SDC | Software Defined Communication Testbed

Overview

Drexel University's SDC seeks to provide a rapid prototyping system for creating radio, optical, and ultrasonic communication systems. SDC aims to provide a wide range of flexibility across OFDM based standards for research as well as tweaking aspects within standards. SDC derives its capability from a full FPGA based SOFDM (Scalable OFDM) baseband implementation designed to give hardware speed with software flexibility through parameters controlled through memory registers from software. Each module of the baseband layer is built with controllers specific to its functionality and intelligence to scale, even during run-time. SDC's research oriented framework makes it easy to inject data, control and algorithms between modules or even into them, with debug ports extracted across the entire design to test outcomes. There is also an interface to allow users to rapidly prototype algorithms in MATLAB.

The Problem

There are several prominent SDRs in the academic community and the Drexel Wireless System Lab (DWSL) has utilized a mix of these platforms for research purposes. However, the majority of SDR platforms that are widely available to the academic and industrial research community have several notable limitations, including:

- Sampling rates and processing capabilities limited so that systems are predominantly limited to IEEE802.11 data rates
- Lack of available FPGA fabric and programming flexibility to allow implementation of newest algorithms proposed by the communications and signal processing community
- Exclusive focus on radio-frequency based communications

Possibilities

You can prototype baseband for OFDM communication standards that vary in modulation schemes, number of subcarriers, interleaving rates, and coding rates on the same testbed.

You can rapidly prototype an OFDM-based baseband that uses 128 subcarriers to selectively dodge noisy channels through selective loading.

You can create PHY algorithms that will integrate well into SOFDM due to its latency insensitive core.

You can test specific areas of your baseband by bypassing modules you are not concerned with.

You can build systems that utilize a conjugate symmetric IFFT/FFT process to prototype real, non-complex signaling appropriate for optical communications.

Our Solution

To address these limitations, using funding from NSF (CNS #0923003 and CNS #0854946), DWSL has developed the SDC Testbed for release to the wireless academic and industrial research community. The SDC Testbed seeks to provide a research and development platform capable of designing and prototyping next generation wireless communication standards which make use of radio, optical, and ultrasonic communication modalities. The vision for this platform builds on the current principles of SDR. Specifically, it extends this concept through utilization of propagation media beyond the RF spectrum while meeting the high bandwidth demands of emerging communications standards. The project will provide a tool for wireless research communities. Furthermore, it is designed to provide a cohesive and affordable hardware/software infrastructure for fast and flexible algorithm development across multiple layers of the communication stack.

Features

- Supports IFFT/FFT block sizes of: 32, 64, 128, 256, 512 and 1024. This makes research of different communication standards viable on the SDC along with areas concerning adaptive spectral learning and spectral optimization.
- Supports BPSK, 4QAM, 16QAM and 64QAM modulation schemes with room and instructions to easily upgrade to higher schemes.
- Uses a Convolution Encoder along with a Viterbi Decoder to support coding rates of 1/2, 2/3, 3/4, 5/6.
- Supports both inter- and intra-symbol interleaving with flexibility to change the block sizes or even skip the block completely.
- SDC's packet detection consists of a scalable core that scales from 16 to 64 point complex correlation.
- SDC's variable packet header plus payload structure makes it easy to send, receive and even engineer headers to optimize control and data flow for desired test routines.
- Intelligent control makes it easy to load subcarriers on the fly and this control also applies to loading pilots or even nulling subcarriers based on preset or runtime criteria.
- SDC's training and channel estimation blocks are capable of scaling and provide flexibility for testing new estimation algorithms.
- SDC's co-simulation framework through MATLAB can be used to build and test algorithms and designs introduced into the core separately on the host PC, increasing turnaround times.

System Requirements

- MATLAB R2009b or higher
- ModelSIM 6.6 SE or higher
- Xilinx SysGEN 12.1 or higher
- Xilinx ISE 12.1 or higher

Hardware Requirements

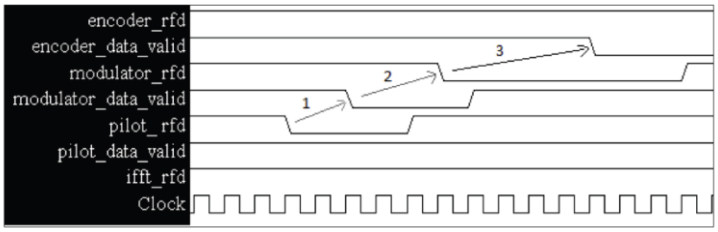
- XILINX Virtex 6 ML605
- Nutaq Radio420

Key Architectural Design

The SOFDM core seamlessly accommodates design time and run time changes to the configuration parameters for the codec, modulation, piloting, and IFFT stages. SOFDM is a comprehensive design with IPs engineered to reuse logic across implementations rather than selecting between pre-built propagation paths to scale with the chosen settings. The table below lists a few of the various parameters of interest specific to each stage.

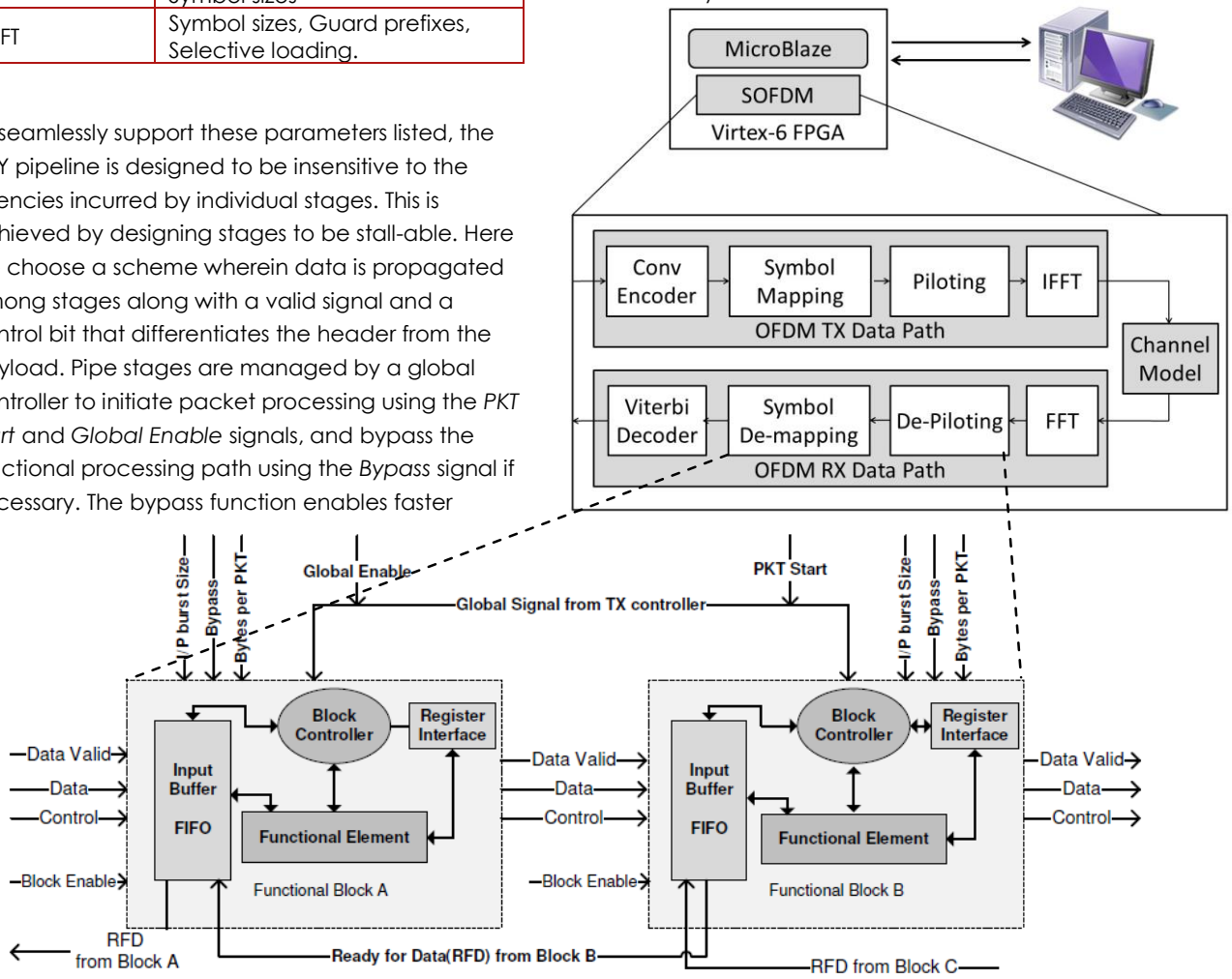
Pipeline Stage	Configuration Parameters
Encoder	Coding rate, Coding polynomial
Interleaver	Inter symbol scaling, Intra symbol scaling
Modulation	Modulation scheme, Data mapping values
Piloting	Pilot positions, Pilot sizes, Symbol sizes
IFFT	Symbol sizes, Guard prefixes, Selective loading.

To seamlessly support these parameters listed, the PHY pipeline is designed to be insensitive to the latencies incurred by individual stages. This is achieved by designing stages to be stall-able. Here we choose a scheme wherein data is propagated among stages along with a valid signal and a control bit that differentiates the header from the payload. Pipe stages are managed by a global controller to initiate packet processing using the *PKT Start* and *Global Enable* signals, and bypass the functional processing path using the *Bypass* signal if necessary. The bypass function enables faster debugging, allowing end-users to skip the functional processing path and simply pass on the input data to the following stage to pinpoint the errors through isolation. Each stage uses the generic architecture consisting of: an input buffer, a register interface, a



block controller and a functional element. All the four components mentioned above are engineered to the functional specificity of the module's function within the PHY.

The waveform above shows the propagation of a stall originating in the piloting block being acknowledged by other PHY modules during a necessity for stall.



This Project is NSF funded through grants CNS #0923003 and CNS #0854946.

wireless.ece.drexel.edu

The Drexel researchers may be contacted for more information as follows:

Kapil Dandekar
Professor & Associate Dean for Research
and Graduate Studies
Electrical and Computer Engineering Department
College of Engineering
(215) 895-2228

DWSL
Rm. 325 Bossone Building
3100 Market Street
Philadelphia, PA, 19107