Altera Wireless RF Framework Data Sheet

DS-01023-1.0 © January 2011

This data sheet introduces and describes the function, key components, and the system requirements of the Altera® wireless RF framework.

The Altera RF framework supports the development of wireless RF card applications and similar systems. With the Altera wireless RF framework, you can integrate third-party mixed signal RF development boards with an Altera FPGA development board. The MathWorks MATLAB tool serves as a host development and analysis environment for these boards.

The wireless RF framework supports digital predistortion (DPD) solution developers. DPD solution evaluation requires real-time operation with the analog hardware to accurately evaluate a solution's capabilities in adapting to changes in the power amplifier's (PA) characteristics. The wireless RF framework provides an expedient route to such real-time DPD solution evaluation.

The reference design uses a Stratix® IV GX development board and Analog Devices Incorporated (ADI) mixed signal digital predistortion (MSDPD) board which supports transmit signal bandwidths of greater than 40 MHz.

For more information about Stratix IV GX development boards, refer to the Stratix IV GX FPGA Development Kit page.

This data sheet contains the following sections:

For more information about ADI MSDPD board, refer to the Analog Devices Incorporated web page.

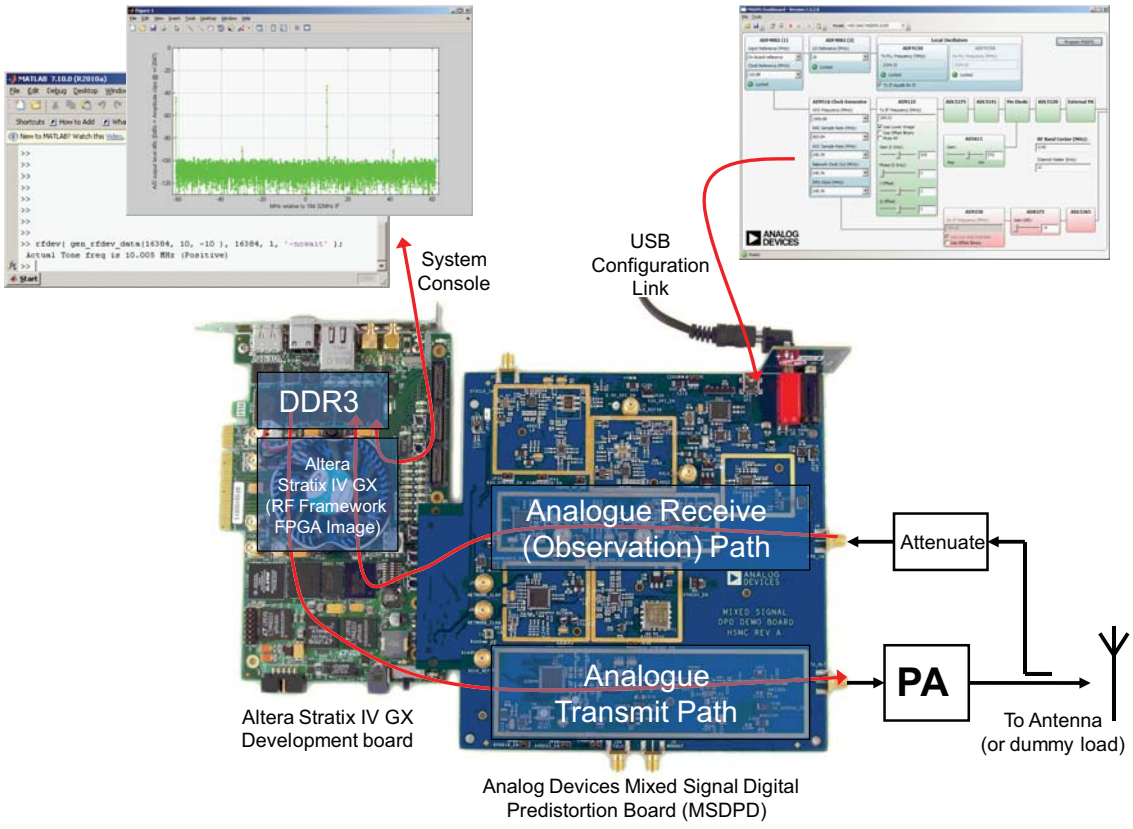
101 Innovation Drive
San Jose, CA 95134
www.altera.com

Functional Description

The MATLAB and the RF framework setup enables you to download waveforms directly from MATLAB to the Stratix IV GX development board. The waveforms are then played out through the transmit path of the ADI MSDPD board. Concurrently, large contiguous blocks of data are captured from the receive path of the ADI MSDPD board. Waveforms are read back to the host MATLAB development and analysis environment. The hardware setup is representative of a basestation downlink transmit path with a feedback DPD observation path. Transmit and capture functions operate in real time at the sample rate required by the analog-to-digital converter (ADC) and digital-to-analog converter (DAC). On the board, DDR3 memory functions as a high-density storage for transmit and receive waveforms.

Figure 1 shows the wireless RF framework setup with a Stratix IV GX development board connected to an ADI MSDPD board and controlled by the MATLAB tool.

Figure 1. Altera Wireless RF Framework



Wireless RF Framework Key Components

The following sections describe the key components of the wireless RF framework:

FPGA Image

The SOPC Builder system design tool and system console are used to create the FPGA image.

The SOPC Builder system design tool integrates Altera intellectual property (IP) components with user-created IP blocks. Additionally, SOPC Builder enables rapid development with the Nios® II processor and supporting debug tools.

System console creates a data transfer pipe between a host PC and the SOPC Builder subsystem in the FPGA. This data transfer pipe enables you to control IP components and download or upload waveforms from the MATLAB environment.

The FPGA image contains the following components:

- An instance of the SOPC Builder subsystem
- DAC and ADC controller interface logic specific to the ADI MSDPD board
- Phase-locked loops (PLLs) to drive the SOPC Builder system and the interface logic
- Board-specific functions such as buttons and LEDs
- Pin naming to aid pin-out for the Stratix IV PCIe development kit

This approach allows the SOPC Builder system to be agnostic to the Altera development board and the third-party RF hardware.

The SOPC Builder subsystem contains the following components:

- An instance of the Altera DDR3 controller, to enable efficient memory accesses
- The 'play out' chain from DDR to analog hardware, which includes a read DMA engine and data formatting blocks
- The capture path from analog hardware to DDR memory data, which includes formatting blocks and a write DMA engine
- The system console connection

You can integrate any signal processing or Nios II processor cores, or both, in the SOPC Builder subsystem to create a complete DPD solution for real-time prototyping. Projects containing examples of these are provided.

MATLAB Development and Analysis Environment

The MATLAB tool integration enables you to evaluate the analog hardware with little or no knowledge of the FPGA hardware or the system console host connection. In MATLAB, several application programming interface (API) functions represent the system console connection.

Example 1 shows the MATLAB code function calls, which read and write to IP components in the SOPC Builder subsystem address space (which connect to an Avalon-MM bus).

For more information about the Avalon Interface Protocol, refer to the *Avalon Interface Specifications*.

Example 1. MATLAB Codes

```
>> MasterWrite32(memory, start_address, data, ['-verbose'])
>> data= MasterRead32(memory, start_address, length, ['-verbose'])
```

Where:

- `memory` is the unique identifier to the system console connection
- `start_address` is the start location for the accesses
- `data` is data that are be transferred (an array of 32-bit signed integers)
- `length` specifies the number of bytes of data to read
- `-verbose` is optional to enhance debug capability

MATLAB API enables you to download waveforms, control your IP components, read back waveforms, and monitor your IP component status information. This setup allows both interactive use of the RF framework and also seamless integration of framework control directly into your own algorithms and test routines.

An example MATLAB application code, using the MATLAB API, is provided which can be used as:

- A simple function call to access a microwave power amplifier (PA) as a 'real world' model for use in rapid DPD algorithm development.
- A demonstration for the system and the analog performance of the third-party mixed signal board.
- An example starting point to create your own system.
- A basis for making additional analog and system tests and measurements.

The example MATLAB application code functions contain debug and hardware test options and perform the following operations:

- Downloads waveforms to the on-board memory
- Configures and controls the waveform transmit and capture
- Returns captured data to the host for analysis and processing
- Displays returned data (for example, fast Fourier transforms [FFTs])

Getting Started

The wireless RF framework installer contains the following components:

- Matlab API—interacts with the hardware
- Matlab application—controls data transmit and capture
- FPGA image in SRAM Object File (**.sof**) with associated register transfer level (RTL) design files
- Demo projects for a Stratix IV PCIe board
- Project examples and demonstration utilities
- Documentation

For more information about the latest project examples or access to the Altera wireless RF framework, contact Altera Support or visit Wireless page.

System Requirements

This section describes the hardware and software requirements to run the wireless RF framework reference design.

Hardware Requirement

The wireless RF framework reference design requires the following hardware:

- Stratix IV GX development board with PSU and cables
- Analog Devices MSDPD board
- Optional daughtercard to support high-speed USB 2.0 link (for more information, contact Altera Support)

Software Requirement

The wireless RF framework reference design requires the following software:

- The Mathworks MATLAB tool (R2010a or R2010b)
- The Altera Quartus® II software (version 10.1 or later)

Document Revision History

Table 1 shows the revision history for this document.

Table 1. Document Revision History

Date	Version	Changes
January 2011	1.0	Initial release.