Multi-Platform Demonstrations using the Iris Architecture for Cognitive Radio Network Testbeds

P. D. Sutton*, J. Lotze*, H. Lahlou*, B. Özgül*, S. A. Fahmy†, K. E. Nolan*, J. Noguera‡, and L. E. Doyle*
*University of Dublin, Trinity College, Ireland
Email: {suttonpd, jlotze, hlahlou, ozgulb, keith.nolan, linda.doyle}@tcd.ie
†School of Computer Engineering, Nanyang Technological University, Singapore. Email: sfahmy@ntu.edu.sg
‡Xilinx Research Labs, Dublin, Ireland. Email: juanjo.noguera@xilinx.com

***Abstract*—This paper presents three reconfigurable radio systems developed within CTVR, The Telecommunications Research Centre and demonstrated at the IEEE International Dynamic Spectrum Access Networks (DySPAN) symposium held in Chicago in October 2008. All three systems were developed using the Iris cognitive radio network architecture. Each system employs a different processing platform.**

Today's radio communication standards feature increasing levels of flexibility and reconfigurability as designers strive to extract as much performance as possible from the resources available. To provide the required flexibility, general processing platforms are being employed to a greater extent than ever before. At the same time, the range and scope of available processing platforms is expanding. The systems presented in this paper use three such general processing platforms; a multi-core General Purpose Processor (GPP), the Cell Broadband Engine (CellBE) and a Xilinx Field Programmable Gate Array (FPGA). The paper provides an overview of Iris and looks at each demonstration system in turn, illustrating the way in which the unique features of each platform are used. The authors present a number of insights gained in the course of developing the systems and address the role of experimentation in emerging wireless networks research.

I. Introduction

In today's radio communication standards [1], [2], there is an increasing demand for flexibility and reconfigurability to make better use of available resources and to provide autonomous system operation. In this paper, three reconfigurable radio systems developed within CTVR, The Telecommunications Research Centre and demonstrated at the IEEE International Dynamic Spectrum Access Networks (DySPAN) symposium held in Chicago in October 2008 are presented.

The first demonstration system is implemented upon a multi-core General Purpose Processor (GPP). The system employs a novel PHY-layer signalling technique to achieve network coordination using a bandwidth-adaptive OFDM-based waveform [3]. Recent years have seen the trend from single to multi-core GPPs as processor designers begin to reach the physical limits of semiconductor-based microelectronics and look for better performance / power consumption trade-offs. The development time required for GPP-based platforms is low due to the wide range of development tools available. However, in order to fully leverage multi-core GPP platforms, skill is needed to extract parallelism from the algorithms being implemented. This system is presented in Section III.

The second demonstration system employs the powerful Cell Broadband Engine (CellBE) which can be found in the Sony Playstation 3 platform [4]. This system uses the compute power of the CellBE to perform real-time cyclostationary signal analysis [5], a highly computationally complex process which would be infeasible using today's GPP-based platforms. The CellBE features a heterogeneous architecture comprising one GPP and eight RISC processors with 128-bit Single Instruction Multiple Data (SIMD) organisation for vector processing. The CellBE demonstration system is presented in Section IV.

The third demonstration system, presented in Section V, features a Dynamic Spectrum Access (DSA) network implemented upon Xilinx Virtex II Pro Field Programmable Gate Arrays (FPGAs). FPGAs provide high levels of processing power due to their massively parallel architectures. While FPGAs require greater development effort than either GPP platforms or the CellBE, improvements in development tools as well as the tools provided with the Iris framework continue to reduce this gap.

Sec. II provides an overview of the Iris cognitive radio network architecture used to develop each demonstration system. The systems are themselves presented in Sec. III, IV and V. Finally, Sec. VI provides a number of insights and concludes.

II. Iris

Iris is an architecture for cognitive radio networks designed specifically for run-time reconfiguration. It is a component-based architecture written in portable C++ for use across multiple CPU architectures and operating systems. A single Iris *component* encapsulates a particular function such as a digital filter or modulator. A network node is constructed in Iris using one or more signal processing chains, each comprising a number of components.

A key feature supporting run-time reconfiguration in Iris is the use of component *parameters*. In building an Iris compo-

This material is based upon work supported by Science Foundation Ireland under Grant No. 03/CE3/I405 as part of CTVR at University of Dublin, Trinity College, Ireland and joint work with Xilinx, supported by Enterprise Ireland under its Innovation Partnership scheme, project IP20060367.

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9264
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9264

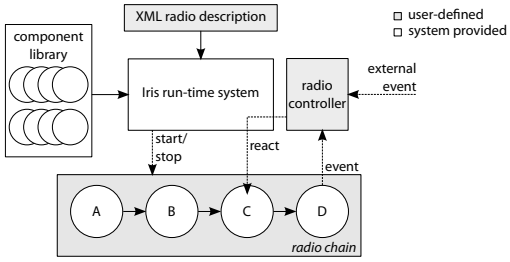

Fig. 1. The Iris architecture.

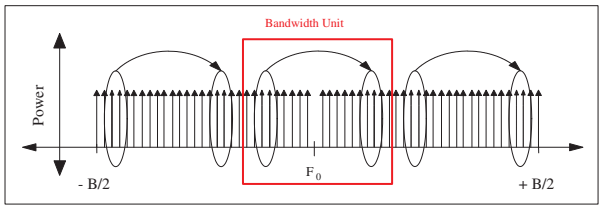

Fig. 2. Waveform Generation

nent, the radio designer can choose to expose a number of variables which are externally controllable. These parameters permit the operation of the component to be altered while the radio is running. Parameter examples include the roll-off of a digital raised cosine filter or the order of a Quadrature Amplitude Modulation (QAM) modulator.

Within a cognitive radio network, node reconfiguration typically occurs as a result of observing the operating environment and the radio itself. In Iris, particular components can be responsible for such observation. Examples include a spectrum analyser or energy detection component. For a node to respond to observations, a mechanism for responding to observations is required. In Iris, component *events* serve as internal triggers, prompting reconfiguration of the node. Events are specified in a component at design-time and may be triggered while the radio runs. For example, an energy detection component may trigger a specific event if the power of the received signal exceeds a predefined threshold.

For a network node to successfully adapt to changes in its operating environment, component events must result in appropriate reconfiguration of component parameters. The element of the Iris architecture with responsibility for this process is the *controller*. A controller listens for particular events from running components and may carry out reconfiguration of the node if required. Controllers can consist of simple predefined reconfigurations in response to events or may carry out more complex analysis of the node, listening for multiple events, using models of node functionality to decide upon appropriate courses of action and learning from past experience. Fig. 1 illustrates the Iris architecture.

A powerful aspect of the Iris architecture is the ability to support heterogeneous processing platforms. This capability permits the radio designer to leverage powerful platforms such as FPGAs, DSPs and the CellBE. The use of a component-based architecture permits code executing upon these platforms to be encapsulated in a single entity with a generic interface for data-passing, reconfiguration and life-cycle management. All parameters and data inputs and outputs of the software wrapper component are routed to the accelerator hardware within this component, encapsulating all low-level tasks. The software component wrapper appears like a normal software component to the Iris system, abstracting away the detail of managing computations on the dedicated hardware or processors.

III. GPP: Bandwidth-Adaptive Waveforms

This demonstration uses Iris on a multi-core GPP platform to implement a bandwidth-adaptive waveform for DSA networks. A novel PHY-layer signalling approach is employed to permit blind bandwidth estimation at the receiver. In this way, the need for a dedicated control channel can be overcome.

The platform used in the demonstration is an Intel dual-core GPP. The RF front-end used is the Universal Software Radio Peripheral (USRP) with an RFX2400 daughterboard [6]. The USRP connects via USB and supports a bandwidth of up to 8 MHz using 16 bit samples.

A. Coordination

One of the key challenges associated with the development of highly reconfigurable radio networks is network coordination. A transmitting node within a wireless network may be capable of adapting its waveform to suit the operating conditions but this optimisation is futile if receiving nodes cannot simultaneously adapt in order to receive and demodulate that waveform.

It has been demonstrated that *cyclostationary signatures* provide a robust and autonomous mechanism to support waveform adaptation in a DSA network [7], without the need for dedicated control channels. In basic terms, a cyclostationary signature is generated by creating an artificial correlation pattern in the spectrum of a signal. In order to detect this spectral correlation, a receiver only needs to know the distance between the correlated portions of the signal spectrum or the cyclic frequency. Using an Orthogonal Frequency Division Multiplex (OFDM) based waveform, cyclostationary signatures can be generated simply by mapping one or more subcarriers onto another.

This demonstration illustrates their use in a bandwidth-adaptive waveform specifically designed for DSA.

B. A Bandwidth-Adaptive Waveform

Fig. 2 illustrates the technique used to generate a bandwidth-adaptive waveform with embedded signatures. An OFDM based waveform is employed which comprises between 1 and 16 bandwidth units, each containing 64 subcarriers. Within each unit, a signature is embedded by mapping a subset of 4 subcarriers. By employing a constant mapping separation in each bandwidth unit, each signature manifests at the same cyclic frequency and a low-complexity single-cycle signature detector can be used at the receiver to estimate the overall

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9264
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9264

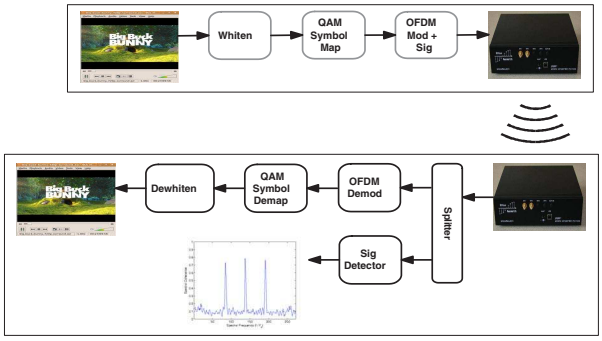

Fig. 3. Demonstration Architecture

signal bandwidth. In our demonstration, each bandwidth unit comprises 56 data-carrying subcarriers, 2 pilots, 2 guard subcarriers (including DC) and 4 mapped subcarriers. Thus it can be seen that the overhead associated with the use of embedded signatures is $\frac{1}{16}$.

For the purposes of the demonstration, a maximum bandwidth of 2 MHz was used. For our waveform design, this gave us a signal bandwidth resolution of 125 kHz.

The architectures of the demonstration transmitter and receiver are illustrated in Fig. 3. At the transmitter, a video stream is encoding using the VideoLAN VLC Player. The stream is whitened, mapped onto a QAM constellation and OFDM modulated before being sent to the USRP for transmission. At the OFDM modulation component, the number of bandwidth units employed is chosen and subcarrier mapping is used to embed cyclostationary signatures. In order to illustrate the use of signatures to enable dynamic bandwidth estimation, the number of bandwidth units generated by the transmitter is dynamically changed every 5 seconds. At each change, the number of units to be used is randomly chosen.

At the receiver, the USRP converts the signal to baseband, applies the analog-to-digital conversion and provides the signal samples to a signal splitter component. The splitter provides the same data stream on two output ports to two separate processing chains. The first chain consists of a signature detector which uses a single-cycle detector to estimate the number of signatures in the received waveform. The output of the detector is displayed graphically to permit its operation to be observed. The second processing chain consists of an OFDM demodulator, a QAM symbol demapper and a data dewhitener, after which the data stream is provided to VLC for display. Upon detection of a change in signal bandwidth, the signature detector triggers a reconfiguration of the OFDM demodulator. In this way, the receiver dynamically reconfigures to demodulate an uninterrupted stream of video data.

IV. CellBE: Real-Time Cyclostationary Analyzer

Cyclostationary feature analysis is a powerful signal processing technique that can be used for signal detection, classification, parametrisation, synchronisation and equalisation [5]. Many of the communications signals in use today may be modelled as cyclostationary signals due to the presence of one or more underlying periodicities which arise due to the coupling of stationary message signals with periodic sinusoidal carriers, pulse trains or repeating codes. These underlying periodicities give rise to correlation patterns within the spectrum of the signal.

One approach for analysing such patterns involves estimation of the Spectrum Correlation Function (SCF) [8]:

$$\hat{S}_\alpha[k] = \frac{1}{L}\sum_{l=0}^{L-1} X_l[k]X_l^*[k-\alpha]W[k], \qquad (1)$$

where $k = 0, 1, \ldots, N-1$, α is the cyclic frequency, $W[k]$ is a smoothing spectral window, $X_l[k]$ is the kth discrete Fourier transform coefficient after applying N-point Fast Fourier Transfrom (FFT) to the lth received signal window, $\{x_{(l-1)N}, x_{(l-1)N+1}, \ldots, x_{lN-1}\}$, and x_n is the nth received signal sample.

A key drawback associated with the use of cyclostationary signal analysis is the computational complexity associated with the calculation of the SCF over a wide range of α. This complexity arises due to the large numbers of complex convolution operations required. The computing power required to perform real-time analysis for a signal of appreciable bandwidth makes it infeasible for a standard GPP. However, the CellBE, capable of 200+ GFLOPS of single precision floating point performance, can provide the high performance computing required for such a signal processing technique while permitting the type of flexibility typically associated with a GPP for similar power consumption [4].

A. CellBE Overview

The CellBE contains one PowerPC Processor Element (PPE) and eight Synergistic Processor Element (SPE), all connected to each other and to external devices by a high bandwidth memory-coherent bus, the Element Interconnect Bus (EIB). The PPE is a 64-bit PowerPC while the SPEs are Reduced Instruction Set Computer (RISC) processors, highly optimised for SIMD instructions.

B. Demonstration System

Fig. 4 illustrates the architecture of the demonstration system. The real-time cyclostationary analyser is implemented using Iris, running upon the CellBE on a Sony Playstation 3 for the SCF estimation and on an Intel GPP for the graphical display using a Matlab interface. Both platforms are linked using Gigabit Ethernet.

Two levels of parallelism are exploited for the SCF estimation. Multiple SPEs operate in parallel where each one computes the SCF for a given data subset. The second level of parallelism is provided within each SPE where vector operations are used for further computational efficiency. Averaging is carried out on the PPE.

In this set up signal analysis can be performed at rates of up to 16 MSamples/sec. While the SCF computation alone can handle much higher rates, we are limited by the maximum receive bandwidth of our RF front-end.

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9264
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9264

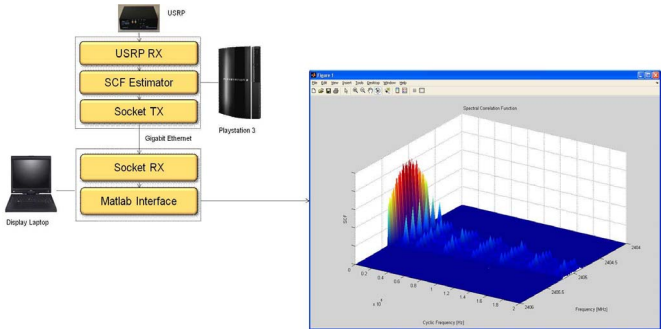

Fig. 4. Demonstration Architecture

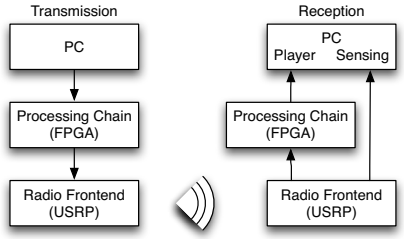

Fig. 5. Demonstration Setup.

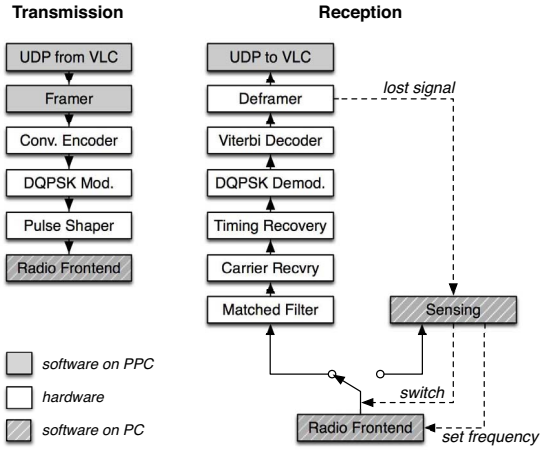

Fig. 6. Transmitter and receiver radio chains.

V. FPGA: Frequency Rendezvous

This system demonstrates frequency rendezvous for DSA using energy detection. Iris is executed on an FPGA-based platform, in this demonstrator hosting a Xilinx Virtex II-pro, although recently the framework has been moved to the more powerful Virtex 5. In a DSA network, the issue of frequency rendezvous arises as a transmitter and receiver must identify a common frequency band in order to establish a communication link. While a dedicated control channel can be used to achieve rendezvous, this represents a single point of failure and a bottleneck. Furthermore, the frequency of this control channel has to be known *a priori*. A better and more flexible solution is to establish rendezvous without a control channel.

This demonstrator shows rendezvous without control channel. The receiver uses spectrum sensing techniques to scan for the transmitter, changes its frequency accordingly, and establishes the link. We use a coded Differential Quadrature Phase Shift Keying (DQPSK) link to stream real-time video data. The transmit frequency can be changed manually. When the receiver loses the connection, it searches the spectrum for the transmitter. Once found, it establishes a link, and continues to play the video. All baseband signal processing for both the transmitter and receiver is running on the FPGA's logic fabric.

A. Sensing Algorithm

Energy detection (ED) is an established method in spectrum sensing [9] which can be employed by a receiving node to estimate the unknown frequency of a transmitter. *Cyclic-feature detection (CD)* [10] is another popular technique that can be applied since most signals contain inherent cyclostationary features (see Sec. IV). ED is more generic as it does not require any signal-specific information, while CD is more robust at low signal-to-noise ratios.

The sensing algorithm in this demonstration is based on ED. However, the receiver can optionally use CD if the so-called *cyclic frequency* for the signal of interest is provided. The algorithm works as follows:

1) Apply the FFT to the received sampled signal over a certain observation window.
2) Calculate an average power spectral density (PSD) function for ED (and an average SCF for CD).
3) Moving average filtering of the PSD (and SCF for CD).
4) Use a decision threshold to detect the signal and estimate the carrier frequency by finding the centre frequency of PSD (and SCF for CD). The threshold value is determined as in [11].

B. Demonstration Setup

For this demonstration, Iris is running under Linux on the embedded PowerPC of the Virtex II-pro FPGA, and the hardware components execute in the logic fabric. We employ the FPGA architecture as presented in [12]. The transmit chain receives UDP packets, through ethernet, containing video data, streamed from a PC using the VideoLAN VLC Player. This is processed by the transmission chain on the FPGA, then transmitted by the USRP radio frontend. At the receiver, the signal is processed by the reception chain in the FPGA, then passed, through UDP, to a VLC Player on a PC which plays the video. An overview is shown in Fig. 5.

Fig. 6 shows the radio chains for this demo. Components implemented in hardware are shown in white. The *DQPSK modulator* and *DQPK demodulator* map information dibits (2-bit units) into phase changes on an I/Q plot, and vice versa, respectively. A root raised-cosine filter is used at both the transmitter (*Pulse Shaper*) and receiver (*Matched Filter*) to reduce spectral sidelobes. When a modulated signal is received, the frequency of the local oscillator is typically off by some margin which is corrected by *Carrier Recovery*. *Timing*

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9264
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9264

Recovery finds the correct sampling points for each symbol, eliminating the effects of timing shifts. We use the standard Xilinx *Convolutional Coder* and *Viterbi Decoder* cores provided as part of CoreGen. The *Deframer* correlates a fixed 64-bit preamble (inserted by the *Framer* at the transmitter), with the streaming data, in order to identify the start of a valid frame of data. The *Sensing* component is implemented in software on a PC. It uses the algorithm described in Sec. V-A to locate the transmitter frequency, then enables the FPGA-based processing chain for reception.

VI. Insights Gained

In developing the systems described in this paper, a number of key insights were gained in terms of the role of general purpose processing platforms for emerging wireless networks and, more generally, in terms of the importance of experimentation in wireless systems research.

A. The Role of General Purpose Processing Platforms

As emerging wireless systems feature more flexibility and reconfigurability, general purpose processing platforms are playing an increasingly important role in these systems. At the same time, the availability of powerful development tools for these platforms is reducing the time required for new system design and development. Heterogeneous multi-core platforms can provide the flexibility of GPP based platforms while delivering the necessary processing power at lower levels of cost and power consumption. As the development tools for such platforms improve, they will find more widespread use in emerging wireless system designs.

B. The Iris Architecture

The range of demonstration systems discussed in this paper serves to illustrate the power of a highly reconfigurable architecture such as Iris. Built specifically to support run-time reconfiguration of wireless systems, Iris allows highly flexible systems to be rapidly prototyped and new concepts to be proven. In developing the demonstration systems described, a number of improvements were made to the original Iris architecture. One such improvement is the introduction of Iris engines which encapsulate a specific domain for the execution of one or more signal processing components. Different engines provide different levels of flexibility and performance. Through the use of multiple engine types, the radio designer can choose the appropriate execution semantics for each part of a node within a wireless network. Iris engines also provide improved support for parallel execution within a network node and can be used as a software wrapper to support heterogeneous processing platforms.

C. The Importance of Experimentation

In the area of wireless systems research, especially the current hot topics of cognitive radio networks and DSA, experimentation is essential for verification of new concepts. Many of the techniques proposed to address challenges such as low-power signal detection, inter-system interference and network coordination can only be truly tested under real-world conditions. In the case of disagreement over what emerging systems are capable of, experimental results provide concrete evidence. This evidence can then be used as the basis for informing new approaches to regulation and management when necessary.

Carrying out wireless systems experimentation is not straightforward. Considerable effort is required to establish an experimental platform and to build competencies in each of the diverse domains involved. In order to carry out experimental research and get credit for this work, a great deal of background work is first needed. However, an increasing number of international conferences are acknowledging the importance of experimental research through special demonstration sessions. Significant examples of this are the demonstration sessions held as part of DySPAN 2007 and 2008. At the same time, open hardware and software platforms such as the USRP and GNU Radio are lowering the entry barriers for new experimental researchers. In an effort to further promote experimental research, the authors hope to make the Iris architecture openly available for use in the near future.

References

[1] C. R. Stevenson, G. Chouinard, Z. Lei, W. Hu, S. J. Shellhammer, and W. Caldwell, "IEEE 802.22: The first cognitive radio wireless regional area network standard," *IEEE Commun. Mag.*, vol. 47, pp. 130–138, May 2009.

[2] Motorola, Inc. (2007) *The Drivers to LTE*. Solution Paper. [Online]. Available: http://business.motorola.com/experiencelte/pdf/TheDriverstoLTESolutionPaper.pdf

[3] P. D. Sutton, B. Özgül, K. E. Nolan, and L. E. Doyle, "Bandwidth-adaptive waveforms for dynamic spectrum access networks," in *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN).*, Chicago, IL, Oct. 2008, pp. 1–7.

[4] D. C. Pham, T. Aipperspach, D. Boerstler, M. Bolliger, R. Chaudhry, D. Cox, P. Harvey, P. M. Harvey, H. P. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Pham, J. Pille, S. Posluszny, M. Riley, D. L. Stasiak, M. Suzuoki, O. Takahashi, J. Warnock, S. Weitzel, D. Wendel, and K. Yazawa, "Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 179–196, 2006.

[5] W. A. Gardner, *Cyclostationarity in Communications and Signal Processing*. New Jersey, NY, USA: IEEE Press, 1993.

[6] *Universal Software Radio Peripheral – The Foundation for Complete Software Radio Systems*, Ettus Research LLC, Mountain View, California, USA, Nov. 2006. [Online]. Available: http://www.ettus.com/downloads/usrp_v4.pdf

[7] P. D. Sutton, K. E. Nolan, and L. E. Doyle, "Cyclostationary signatures in practical cognitive radio applications," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 13–24, 2008.

[8] W. Gardner, *Introduction to Random Processes with Applications to Signals and Systems*, W. Gardner, Ed. MacMillan, 1985.

[9] A. Ghasemi and S. E. Sousa, "Spectrum sensing in cognitive radio networks: Requirements, challenges, and design trade-offs," *IEEE Commun. Mag.*, vol. 46, pp. 32–39, Apr. 2008.

[10] W. A. Gardner, "Signal interception: A unifying theoretical framework for feature detection," *IEEE Trans. Commun.*, vol. 36, pp. 897–906, Aug. 1988.

[11] T. J. O'Shea, T. C. Clancy, and H. J. Ebeid, "Practical signal detection and classification in gnuradio," in *SDR Forum Technical Conference (SDR)*, Denver, Colorado, Nov. 2007.

[12] S. A. Fahmy, J. Lotze, J. Noguera, L. E. Doyle, and R. Esser, "Generic software framework for adaptive applications on FPGAs," in *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, USA, Apr. 2009, pp. 55–62.

Digital Object Identifier: 10.4108/ICST.CROWNCOM2010.9264
http://dx.doi.org/10.4108/ICST.CROWNCOM2010.9264