RADIO WAVEFORM DEVELOPMENT SYSTEM PROVIDING AN INTEGRATED APPROACH TO SDR WAVEFORM DESIGN AND IMPLEMENTATION

Nirali K. Patel (Coherent Logix, Inc., Austin, TX, USA, patel@coherentlogix.com);
Kevin A. Shelby (Coherent Logix, Inc., Austin, TX, USA, shelby@coherentlogix.com);
Brian A. Dalio (Coherent Logix, Inc., Austin, TX, USA, dalio@coherentlogix.com)

ABSTRACT

This paper presents a Radio Waveform Development System (RWDS) intended to enable rapid software-based waveform design, implementation and real-time characterization. The RWDS is based on a multi-core signal processor supported in a unified development environment seamlessly spanning algorithm design to final system implementation. It allows for dynamic allocation of processing resources between multiple waveforms and scaling to enable additional processing capability as needed. The modularity provides an easy way to configure the system to accommodate a variety of waveform applications providing a path to final form factor while preserving the software stack. Other benefits include model-based design support and library driven module reuse to shorten development time. The RWDS is currently in use across multiple industry segments spanning mobile radio, military communication, first responder and satellite navigation systems. We present the use of RWDS in the context of a reference waveform implementation the details of which are given in terms of a functional decomposition of the system design, mapping to hardware modules and performance characterization.

1 Introduction

The intent with the RWDS is to promote an SDR outlook throughout the waveform development process from algorithm design and design tradeoff analysis through system implementation and performance characterization. The objective is to provide integrated access to the signal processing complex in a modular framework applicable to a broad assortment of waveform types and system configurations easing the transition to form factor prototype suitable for product deployment.

Conventional SDR development platforms [1][2][3] built around a combination of DSP, GPP and FPGA, employ separate design tools for individual system components resulting in a disjoint development flow. This requires early allocation of processing resources making repartitioning as the design progresses toward final implementation cumbersome and time consuming. The development platforms are often customized for a particular application limiting their ability to scale to support a diverse application set.

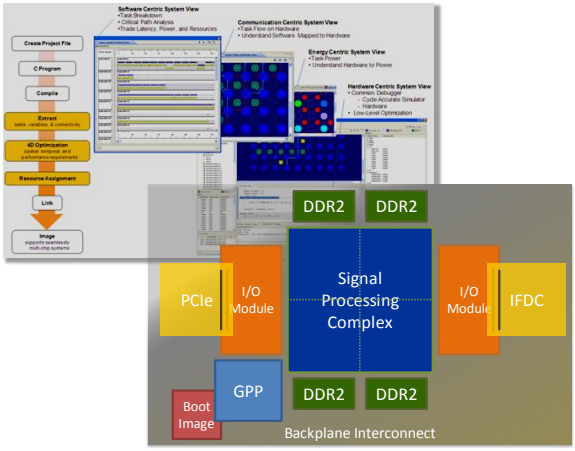

Figure 1: System Architecture and Development Environment

The RWDS supports functional verification, system analysis and performance characterization at speed in an integrated hardware development environment thereby increasing designer productivity. The system comprises a signal processing complex flanked by memory and various IO modules invoked as needed depending on the application waveform requirements. A GPP provides configuration and control and hosts the development environment as shown in Figure 1.

Considerations for configurability are supported throughout the system architecture to include programmability in the signal processing complex plus modular extensibility of the hardware platform including the potential to add signal processing modules within reach of a single RWDS backplane. Processing throughput can be further extended by daisy chaining backplanes together, support for which is native to the development environment requiring no change in programming model. Two processors arranged side-by-side are simply viewed as a larger processing array with processing cores interconnected no differently than if resident on the same chip.

System configurability extends to the IO modules in terms of the number and interface type that can be slotted alongside the processing module(s). A typical

SDR configuration includes dual data converters providing I/Q sampling to/from a radio front end and PCI Express to access the MAC development environment. The data converter card additionally includes provisions to emulate a variety of sampling interfaces varying with regard to sampling rate (including any oversampling), bit resolution and intermediate frequency (IF). The aim is to adapt a desired baseband signal interface to accommodate a range of off the shelf RF transceivers permitting design analysis configured as intended for product deployment.

2 RWDS System Overview

This paper describes the RWDS and mechanisms employed to facilitate waveform development in a real-time SDR environment. Section 2 presents a system overview including the RWDS hardware and software architectures. The tool flow and associated development environment are described in Section 3. A reference waveform implementation is discussed in Section 4. The paper concludes with a review of the objectives meant to enable seamless transition from real-time development environment to waveform implementation in a deployable product form factor.

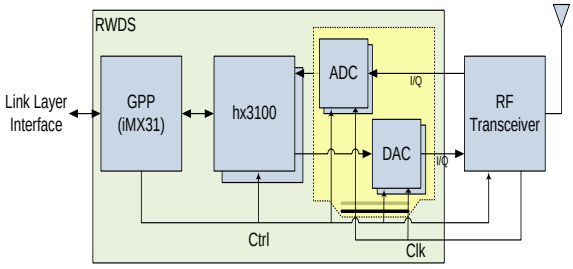

Figure 2: System Configuration

Primary responsibility for tasks associated with modulation/demodulation, scrambling/de-scrambling, FEC encoding/decoding, frame packing/unpacking and digital up/down conversion including resample filtering lies within the hx3100™, the current generation of HyperX™ programmable signal processors as depicted in Figure **2**.

A data converter card provides signal conversion to/from IF. Containing dual DACs and ADCs, it is capable of accommodating real or I/Q sampling over a broad range of signal bandwidths. It provides access to the radio front end via analog I/Q signal pairs. It also maps the radio control interface back to the signal processing complex.

Finally a General Purpose Processor (GPP) enables system configuration and hosts the operating environment including definition of the application waveform.

2.1 Hardware Architecture

The RWDS provides access to the processor fabric allowing real-time data streaming at baseband or low IF under full control of an integrated software development environment. Figure 3 shows the hardware architecture.

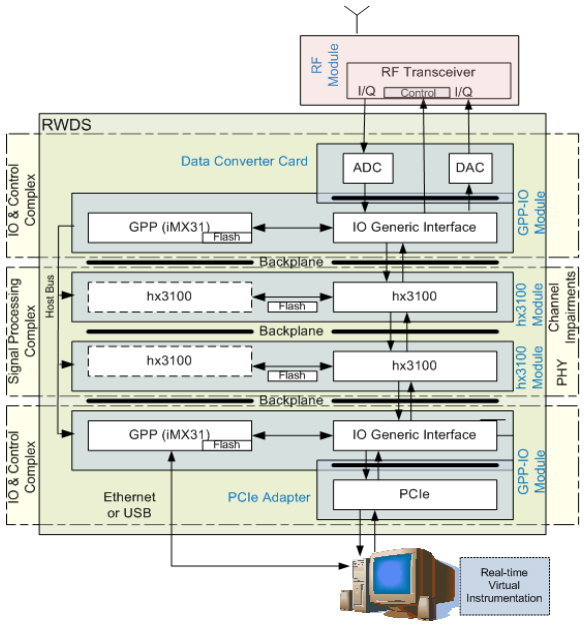

Figure 3: RWDS Hardware Architecture

Serving as the main building block for waveform processing, the processor module consists of one or more hx3100 low power processors each comprising a 10x10 array of processing cores and instrumentation for measuring core and IO power. The system supports seamless extension of the processing fabric adjoining multiple hx3100 modules via RWDS backplane. An overview of the HyperX architecture and its advantages in an SDR context are described in [1].

The GPP-IO Module includes an ARM based iMX31 processor, allowing applications to control system configuration options while providing direct access to the hardware through software control. The IO portion of the module supports a programmable interface to peripherals including data converters, Ethernet and USB adapters, VGA, DVI, etc.

The Data Converter card connects to the GPP-IO module as a daughter card. It contains two dual channel 14 bit, 150 MHz ADCs and one 14-bit 300 MHz dual channel DAC. Each data converter card supports 1x2 SIMO natively and can be ganged together to provide MIMO support with additional data converter cards. The PCIe Adapter card provides high speed real-time data streaming capability in to the hx3100 processor fabric. It plugs on to the GPP-IO module as a daughter card and connects to a host computer via a cable. To the host computer the

RWDS appears to be plugged into the PCIe slot and to the hx3100 processor the high speed data streams appear in or out of its communication ports.

As the motherboard of the system, the System Module supports any combination of hx3100 and GPP-IO modules with plug and play capability. The backplane is extensible across System Modules.

A wideband tunable RF transceiver provided with the RWDS supports carrier frequencies from 100 MHz to 1 GHz in 0.5 MHz steps while providing up-down conversion from-to IF. However, the Data Converter card supports interface to any third party RF front-end.

A variety of interface methods are provided to get information to and from the RWDS. Those methods include Ethernet and USB resident on the GPP-IO Module, Daughter card connections via the GPP-IO Module – e.g. PCIe, Data converter, etc. and direct SCSI cable connection to the development system backplane. SCSCI cable connection also permits daisy chaining multiple RWDS instances to extend processing throughput when necessary.

Numerous RFIC chipset providers offer complete solutions including data conversion. These solutions are invariably intended for dedicated use in a specific frequency band. For flexibility and minimization of engineering effort a better solution is to integrate off the shelf RF modules with a programmable interface such as the Data Converter card.

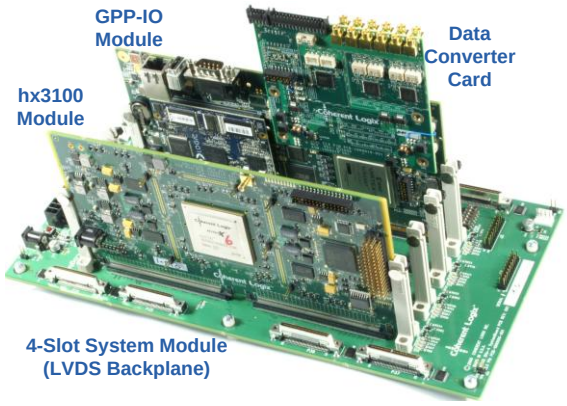

Figure 4: RWDS 4-slot Hardware

Figure 4 shows the RWDS hardware mapped to a 4-slot backplane. A typical system configuration supporting radio waveform development affording different test configurations from baseband to RF is described below referenced to Figure 3:

- ***Slot 1*** – populated with the GPP-IO Module housing the Data Converter card.
- ***Slot 2*** – contains an hx3100 Module providing real-time channel emulator (e.g. Multipath + AWGN) or pass through running on the hx3100 processor.
- ***Slot 3*** – consists of an hx3100 Module implementing the baseband radio waveform application.
- ***Slot 4*** – contains another GPP-IO Module with the PCIe adapter card providing connectivity to a host PC. The PCIe interface supports real-time virtual instrumentation such as constellation display and BER/PER plots. This interface might additionally be used to access the upper layer development environment on a host PC.

This configuration supports test and verification of the Transmit-Receive chain at baseband and IF. For air interface testing, the Data converter card is connected to an RF Transceiver and the channel emulator Module is removed. Depending on the bandwidth requirements, interface requirements and system access needs for waveform development, testing and verification, the GPP-IO and hx3100 modules can be connected in different ways across the System Module.

2.2 Data Converter Interface

Scaling to accommodate a wide variety of application waveforms requires flexibility in the data converter interface. The RF front-end for instance, sourced from a number of third party vendors, will exhibit vast differences in signal bandwidth and IF tuning range. It is the responsibility of the data converter card to match the sampling rate and bit resolution required for baseband processing. The RWDS data converter interface offers a high degree of configurability to address an anticipated range in front end characteristics as shown in Table 1 alleviating the need for dedicated converter cards to meet individual transceiver requirements.

	ADC	DAC
Sampling Rate	≤ 150 MS/s	≤ 600 MS/s
IF Frequency	0 - 450 MHz	
Signal Format	Differential LVDS (configurable for single ended)	
Bit Resolution	14-bits (11.5 ENOB)	14-bits (11.9 ENOB)
Resampler	CIC, Integer, Fractional implemented in separate logic	
IF Conversion	NCO + Digital up/down conversion implemented in separate logic	

Table 1: Data Converter Interface Configurability

2.3 System Software

Hosted on the GPP of the GPP-IO module, a set of APIs provide an abstracted view of the RWDS

development platform. This provides the user the ability to design and test applications specific to their problem domain while minimizing concerns about the underlying implementation. Implemented in C/C++, the APIs take the object approach to the architecture – each component builds upon the services provided by underlying components. For characterizing the system, there are additional methods for interrogating the internal power and environmental sensors on the system. At a high level, the API provides an easy and intuitive way to determine a systems configuration dynamically. This allows the application to determine if the hardware is appropriate and if so to load the appropriate software, bitstreams, and other images to bootstrap the electronics. Lastly, the hierarchical nature of the API lends itself to extension by the application engineer.

An interactive and scriptable diagnostic environment that facilitates both high-level debugging as well as register-level access to the platform is also provided with the RWDS. The diagnostics provide a set of tools for use by engineering and manufacturing for board bring up, test, and debug. Items of interest to engineering include the ability to arbitrarily program registers and memory, communicate across the various system busses, read/write sensors and EEPROMs, and perform measurements if equipped. Furthermore, routine tasks are made much easier by a full command line interface and interactive shell, command history, and embedded TCL scripting. The diagnostics provide sufficient control to manipulate the test environment to eliminate the cycle of edit code, compile, test, and repeat.

3 Development Tools

The HyperX ISDE [1] provides an ANSI-C based unified development and verification flow without requiring detailed knowledge of the HyperX architecture. Design optimizations (e.g., trading performance, power, and latency) are performed without changing C code. The flow is platform independent until compile time and has proven greater than 5x reduction in development time over current (DSP/FPGA) design flows. This is based on porting multiple waveforms including spread and non-spread modes associated with Soldier Radio Waveform (SRW) and a scalable multi-carrier LTE system.

Recent addition to the RWDS development tools includes the Model-based design flow [5]. This is an automated flow tightly integrated with the HyperX ISDE that translates a Simulink design to C code for execution on HyperX processor chip(s). It provides a rapid and seamless translation from a Simulink design to an implementation on a HyperX processor chip or system of chips without the user writing any source code. The advantage of the Simulink translation flow is to very quickly get a measure of the performance (latency and throughput) and physical requirements of the application (data memory, instruction memory, and communication needs) and to get a starting point for creating a more efficient implementation. In cases where parts of the Simulink design do not meet the performance achievable with a hand crafted and tuned version specifically targeted to the HyperX processor architecture, the flow supports library replacement of Simulink derived blocks with optimized libraries of HyperX implementation.

The real-time analysis (RTA) tools provide a powerful real-time, software-based analysis and characterization environment for evaluating the performance of SDR receivers in the presence of dynamic channel conditions as well as additive white Gaussian noise [6]. Test parameters (e.g., SNR, multi-path model parameters and packet size) are sent to the RWDS, messages are automatically generated, transmitted through a dynamic channel emulator, and presented to the receiver. The decoded results are compared to the original message and relevant statistics collected. All operations are under the control of an interactive host system program. Since the system runs in real-time, results can be accumulated as fast as the waveform can operate. Hundreds to thousands of test results per second can be accumulated and used to characterize the performance of multiple receiver configurations running on the RWDS.

As a component of a complete SDR platform, the user is also expecting the ability to use optimized libraries to develop their waveforms. Library Elements are intended to provide reference implementations organized with respect to categories of inter-related and hierarchical SDR functionality as shown in Figure 5 below.

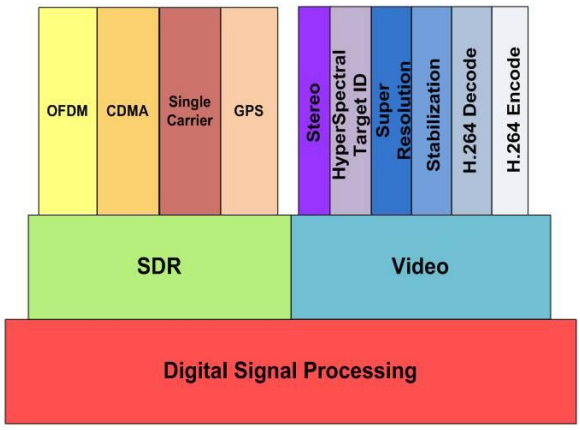

Figure 5: Library Element Categories

This facilitates an efficient waveform development in and around the HyperX processor and also promotes modular reuse from a diverse palette of standard processing tasks.

Library elements are categorized according to recognized commonalities in underlying structure, application or system design intent to maximize waveform coverage with minimal development overhead. A methodology for developing, distributing and maintaining library elements in a consistent, repeatable, sustainable framework is included.

4 Reference Waveform Implementation

4.1 Scalable OFDM System Overview

The RWDS includes a reference waveform implementation derived from an OFDM functional library as depicted in Figure 6. Developed around the Inverse and Forward Fast Fourier Transform pair (I/FFT), the reference waveform extends library functionality to include scalability in signal dimension to accommodate a range of channel bandwidths.

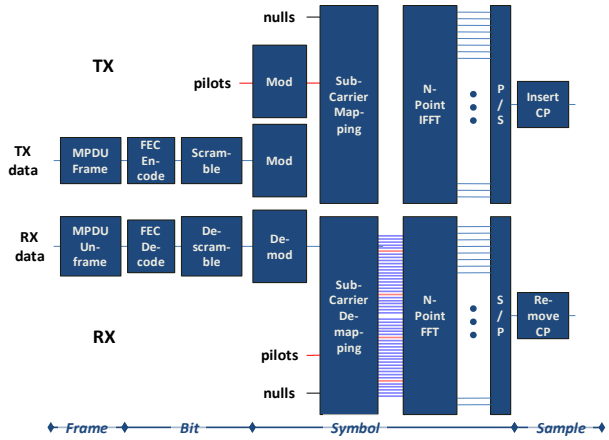

Figure 6: Scalable OFDM System Block Diagram

4.2 Frame Structure

Data frames are organized according to the Resource Block (RB) structure illustrated in Figure 7.

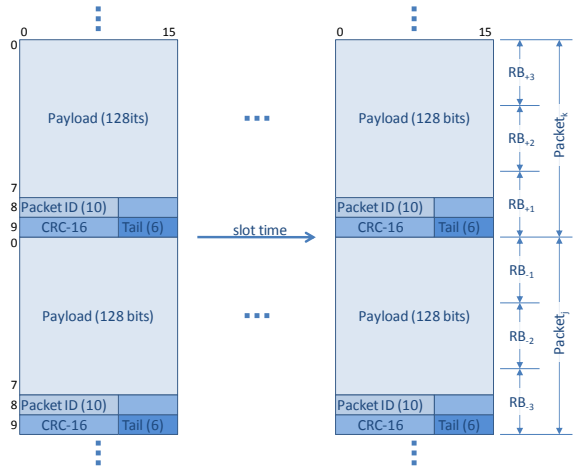

Figure 7: Frame Structure

The frame structure includes a packet ID and CRC to facilitate performance characterization. Tail bits are inserted at the end of each frame to return the encoder trellis to a known state.

The waveform employs fixed modulation to ease receiver implementation demonstrating scalability in signal dimension ranging from 128 to 512-pt I/FFT supporting 1.92-7.68 Mb/s carrying capacity.

4.3 Design Layout and Resource Summary

The design resource mapping for Scalable OFDM is shown in Figure 8. Capabilities available with the RWDS for real-time analysis include constellation display and BER/PER assessment, examples of which are also shown in the figure.

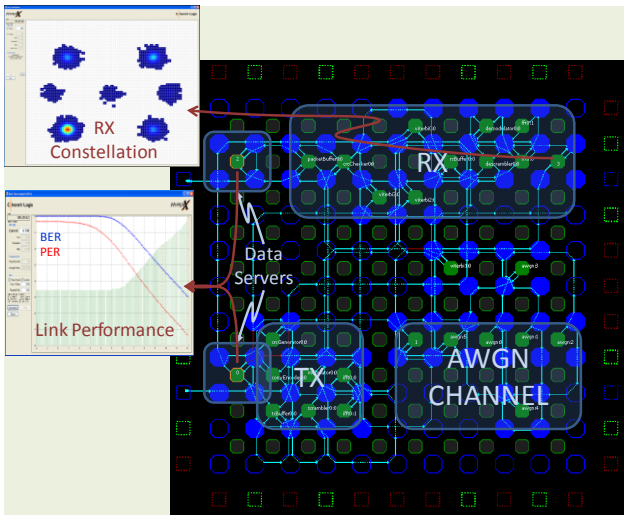

Figure 8: Scalable OFDM Design Layout and Analysis

5 Closing Remarks

We presented the RWDS, a scalable hardware and software development system providing access to the HyperX signal processing complex in a real-time development environment. The system enables at speed waveform design and analysis from initial algorithm exploration through final implementation and performance characterization.

The path to form factor prototype entails moving the signal processing chain, through the data converter interface, to a shared platform placed alongside the RF transceiver. Instrumentation and glue logic associated with the development system are removed. The iMX hosting the development environment can also be eliminated migrating responsibility for system configuration and control to another GPP that runs the MAC processing, for instance. The application software stack, having been runtime qualified on the RWDS, is fully preserved requiring no code modification when mapped to a deployable product form factor.

6 References

[1] Spectrum Signal's SDR-5001 radio transceiver unit, http://www.spectrumsignal.com/products/pdf/sdr_5001.pdf

[2] Lyrtech SFF SDR development platform, http://www.lyrtech.com/Products/SFF_SDR_development_platforms.php

[3] Iveia SDR Development System, http://www.iveia.com/pro_sdr.php

[4] B.A. Dalio and K.A. Shelby, **"**The Implementation of OFDM Waveforms on an SDR Development Platform supporting a Massively Parallel Processor", Proceedings of the SDR '09 Technical Conference and Product Exposition.

[5] M Beardslee and M Hall, "Rapid Prototyping of Communication Waveforms from a Model-based Design Language", Coherent Logix, Inc. July 2010. Submitted for selection at Proceedings of the SDR '10 Technical Conference and Product Exposition.

[6] B.A. Dalio and I Aguayo, **"**'Real-Time, Software-Based Characterization of Receiver Dynamic Channel Performance on an SDR Development Platform", Coherent Logix Inc. July 2010. Submitted for selection at Proceedings of the SDR '10 Technical Conference and Product Exposition.