Multi-Channel Transceiver FPGA Core
Supports implementation of a multi-channel radio on Spectrum SDR platforms

Benefits
- Speeds implementation of a full functional multi-channel radio on Spectrum’s COTS flexComm™ software defined radio (SDR) platforms
- Frees up resources for other application code through efficient processor utilization.
- Easily configures and scales to support various combinations of narrowband and wideband applications.
- Precise timing enables the implementation of Burst Waveforms including Frequency Hop and Time Division Multiplex.
- Extends across any Spectrum platform (3u or 6u cPCI, PCI/PCI-X, VME) that supports Spectrum’s XMC modules
- Spectrum’s Application Engineering Services team will customize to specification

Applications
- Military Communications/Tactical MILCOM
- Satellite Communications
- Electronic Countermeasures/Electronic Support Measures
- Signals Intelligence (Wideband Spectral Analysis, Multi-Channel Direction Finding: TOA/TDOA Calculations available)
- Phased Array Surveillance RADAR

Description
The Multi-Channel Transceiver (MC-XCVR) core for field programmable gate arrays (FPGAs) provides a mechanism to quickly implement a fully functional multi-channel radio on Spectrum’s flexComm SDR platforms. The core has been developed to enable communication channel dimensions in time, frequency and code for both continuous and burst signals. The core is targeted to any of Spectrum’s XMC IF transceiver modules supporting Xilinx Virtex-4 technology, such as the XMC-3321, and can be mounted to any Spectrum carrier board capable of supporting an XMC module.

The FPGA core itself implements high-speed digital up and down conversion, data formatting and time stamping, and high-speed transport between the FPGA and a host processor (see Figure 1). An API is provided allowing the user to easily setup the core components and transport high-speed data to and from the FPGA. System examples with full software source are also provided to demonstrate the MC-XCVR core capabilities and use of the API on a number of different platforms.

Features
- Highly efficient digital up conversion and/or digital down conversion
- Application Programming Interface (API) that simplifies the setup and operation of the core, including initialization and control functions, digital down converter (DDC) and digital up converter (DUC) functions
- High-speed dynamic programmability of filter taps.
- Integrated with key radio physical interfaces including timing, interrupt, and GPIO
- System examples to demonstrate the use of the API, SPS Library, and the operation and performance of the core while operating on a flexComm platform
- Ability to set center frequency and the set number of instantiations
- Scalable from narrowband to wideband (up to 1.25 MHz)
- Integrated debug and customization interfaces
- Customized core is delivered as a package that includes source code and training by Spectrum Applications Engineering

Figure 1. Block Diagram
Core Design

The main function of the MC-XCVR core is to digitally convert signals between digital IF and baseband quadrature (Zero IF) frequencies. Spectrum’s DDC and DUC cores perform the digital down an up conversion. The MC-XCVR core can contain a selectable number of DDC and DUC channels allowing the user to independently tune multiple signals within each IF channel.

The baseband data interface of the DDC cores is formatted, time stamped, and packetized before it is transferred from the FPGA over Spectrum’s high-speed communication fabric, Solano® or a RapidIO data link to a host processor or digital signal processor (DSP). Similarly, baseband data transmitted from the processor to the FPGA is de-packetized, formatted, and up-converted by the DUC. An arbiter is used to control the flow of multiple baseband channels over a single logical channel. Figure 2 shows an example implementation of the MC-XCVR core on Spectrum XMC-3321 dual transceiver module. Provisions have been made to allow the real time modification of filter taps in support of adaptive algorithms.

Software Design

The software package included with the MC-XCVR contains an API and system examples. The API provides the user with high-level software routines to initialize the platform, setup and configure the MC-XCVR core and its components, construct and deconstruct data packets, upload filter taps and transfer digital baseband data to and from the FPGA and the host processor. The system examples provide application code that demonstrates the use of the API and the basic operation and performance of the MC-XCVR core. The bandwidth of the DDC and DUC and the number of filter taps is programmable and customizable.

[Receive Path]

The number of receive paths in the MC-XCVR core is customizable. A receive data path originates from the digital IF input which is typically direct from an analog-to-digital converter (ADC). The IF is processed by the DDC bank where the individual channels are tuned, filtered, and decimated. The number of receive paths and the number of channels for each path may be set through a core parameter.

From the DDC bank, the data passes through the packetizer bank where the data is formatted, time tagged, packetized, and buffered. There is one packetizer for each DDC or channel. The high-speed data arbiter multiplexes the packets generated by the packetizer bank and transmits the packets to the host processor via the Solano or RapidIO FPGA wrapper interface.

[Transmit Path]

The number of transmit paths in the MC-XCVR core is customizable. The transmit path originates from the host processor or DSP on the carrier board. The processor packetizes the digital baseband data and transmits it to the FPGA through the Solano or Rapid I/O high-speed data interface. The high-speed data arbiter routes the incoming packet to the appropriate transmit path based on the path number contained in the packet header. Note that for the transmit side, there can only be one channel per path. The number of paths are set through a core parameter.
The packets for each transmit path are processed by the depacketizer, which strips the header information from the packet. The resulting baseband data stream is up converted to a user tunable center frequency by the DUC. The output of the DUC is at the IF rate which is typically fed directly to a digital-to-analog converter (DAC).

[Time Generator]

The time generator core is a multi-port time keeping core with an internal sub-microsecond resolution clock. There is one output port for each packetizer, and each port has its own user configurable group-delay compensation. This value is used to compensate for the processing delay through the DDC cores so that time stamps may accurately reflect the time at which a sample arrived at the ADC and not the time that it reached the packetizer.

The internal clock runs off the system clock, which, depending on the platform is some factor of the sample clock. For the system examples, this value is always 96 MHz. The internal clock is simply a 32-bit counter that terminates and restarts when it reaches one second’s worth of clocks (in the examples, the clock will count from 0 to 95,999,999).

The time generator has two modes of operation, simple time and IRIG-B time. In simple time mode, a 24-bit seconds counter is incremented every time the internal clock reaches terminal count. In this mode, the seconds counter and the internal clock counter are the time outputs of the core. In IRIG-B mode, an external IRIG-B signal is decoded and the BCD date and time values along with the internal clock count are the outputs of the core.

In IRIG-B mode, a 1PPS signal is derived from the signal that restarts the internal clock. If the IRIG-B signal and the internal clock are synchronized, then the internal clock should terminate and restart as normal. However, if these signals are not synchronized, the internal clock may drift or jitter.

[Control Path]

Each component within the MC-XCVR Core has its own memory-mapped registers for setup and configuration. These registers are accessible over a common bus called the Spectrum Core Bus (SCB). Inside the MC-XCVR core is a bridge that connects the SCB with the host processor through the wrapper processor bus interface (see Figure 3). The bridge is memory mapped to the processor interface so that the host processor can access all of the components and associated configuration registers on the SCB.

In the Simple Time mode, the seconds counter is incremented every time the internal clock reaches terminal count. In this mode, the seconds counter and the internal clock counter are the time outputs of the core. In IRIG-B mode, an external IRIG-B signal is decoded and the BCD date and time values along with the internal clock count are the outputs of the core.

In IRIG-B mode, a 1PPS signal is derived from the signal that restarts the internal clock. If the IRIG-B signal and the internal clock are synchronized, then the internal clock should terminate and restart as normal. However, if these signals are not synchronized, the internal clock may drift or jitter.

[Core Parameters]

The MC-XCVR core offers a series of parameters used to characterize and generate the core for a particular application. The parameters are VHDL generics that are applied to the component of the MC-XCVR. These parameters include clock rate, number of transmit paths, number of receive paths, channel configuration for each receive and transmit paths, maximum decimation/interpolation of cascaded integrator comb (CIC), input/output bit width of IF data, FIR functions, and many more.

Software Design

Included with the MC-XCVR core is an API that simplifies the setup and operation of the core and a library of routines for manipulating and formatting packets based on the Spectrum Packet Specification (SPS). System examples are also provided to demonstrate the use of the API, SPS Library, and the operation and performance of the MC-XCVR core.

The API utilizes the Spectrum Core Library and quicComm™ to communicate with the MC-XCVR core. quicComm provides an abstraction layer between the hardware and the software enabling portability of the API to multiple Spectrum platforms. The Spectrum Core Library is a library of routines for accessing and manipulating the memory-mapped registers of the components on the Spectrum Core Bus.
[ Application Programming Interface (API) ]

The API contains functions that control the various components of the MC-XCVR core in the FPGA and functions to transfer high-speed data to and from a host processor or DSP. The API includes routines to implement the following functions:

- Initialize and configure the system, including loading the FPGA design
- Set the DDC and DUC filter parameters and center frequencies
- Construct, de-construct, and manipulate packets
- Set the time and time group delay compensation values
- Send and receive high speed data to and from the FPGA

Multiple processors may use the API but only one processor can initialize the system. The application designer must take care to ensure that only one processor is responsible for controlling and manipulating any one component in the MC-XCVR Core.

[ Spectrum Core Library ]

The Spectrum Core Library is a library of software routines designed to provide an abstraction layer between upper layer software components and the MC-XCVR core components. The API utilizes the Spectrum Core Library to discover and query components, allocate and de-allocate handles for the components, and provide a means to access registers and memory within the components.

Spectrum’s Application Engineering Services (AES) organization routinely develops custom IP cores for clients. For more information on custom cores, please contact Spectrum Sales.

[ Spectrum Packet Specification Library ]

The Spectrum Packet Specification (SPS) Library contains routines for simplifying the manipulation and construction of SPS packets.

[ quicComm ]

quicComm is a library supplied by Spectrum that provides the software interface to the hardware. For this core, quicComm performs the following functions:

- Initializes and resets all Spectrum hardware components
- Loads the FPGA design
- Initializes high-speed data links
- Provides high-speed data read and write functions
- Provides the register-level abstraction layer for the Spectrum Core Library

For more details, please see the quicComm datasheet at www.spectrumsignal.com.

[ Training ]

Spectrum will provide customized training with every purchase of the MC-XCVR core and Spectrum hardware. An Applications Engineer will work with you to determine the optimal DDC/DUC parameters depending on your application requirements. During the training, our Applications Engineer will show you how to configure and build the core, and then integrate it into an application.

[ Customization ]

For customers who have special or more extensive application needs, Spectrum will customize the core to your requirements. Such customizations may include, but are not limited to, adding functionality or custom IP components or developing end-to-end applications or waveforms. Spectrum’s engineers will help you develop a Statement of Work (SOW) and Requirements Specification, develop to the SOW/specifications, provide customized training, support integration activities and lifecycle support.

[ Ordering Information ]

- Training-065 1-DAY Multi-Channel Transceiver (MC-XCVR) FPGA Core, Customization and Training – BBY or ON-SITE EXTENSION
- Training-066 1-DAY Multi-Channel Transceiver (MC-XCVR) FPGA Core, Customization and Training – ON-SITE STAND ALONE

Contact Spectrum sales for special customizations or application development.

[ Future Options ]

Future options may be implemented at the discretion of Spectrum Signal Processing Inc. based on market demand.**

Future options include fractional resampler, digital pre-distortion, high-speed BPSK/QPSK modems, direct sequence spread spectrum modem, OFDM modem, Spectrum sensing technologies, AGC and OCP support.