter Demonstration

Figure 5-2 illustrates the hardware path exercised by the transmitter demonstration software. The application enables all available channels and issues a software trigger to initiate signal generation.

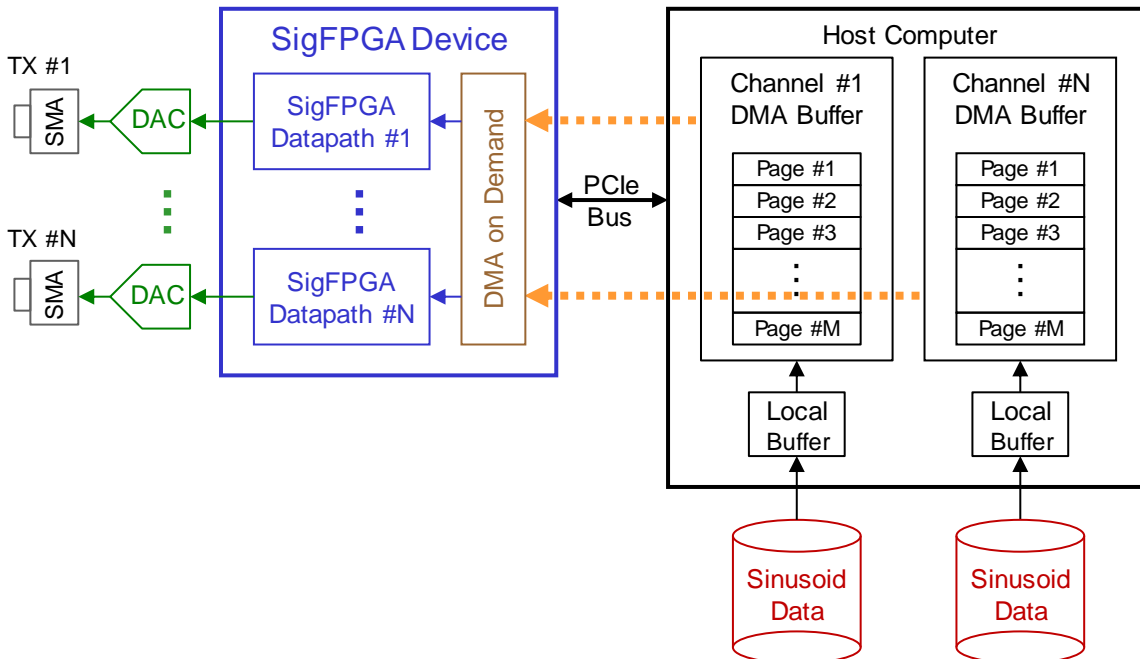


Figure 5-2 Transmitter Data Flow

The SigFPGA™ datapath processes the number of samples requested by the channel configuration. All channels are configured identically for demonstration purposes.

The demonstration software loads digital sinusoid samples into a local memory buffer before the channels are enabled. Once processing begins, the application continuously transfers the contents of the local buffer to a DMA buffer to demonstrate how a customer application would supply outgoing data. The SigFPGA™ device continuously reads samples from the DMA buffer in host memory and supplies them to the datapath. The DMA transactions are mastered by the SigFPGA™ device, no software intervention is required. Consult the DMA on Demand Operating Guide (REF-004-000-Rxx) for further details about DMA transactions.

It is important to note that the waveform stored in the local buffer does not get updated once channel processing begins. The same block of samples will be repeated through the datapath if the requested number of samples exceeds the size of the local buffer. This will result in phase discontinuities if the final waveform sample does not represent the end of a period.

No external equipment is required to run the demonstration code, but a spectrum analyzer can be used to monitor the output waveform of each channel. Each SigFPGA™ product is assigned an analog Front End Reference Manual (REF-000-yyy-Rxx) that contains information about the expected output signal power.

### **5.3 Transceiver Demonstration**

SigFPGA™ products that are equipped with both receiver and transmitter channels will run both demonstrations concurrently. A single software trigger will initiate processing on all channels simultaneously, regardless of direction.



## 6.0 Run-time Messages

The SigFPGA™ application reports status, warning, and error messages to the screen as it executes. Appendix 7.0 contains an example report that was generated by a system equipped with both receiver and transmitter devices.

### 6.1 Open Device

Three messages are generated when a hardware device is successfully opened by the application.

- (1) Firmware revision reported by the hardware.
- (2) Model number reported by the hardware.
- (3) Clock status.

The first two messages simply convey information to identify the hardware. The third message reports the health of the clock network and the specific clock source detected by the hardware. This message will change depending on whether there is an external reference or external sample clock connected through the front panel. An error will be reported if any type of clock problem is detected.

### 6.2 QDR II+ SRAM Test (Optional)

If the optional QDR II+ SRAM is present, a test is performed to verify that it is functioning correctly. A unique value is written to each memory address and read back to compare against the expected value. The number of errors detected for each SRAM is reported.

### 6.3 PCIe Performance Benchmark

It is important to know whether the host computer can supply the PCIe bus bandwidth required by a SigFPGA™ device. A performance test is run by the application to measure actual performance with all available channels continuously making DMA requests. The number reported by the benchmark must be above the expected throughput of the device to avoid losing data.

The performance benchmark is only provided for convenience, it can be eliminated from the code once the information is collected.

### 6.4 Starting Channels

A message is generated as every channel on the device is started in sequence. Channel processing does not begin immediately, the channels are simply armed and waiting for the requested trigger.

### 6.5 Start/Stop Processing

A software trigger is issued to start processing samples on all channels simultaneously. The application waits silently for each channel to complete the requested number of samples and stop. Terminating the process while the channels are active (CTRL-C) may crash the computer since the hardware is still accessing protected DMA buffer space in host memory.

The demonstration software copies fresh data out of the receiver DMA buffers into local memory while the channel is active. Similarly, fresh data is copied into the transmitter DMA buffers. The DMA buffers can be monitored by either polling the status bits or waiting for an interrupt. The demonstration software defaults to polling.

## 6.6 Retrieve Temperature and Power Status

There are multiple temperature sensors and power monitors available on the SigFPGA™ device. The application queries all available sensors and reports the results when processing is complete. Any temperature above 90 degrees C indicates inadequate cooling for the hardware.

## 6.7 Review Channel Status

There are several error conditions monitored by the device throughout operation. These are reviewed and reported when channel processing completes.

- (1) Read or write to an illegal BAR address.
- (2) Loss of stable clocks.
- (3) ADC over-range.
- (4) FIFO overflow (receiver) or underflow (transmitter).
- (5) Loss of DMA data.

The first message simply indicates that software issued a PCIe read or write command to a register address that does not exist in the hardware. A clock error would only occur if there were some type of hardware event that causes a clock to lose lock. The ADC over-range occurs when the input signal exceeds the maximum amplitude supported by the device, but it is worth noting that not all ADC chips offer this status. A FIFO error will usually occur when the PCIe bus is not fast enough to meet the throughput demands of all active channels competing for resources. A loss of DMA data indicates that the software servicing the DMA buffers in host memory is not fast enough to keep up with the refresh rate demanded by the hardware. This might be resolved by increasing the size of the DMA buffers or changing the number of pages between markers. However, it is always possible that the host processor is simply not fast enough to keep up with the transfer of data between the device and host.

## 6.8 Save Captured Data (RX Only)

Every receiver channel will produce two text files as discussed in Section 5.1. Both files will contain signed integer values. A real data sample occupies a single line in the file while a complex sample occupies two lines. The real (in-phase) value always precedes the imaginary (quadrature) value of a complex sample. The maximum allowed positive or negative value is dictated by the data item size selected in the channel configuration.

## 7.0 Appendix A – Example Run-time Report

The following output messages were produced by the SigFPGA™ application running on a host computer with a Model 372 transceiver.

\*\*\*\*\* SigFPGA \*\*\*\*\*

Initializing structures...

Opening device #0 on the host...

Firmware revision (MMDDYYYY): 01102018

Initializing Red Rapids Model 372...

DAC interpolation set to x4.

External sample clock active, converter clocks locked.

Measuring the PCI performance...

Measured RX PCI performance: 3.195 Gbytes/sec

Measured TX PCI performance: 1.225 Gbytes/sec

Measured RX PCI latency: 7.400 usec

Measured TX PCI latency: 10.176 usec

Starting channels...

Channel #1 started.

Channel #2 started.

Channel #3 started.

Channel #4 started.

Waiting for all channels to stop processing (interrupts disabled)...

WARNING: CTRL-C FROM THE KEYBOARD MAY CRASH THE COMPUTER!

Retrieving temperature status...

INFO: FPGA die temperature is 45.09 degrees Celsius.

INFO: ADC proximity temperature is 33.00 degrees Celsius.

INFO: QDR proximity temperature is 28.00 degrees Celsius.

INFO: Shielded proximity temperature is 31.00 degrees Celsius.

INFO: Card center proximity temperature is 32.00 degrees Celsius.

Retrieving power status...

INFO: Total power dissipation is 12.02 Watts.

INFO: Supply #1 summary: 3.36 Volts, 0.30 Amps, 1.01 Watts

INFO: Supply #2 summary: 12.19 Volts, 0.19 Amps, 2.34 Watts

INFO: Supply #3 summary: 12.18 Volts, 0.71 Amps, 8.68 Watts

Reviewing channel status...

Saving captured data on all receiver channels...

Writing channel #1 data buffer contents to file rx1data.txt.

Writing channel #1 DMA buffer contents to file dma1buf.txt.

Writing channel #2 data buffer contents to file rx2data.txt.

Writing channel #2 DMA buffer contents to file dma2buf.txt.

...all operations completed.