Dataflow Modeling and Design for Cognitive Radio Networks

Lai-Huei Wang
Shuvra S. Bhattacharyya
ECE Department
University of Maryland
College Park, MD 20742
(laihuei,ssb)@umd.edu

Aida Vosoughi
Joseph R. Cavallaro
ECE Department
Rice University
Houston, TX 77005
(vosoughi,cavallar)@rice.edu

Markku Juntti, Jani Boutellier
Olli Silvén
Dept. of Communications Engr.
Dept. of Computer Science & Engr.
Univ. of Oulu, Finland FI-90014
(markku.juntti,jani.boutellier,
olli.silven)@ee.oulu.fi

Mikko Valkama
Dept. of Communications Engr.
Tampere Univ. of Technology
Tampere, Finland FI-33101
mikko.e.valkama@tut.fi

Abstract—**Cognitive radio networks present challenges at many levels of design including configuration, control, and cross-layer optimization. In this paper, we focus primarily on dataflow representations to enable flexibility and reconfigurability in many of the baseband algorithms. Dataflow modeling will be important to provide a layer of abstraction and will be applied to generate flexible baseband representations for cognitive radio testbeds, including the Rice WARP platform. As RF frequency agility and reconfiguration for carrier aggregation are important goals for 4G LTE Advanced systems, we also focus on dataflow analysis for digital pre-distortion algorithms. A new design method called parameterized multidimensional design hierarchy mapping (PMDHM) is presented, along with initial speedup results from applying PMDHM in the mapping of channel estimation onto a GPU architecture.**

I. INTRODUCTION

Efficient and cognitive use of frequency spectrum has been recognized as a key technology to enhance the efficiency of wireless systems. Both the NSF and Tekes have identified these research challenges and needs in several recent workshops on Future Wireless Networks [1], Enhancing Access to the Radio Spectrum (EARS) [2], and the TRIAL Cognitive Environments and Testbeds [3]. This WiFiUS (Wireless Innovation between Finland and U.S.) project seeks to address these challenges from the RF circuit level to efficient baseband processor architectures through a multi-disciplinary collaboration among the four universities which specialize in RF, algorithms, design modeling and testbeds.

Concurrent use of a plurality of network and radio technologies is a fact which will remain. All this requires flexibility and configurability from the transceivers regardless of the fact of how the spectrum sharing itself is realized. Increasing bandwidths and data rates pose more and more challenges to the whole baseband (BB) processing chain and also the radio frequency (RF) processing. The major challenge comes from the fact that the variety of nodes will increase.

At the same time, *flexibility and configurability* of the implementation at the RF, baseband, and MAC layers, with cross-layer modeling and control, will be important to *realize the efficiency potential of spectrum sharing*. The flexible use of RF spectrum over a multitude of different frequencies requires advanced configurable radio devices. Receiver configuration

based on programmable paradigms has received attention in recent years but practical solutions are still lacking, especially for the radio frequency (RF) components. We plan to build upon our related work on compensation of nonlinear distortion [4]. At the same time, programmable baseband computation and related design chains have significantly developed. This enables more efficient control of computational resources and hardware. The software based adaptive configuration of radio frequency chains is still in its infancy, but it is a key ingredient of the frequency agile radios needed for cognitive devices and flexible RF spectrum use. A major gap which we are seeking to address in this project is the lack of models, structured design methods, and comprehensive understanding of realistic configurable RF chains and radio modules.

In this paper, we focus on the dataflow modeling and design aspects and show how these can be applied to physical layer algorithms that are used in cognitive radio networks. Dataflow methodologies are a promising candidate for the modeling, analysis and verification of cognitive radio systems [5], [6]. Dataflow also offers an excellent basis for building automated synthesis tools that generate actual system implementations out of models. As dataflow models are abstract and platform independent, the same model can be used to generate implementations for very different devices and implementation constraints from low-power sensor nodes to high-end mobile terminals.

II. PARAMETERIZED DATAFLOW MODELING FOR COGNITIVE RADIO SYSTEMS

A. Background

1) Parameterized Synchronous Dataflow: Dataflow modeling techniques are widely used in the design and implementation of communication systems (e.g., see [7]), and major commercial tools, such as Agilent SystemVue and National Instruments LabVIEW, provide dataflow-based design capabilities for communication system design.

Parameterized dataflow is a meta-modeling technique that can significantly improve the expressive power of an arbitrary dataflow model that possesses a well-defined concept of a graph iteration [8]. Parameterized dataflow provides a method to systematically integrate dynamic parameter reconfiguration into such models of computation, while preserving many

of the properties and intuitive characteristics of the original models. The integration of the parameterized dataflow meta-model with *synchronous dataflow* (*SDF*) provides the model of computation referred to as *parameterized synchronous dataflow* (*PSDF*). PSDF offers valuable properties in terms of modeling systems with dynamic parameters, supporting efficient scheduling techniques, and natural integration with popular SDF modeling techniques [9].

2) Multidimensional Design Hierarchy Mapping: The *Multidimensional Design Hierarchy* (*MDH*) mapping method is a design method that builds on the multidimensional synchronous dataflow (MSDF) [10] model of computation, and facilitates hierarchical exploitation of parallelism in multidimensional signal processing applications [11]. This approach allows designers to explore alternative implementations in a manner that separates platform-specific parallel processing optimization from behavioral specifications, thereby enhancing portability and trade-off exploration. More specifically, the multidimensional design hierarchy model provides an intermediate model that offers a formal linkage between hierarchical layers of parallelism in the target platform and corresponding subsystems of the application that will be mapped onto these layers. In the MDH approach, graph clustering and MDSDF dataflow analysis are applied to map applications to target platforms that employ parallelism at multiple levels [11].

In this paper, we integrate the PSDF model for dynamic parameter reconfiguration and the systematic mapping method of MDH to offer a novel design framework for dynamically structured signal processing systems that require significant amounts of run-time flexibility in reconfiguring operations, subsystems, and system parameters. We refer to our proposed framework as *Parameterized Multidimensional Design Hierarchy Mapping* (*PMDHM*).

B. PMDHM Framework

In this section, we present our proposed PMDHM framework for dataflow-based design, which is targeted to the flexible, multi-level reconfigurability, and intensive real-time processing requirements of emerging wireless communication systems.

A PSDF specification is composed of three cooperating PSDF graphs, which are referred to as the *init*, *subinit*, and *body* graphs of the specification. Actors and edges in PSDF graphs can be annotated with arbitrary parameters, which can be changed at runtime. Such actors and edges correspond, respectively, to functional components and intra-component connections in signal processing flowgraphs (e.g., see [7]). Parameters of actors and edges in a PSDF graph can only change between iterations of the enclosing dataflow graph. The init graph executes once during each iteration and is allowed to configure the associated subinit and body graphs. The subinit graph executes once during each execution of the corresponding PSDF subsystem. During such an execution, the subinit graph executes; new parameter values computed at outputs of the subinit graph are propagated to corresponding parameters in the body graph; and then the body graph executes based on the updated set of parameters. Parameter changes that are computed by the subinit graph cannot modify the consumption and production rates of the enclosing PSDF actor.

For selected subsystems in a PSDF-based system design, a new design transformation called the *parameterized multi-level hierarchical transformation* (*PMHT*) can be employed to efficiently map the subsystem to a given target platform that employs parallelism at multiple levels (e.g., instruction-level, accelerator-level, and inter-core parallelism). Designers can thus select subsystems that have critical constraints (e.g., on performance, energy efficiency or resource utilization) for application of the PMHT.

For each alternative body graph that results from different sets of parameter configurations (e.g., application or subsystem modes) in the init graph, the PMHT approach transforms an application graph with parameterized production and consumption rates (i.e., dataflow rates that are represented as functions of system parameters) into a hierarchical organization of graphs such that the structure of the hierarchy helps the designer to map the design onto the hierarchical parallel structures in the target platform.

When applying the PMHT, actor clustering is first performed to combine one or more connected actors into units that are viewed as "supernodes" from the previous hierarchical level. We refer to these units as *mapping clusters*. In each mapping cluster, special interface actors, called *intin* and *intout* actors, are inserted to represent the injection of data into and out of the associated supernode. The standalone dataflow graphs representing the internal functionality of the mapping clusters, called intermediate representation (IR) graphs, can be utilized for further implementation analysis at each level. Presently, this clustering process and the associated transformation process is carried out by the designer, through systematic guidance by our proposed PMDHM methodology. Automating these processes of clustering and transformation are useful directions for further investigation, which we are actively exploring in our ongoing work.

The supernodes constructed using this process are used for efficient mapping of flowgraph structures into architectures that employ multi-level parallelism. Such architectures, such as programmable digital signal processors and graphics processing units (GPUs), are becoming increasingly important in the realization of cognitive radio systems. Given a target platform T, we let $n(T)$ denote the number of levels of parallelism (e.g., instruction-level or intra-core parallelism, as described above) in the platform that we explicitly consider in the mapping process, and associated with each level of parallelism, we assign a unique index k ($1 \le k \le n(T)$).

Given a mapping cluster C and an actor α within C, we define the *hierarchical firing vector* $H(\alpha)$ for α to be a vector that represents, at each level of target platform parallelism, how many simultaneous (parallel) executions of α can be supported based on the flowgraph structure associated with that level. $H(\alpha)$ is a vector that has $n(T)$ elements, each of which is a positive integer. $H(\alpha)[i] = j$ means that at level i of the target platform T, up to j executions (firings) of α can execute in parallel.

The vector $H(\alpha)$ thus provides in a concise and precise form the parallel processing potential of a given signal processing flowgraph component relative to a given target platform. For application to PSDF-based design and implementation, we extend the concept of the hierarchical firing vector H
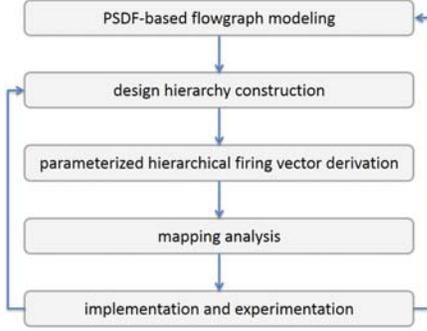
Fig. 1. Illustration of our proposed PMDHM-based design methodology for cognitive radio system design.

Fig. 2. Dataflow graph for channel estimation.

so that the vector elements can be positive-integer-valued *parameterized expressions* in actor, subsystem, and system-level parameters. The resulting *parameterized hierarchical firing vector P* therefore captures variations in parallel processing potential in terms of relevant application parameters. Such variations can then be analyzed to provide efficient methods for reconfiguring processing structures as different application modes (e.g., different communication standards or operational constraints) are encountered at run-time. As an example, an expression of the form $P(\alpha)[i] = f(p_1, p_2)$ can be used to represent (in terms of a given function f) the relationship at platform level i between parallel processing potential and a pair of parameters (p_1 and p_2).

Figure 1 summarizes the developments of this section with an illustration of our proposed PMDHM-based design methodology for cognitive radio system design. In Section II-C, we demonstrate the application of this design methodology to a practical communication system example.

C. PMDHM Design Example

In this section, we develop a case study, which demonstrates our proposed new PMDHM design methodology on the GPU-based implementation of channel estimation for wireless communication systems. Through this concrete example, we demonstrate how PMDHM can be applied to efficiently and systematically explore implementation trade-offs across various design configurations for the targeted channel estimation system.

Orthogonal frequency division multiplexing (OFDM) is applied extensively to high-speed wireless communication systems because of its spectral efficiency, robustness in terms of multipath propagation, and high bandwidth efficiency [12]. Channel estimation (CE) is an important issue in many wireless OFDM systems for demodulation and decoding. The fading channels of OFDM systems, in general, can be modeled as two-dimensional (2D) signals in terms of time and frequency.

Pilot-assisted channel estimation is one of the most popular schemes for estimating channel response in OFDM systems. This scheme operates by transmitting a pilot signal that is known at both the transmitter and receiver sides. At the receiver side, after computing the channel responses on pilot
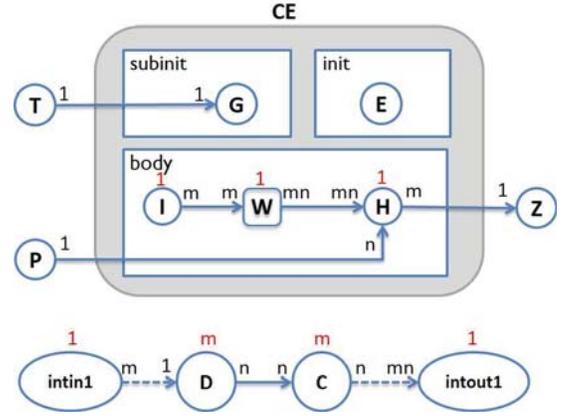
subcarriers, a 2D interpolation method is used to estimate the channel responses on data subcarriers. In this case study, we apply the technique of 2D minimum mean square error (MMSE) filters [13] for the interpolation. The channel response on data i can be found by $h_i = w_i r$, where r is the vector of channel responses on pilot subcarriers, and w_i is the coefficient vector of the MMSE estimator for data i. This coefficient vector can be computed as $w_i = (R + N_0 I)^{-1} d_i$, where R is the auto-correlation matrix of the responses on pilots, d_i is the cross-correlation vector between responses on data i and pilots, N_0 is the noise variance, and I denotes the identity matrix.

Using the PMDHM framework, we model the targeted 2D channel estimator as shown in Figure 2 (top). Here, m and n are two parameters that represent the numbers of pilots and data samples, respectively, in a 2D resource block (i.e., the basic unit for 2D channel estimation). Since the parameters m and n both affect the production and consumption rates (dataflow rates) of PSDF actors in the system, they can only be configured in the init graph. This is a restriction imposed by the PSDF model to enhance predictability and the potential for deriving efficient implementations. Actor T generates a token that encapsulates the the pilot pattern that is to be applied. In the subinit graph, actor G reads this pilot pattern token, and correspondingly configures the parameters related to the pilot pattern in the actors of the body graph.

The source actor P produces tokens associated with the responses on the pilot subcarriers, which are read in the body graph. The body graph contains three actors, I, W, and H. Based on the parameters set in the init and subinit graphs, the source actor I generates locations for the data subcarriers, which are consumed by actor W to compute the corresponding coefficients of the 2D MMSE filter. Actor H reads filter coefficients from W and pilot channel responses from P to interpolate and output the resulting responses on the data subcarriers. Finally, the sink actor Z reads and stores tokens that encapsulate the responses.

In our experiments, we apply GPU acceleration to actor W to enhance performance when implementing our dataflow model of the channel estimation system. The regular computational structure of this actor makes it well suited to GPU

(a) Speedup gain.

(b) Kernel processing times.
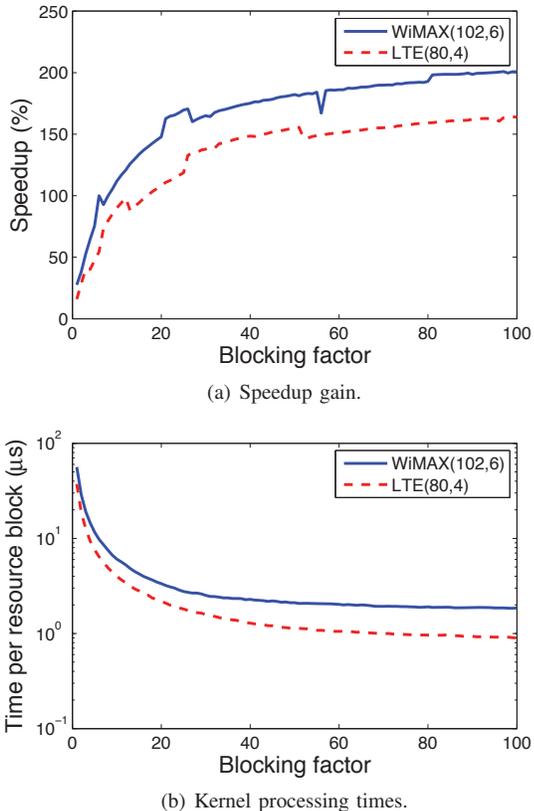
Fig. 3. Experimental results for PMDHM-based implementation of channel estimation.

mapping. The PMHT method is applied to W in conjunction with optimized vectorization (i.e., application of block processing to improve throughput and pipeline utilization) to exploit the multiple levels of parallelism in the targeted GPU. The IR graph for the first level is shown in the body graph of Figure 2. Similarly, the second-level IR graph is illustrated in Figure 2 (bottom), where actors D and C compute the cross-correlation d and the filter coefficients, respectively, for the input data subcarriers.

Using the PMDHM approach, the parameterized hierarchical firing vector for W is found to be $P = (1, m)$. By applying vectorization with a parameterized vectorization factor (degree of block processing), denoted by β, we can implement actor W on the GPU with the parameterized hierarchical firing vector $Q = (\beta, m)$. In this form of vectorization, β resource blocks are required for each input block on which actor W operates.

In our experiments, an NVIDIA GTX260 GPU and an Intel Xeon 3GHz CPU are used. We compare the performance between (1) the CPU implementation (all actors mapped to the CPU), and (2) a heterogeneous implementation with W mapped to the GPU and all other actors mapped to the CPU. We carry out these experiments for the two sets of parameter configurations $(m, n) = (80, 4)$ and $(m, n) = (102, 6)$, which are used in the LTE and WiMAX standards, respectively. Figure 3(a) shows our experimental results in terms of the speedup gain of the heterogeneous implementation compared to the CPU implementation.

From the experimental results, we see that the heterogeneous implementation outperforms the CPU implementation for sufficiently large β ($\beta > 6$ and $\beta > 16$ for WiMAX and LTE, respectively). This is because optimized performance on the GPU requires effective utilization across a large number of threads [14]. Figure 3(b) illustrates the processing time per resource block for the GPU kernel (GPU-targeted functional component) associated with actor W. This measurement includes the time for memory transfer between the CPU and GPU.

From the results, the processing times for both cases are high when the numbers of threads are small (i.e., for small values of β). As β increases, the processing times decrease rapidly (since increasing numbers of threads are executed in parallel), and this performance improvement starts to saturate at approximately $\beta = 30$ and $\beta = 40$, respectively, for the WiMAX and LTE settings. As shown in Figure 3(a), a large block processing (vectorization) factor can provide significant speedup on the heterogeneous platform — e.g., 200% and 155% speedup gain with $\beta = 100$ for the WiMAX and LTE settings, respectively. However, such performance improvement leads to an increase in system latency since more resource blocks must be available to satisfy the vectorization requirements.

In summary, the channel estimation case study presented in this section demonstrates the flexibility and potential for systematic design trade-off exploration offered by our proposed PMDHM framework. The agility of our design methodology is demonstrated in this case study by the application to heterogeneous platforms (GPU and CPU devices), communication standards (WiMAX and LTE), and system-level parameter configurations through a unified framework for modeling and analysis. Our PMDHM-based design methodology therefore provides a promising foundation for cognitive radio system design, where diversity in processing resources and functionality must be managed strategically and reliably at run-time under stringent operational constraints.

III. COGNITIVE RADIO REALIZATION, SYNTHESIS AND PERFORMANCE EVALUATIONS

In this section, we look at the the complexities in cognitive radio realization and non-contiguous channel aggregation in particular. As an example for required baseband processing algorithms in cognitive radio architectures, we focus on digital pre-distortion and show how dataflow modeling can help achieving a flexible design. Dynamic spectrum access [15] and cognitive radio systems have attracted a lot of interest recently as they facilitate a more efficient use of RF spectrum. A cognitive radio framework enables an unlicensed (secondary) user to utilize a licensed spectrum band as long as it conforms to the rights of the licensed (primary) users of the band. One approach to realizing dynamic spectrum access is allowing a secondary user to communicate across several non-contiguous frequency bands while it avoids interfering with primary user transmissions. As a result, the secondary user can benefit from an aggregation of channels which sum up to a sufficiently high overall transmission bandwidth. For example, non-contiguous orthogonal frequency division multiplexing (NC-OFDM) is an efficient method for the above-mentioned non-contiguous channel aggregation approach [16]. NC-OFDM advantages

from the capabilities of OFDM transmission and provides a suitably configurable data transmission solution for cognitive radio systems by activating and deactivating sub-carriers based on dynamic spectrum sensing measurements [17].

One drawback of the above-mentioned technique is that the RF front-end components impairments become more apparent because the secondary user operates across multiple non-continuous frequency bands and over a potentially large dynamic range. For example, the power amplifier (PA) is one of the important components in RF front-end which is known for its non-linear behavior in wide bands. For high peak-to-average power ratio (PAPR) of multi-carrier transmissions, PA's non-linearity generates out-of-band spectral leakage that may interfere with transmissions on the neighboring channels. As a result, PA's non-linear distortion can result in unreliability in cognitive radio systems. Digital pre-distortion, which is a well-studied technique to compensate for the non-linearity of PA, facilitates reliable spectrum sharing of primary and secondary users [18], [19].

Advanced cellular communication standards have considered the scarcity of RF spectrum and the benefits of spectrum sharing techniques. To achieve high data rates, carrier or channel aggregation (CA) has been contained within the specification of 4G LTE Advanced [20]. LTE-A carrier aggregation increases the overall transmission bandwidth by utilizing multiple intra-band contiguous or non-contiguous channels, but the challenge is to ensure that the performance is not degraded as a result of operation on a wider bandwidth. Furthermore, due to high levels of spectrum fragmentation, in many cases only small bands may be available and therefore carrier aggregation over more than one band (inter-band aggregation) is also considered in LTE-Advanced standard specification. Power amplifier spectral regrowth (inter-modulation) problem becomes even more complex in inter-band aggregation.

Therefore, implementing the physical layer of cognitive radio systems is challenging; the physical layer design must be flexible and reconfigurable as the medium characteristics are dynamically changing. For example, the digital pre-distortion filter that mitigates the power amplifier non-linearity over non-contiguous channels must be adapted and reconfigured based on the channel characteristics. One approach is to make the pre-distortion processing parametrized. However, unfortunately the current physical layer design methodologies do not consider flexibility and reconfigurability as much as it is needed. A major complexity in the physical layer design which takes a lot of design effort is to make the parallel paths of sub-blocks in the system properly synchronized. In these designs, even a small change in a sub-block may result in a need for a complete re-timing of the design. In a graphical digital signal processing (DSP) design environment like Xilinx System Generator, re-timing may require inserting delay elements in data-paths and when the DSP hardware is modeled using a hardware modeling language like Verilog, re-timing requires rethinking the size of the intermediate buffers and modifying the control unit finite state machine.

The inflexible nature of the physical layer design makes design modifications hard. For the same reason, the on-the-fly reconfiguration of the physical layer is very challenging. In order to have a reconfigurable data-path, a dataflow modeling design methodology is very promising. Dataflow modeling makes the system timing easier as the system can be implemented as synchronous sub-blocks that are connected together with asynchronous buffers. However, while dataflow modeling facilitates the dynamic reconfiguration of a system and it allows systematic design optimizations, it requires inserting buffers on the communication links between sub-blocks (actors). Since memory is expensive and limited on platforms like FPGA, buffer minimization must be considered in dataflow modeling of cognitive radio physical layer.

In order to understand the interactions and design space exploration among RF, baseband algorithms, and hardware architectures, we need to prototype our proposed cognitive radio solutions on a testbed. Implementing these solutions on a testbed allows us to perform a cross-layer analysis of the achieved spectrum and energy efficiency, as well as the design and implementation complexities. In addition, we develop advanced RF architectures to model the performance and operation of wireless terminals with realistic computational platforms. We analyze the overall complexity of the systems that are developed and synthesized based on parametrized and dynamic dataflow modeling approach. This enables making qualitative conclusions about the feasibility and economic viability of the developed concepts, system and transceiver algorithms, RF architectures and design processes for the future cognitive radio architectures.

We use system level performance evaluation tools and framework to evaluate the system level performance and related spectrum and energy efficiency. Such evaluation will also provide a systematic approach for characterizing the operational trade-offs — including trade-offs among data rate, energy consumption, circuit area (cost), and communication reliability — across the cross-layer, cognitive radio design space. Such trade-offs will provide a mapping into estimates for key implementation metrics as a function of different combinations of design parameters. Such parameters, such as sample rate, bit precision, filter configurations, and system- and transceiver-level algorithm selection, will then be adapted systematically — in response to time varying application requirements and operational constraints — through the control of parameterized dataflow models (e.g., see [8], [21]).

For hardware demonstrations and over-the-air experiments, we leverage WARP (Wireless open Access Research Platform) [22], [23]. WARP is an open platform that is designed to enable fast prototyping of physical layer algorithms. WARP v3 has a Virtex-6 FPGA as the main processing unit and two integrated radio 2.4/5GHz transceivers. In addition, WARP can be extended with more advanced RF modules through an FMC (FPGA Mezzanine Card) connection. See Figure 4 for a picture of WARP v3 and an independent radio transceiver board that can be connected to WARP via FMC. We use and extend WARP project's open-source real-time OFDM reference design to implement digital pre-distortion and filter modules to study proof of concept in RF reconfiguration for cognitive radio systems. We evaluate the adaptability of the proposed cognitive radio architectures by systematic over-the-air tests using both real-time OFDM reference design on WARP and the accelerated simulations through WARPLab [24] operation mode. The measurements and experiments are currently being designed and in progress.

(a) WARP v3 board with two integrated radio transceivers and FMC (FPGA Mezzanine Card) expansion connector.

(b) FMC radio transceiver board.

Fig. 4.   WARP v3 testbed.

IV. Summary

In this paper, we have outlined the four main focus areas of our WiFiUS collaboration: cognitive radio systems and algorithms, flexible radio architectures, design tools and hardware platforms, and realization, synthesis and performance evaluations. In particular, we have presented work-in-progress in the dataflow-based modeling and synthesis, and realization areas. This includes initial results in dataflow modeling of modules in an OFDM baseband reference design and performance speedup analysis on a GPU. Additionally, initial research in pre-distortion algorithms and the interface with the Rice WARP testbed are described. Future work will focus on the integration and adaptation of the proposed new parameterized multidimensional design hierarchy mapping methodology to RF testbeds.

Acknowledgements

References

[1] NSF, "NSF Workshop Report on Future Wireless Communication Networks," November 2-3 2009, http://www.networks.rice.edu/nsf-workshop/index.html.

[2] NSF, "NSF Workshop Report on Enhancing Access to the Radio Spectrum (EARS)," August 4-6 2010, http://wireless.vt.edu/nsf/index.html.

[3] Tekes Finnish Funding Agency, *TRIAL: Trial Environment for Cognitive Radio and Networks*, 2012, http://www.tekes.fi/programmes/trial.

[4] M. Valkama, A. Shahed, L. Anttila, and M. Renfors, "Advanced digital signal processing techniques for compensation of nonlinear distortion in wideband multicarrier radio receivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 6, pp. 2356–2366, June 2006.

[5] J. Boutellier, S. S. Bhattacharyya, and O. Silven, "A low-overhead scheduling methodology for fine-grained acceleration of signal processing systems," *Journal of Signal Processing Systems*, vol. 60, no. 3, pp. 333–343, September 2010.

[6] Janne Janhunen, Teemu Pitkanen, Olli Silvén, and Markku Juntti, "Programmable implementations of MIMO-OFDM detectors: Design, benchmarking and comparison," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 1581–1584.

[7] S. S. Bhattacharyya, E. Deprettere, R. Leupers, and J. Takala, Eds., *Handbook of Signal Processing Systems*, Springer, 2010.

[8] B. Bhattacharya and S. S. Bhattacharyya, "Parameterized dataflow modeling for DSP systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2408–2421, October 2001.

[9] B. Bhattacharya and S. S. Bhattacharyya, "Quasi-static scheduling of reconfigurable dataflow graphs for DSP systems," in *Proceedings of the International Workshop on Rapid System Prototyping*, Paris, France, June 2000, pp. 84–89.

[10] P. K. Murthy and E. A. Lee, "Multidimensional synchronous dataflow," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 2064–2079, August 2002.

[11] L. Wang, C. Shen, G. Seetharaman, K. Palaniappan, and S. S. Bhattacharyya, "Multidimensional dataflow graph modeling and mapping for efficient GPU implementation," in *Proceedings of the IEEE Workshop on Signal Processing Systems*, Québec City, Canada, October 2012, pp. 300–305.

[12] O. Edfors, M. Sandell, J.-J. van de Beek, D. Landstrom, and F. Sjoberg, "An introduction to orthogonal frequency division multiplexing," Tech. Rep., Lulea University of Technology, Sweden, September 1996.

[13] P. A. Hoeher, S. Kaiser, and P. Robertson, "Two-dimensional pilot-symbol-aided channel estimation by wiener filtering," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1845–1845.

[14] *CUDA C Best Practice Guide*, 2013, http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html, visited on March 8, 2013.

[15] S. Pagadarai, Alexander M. Wyglinski, and R. Rajbanshi, "A novel sidelobe suppression technique for OFDM-based cognitive radio transmission," in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, October 2008, pp. 1–7.

[16] Zhu Fu, L. Anttila, M. Valkama, and Alexander M. Wyglinski, "Digital pre-distortion of power amplifier impairments in spectrally agile transmissions," in *35th IEEE Sarnoff Symposium (SARNOFF)*, May 2012, pp. 1–6.

[17] Zhou Yuan, Srikanth Pagadarai, and Alexander M. Wyglinski, "Feasibility of NC-OFDM transmission in dynamic spectrum access networks," in *Proceedings of the 28th IEEE conference on Military communications*, Piscataway, NJ, USA, 2009, MILCOM'09, pp. 1078–1082, IEEE Press.

[18] Hyun-Woo Kang, Yong-Soo Cho, and Dae-Hee Youn, "On compensating nonlinear distortions of an OFDM system using an efficient adaptive predistorter," *IEEE Transactions on Communications*, vol. 47, no. 4, pp. 522–526, April 1999.

[19] L. Anttila, P. Handel, and M. Valkama, "Joint mitigation of power amplifier and I/Q modulator impairments in broadband direct-conversion transmitters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 4, pp. 730–739, April 2010.

[20] 3rd Generation Partnership Project, "Evolved Universal Terrestrial Radio Access (E-UTRA); Carrier Aggregation; Base Station (BS) radio transmission and reception," Tech. Rep. TR 36.808, V10.0.0, Technical Specification Group Radio Access Network, June 2012.

[21] C. Shen, W. L. Plishker, D. Ko, S. S. Bhattacharyya, and N. Goldsman, "Energy-driven distribution of signal processing applications across wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 6, no. 3, June 2010, Article No. 24, 32 pages.

[22] Rice University WARP Project, http://warp.rice.edu.

[23] K. Amiri, Y. Sun, P. Murphy, C. Hunter, J.R. Cavallaro, and A. Sabharwal, "WARP, a unified wireless network testbed for education and research," in *IEEE International Conference on Microelectronic Systems Education*, San Diego, CA, June 2007, pp. 53 – 54.

[24] Rice University WARPLab M-Code Framework, http://warp.rice.edu/trac/wiki/WARPLab.