A SOFTWARE-DEFINED RADIO PROTOTYPING PLATFORM FOR COGNITIVE RADIO APPLICATIONS

Christian Kocks, Alexander Viessmann, Andrey Skrebtsov, Guido H. Bruck, Peter Jung

Department of Communication Technologies
University of Duisburg-Essen
Duisburg, Germany
info@kommunikationstechnik.org

ABSTRACT

During the last decades, the demand for higher data rates in wireless communication systems grew steadily. While the number of communication systems increased significantly, the amount of available spectral resources has remained constant. To exploit spectral resources more efficiently, cognitive radio concepts have become of utmost interest. The goal of cognitive radio systems is to sense the system's immediate environment and react upon its findings. Thereby, non-occupied frequency bands can be identified and used for data transmission. To allow the adaptation of a communication system to the outcomes of the sensing operation, a software-defined radio architecture is advantageous. Systems based on a software-defined radio architecture enable the reconfiguration of software defining the physical as well as the media access control layer. Within this manuscript, a software-defined radio prototyping platform developed by the authors is presented. It is ideally suited for rapid prototyping of wireless communication transceivers with cognitive radio functionality. A realization concept for a cognitive radio demonstrator is given for a TV white space application where non-occupied frequency ranges within the UHF band primarily used for TV broadcasting are sensed. Dynamically, the non-occupied frequency bands are then accessed for data transmission based on an IEEE 802.11 system.

1. INTRODUCTION

The recent years revealed an increasing demand for wireless communication systems such as wireless LAN (IEEE 802.11), WiMAX (IEEE 802.16), 3GPP UMTS and LTE and various digital broadcasting standards. To keep up with the demands for steadily higher data rates as well as for communication devices providing multi-standard capabilities, the hardware manufacturers are required to accelerate their development processes. Numerous research activities in the last two decades showed that software-defined radios help to overcome this problem [1], [2].

The idea of software-defined radio reaches back to the 1980s where reconfigurable radios were developed for radio intelligence in the short-wave range [1]. However, with publications such as the special issues on software radios in the IEEE Communication Magazine [3], software radio became popular to many radio developers. A transceiver can be considered as a software radio if its communication functions are realized as programs running on a suitable processor [1]. Ideally, a software radio directly samples the antenna output. Due to hardware limitations, e.g. concerning analog-to-digital conversion, this approach is not feasible. As a result, the idea of software-defined radio was introduced which is a practical realization of a software radio [1]. The received signal is filtered and down-converted to an intermediate frequency (IF) or directly to base-

band where it is converted to the digital domain. With the availability of software-defined radio based transceivers, the architecture can be used to implement multiple communication systems running on a common hardware platform by a simple reconfiguration process. Core part of such reconfigurable software-defined radio transceivers is a digital processor running the software. This software can be divided into a signal processing part and a scheduling part [4]. The deployment of digital signal processors pave the way towards rapid prototyping capabilities of the platform since signal processing algorithms can be implemented very easily using high-level programming languages such as C/C++. This allows rapid algorithm benchmarking already during the standardization process of a future communication system. Nevertheless, solely relying on the computational power of a digital processor is insufficient for modern communication systems. Since one important aspect for the design of new communication systems is a higher spectral efficiency, in particular the complexity of forward error correction schemes increases, leading to a steady growth of the signal processing requirements. Hence, even the availability of multi-core digital signal processors does not allow real-time implementations of modern communication systems. To circumvent this limitation, hardware-based implementations of certain digital signal processing blocks are favorable. Therefore, field-programmable gate arrays (FPGA) are suitable devices which allow the integration of highly parallelized algorithms on the one hand and reconfigurability on the other hand. However, the implementation efforts are typically significantly higher than for digital processors. To allow an efficient interoperation of digital signal processors and FPGAs, high-speed interfacing must be guaranteed. A platform taking into account the aforementioned aspects can be considered to fill the gap between the theoretical considerations concerning algorithm design and the final system implementation.

Reconfigurability is a crucial prerequisite for cognitive radio applications. A cognitive radio is a software-defined radio that additionally senses its environment and tracks changes [5]. By reacting upon its findings, it can adapt itself to the underlying conditions. Cognitive radios based on software-defined radio can be configured via control units. Such control units need information about the type and standard of the radio communications link and software modules for the signal processing path in order to reconfigure the transceiver properly. Due to the scarcity of spectral resources resulting from the increasing demand for wireless communication systems, cognitive radio is presently of utmost interest [6] and will help to exploit the frequency spectrum more efficiently in the future.

An application for cognitive radio is the reuse of unused frequency bands in the TV spectrum, referred to as TV white space. In this context, one distinguishes between primary and secondary communication systems. The primary communication system is the TV broadcast signal which has highest priority for transmission. However, in case certain frequency bands are not used by the primary system, they can be reused by a secondary system provided that the corresponding regulation authority permits such techniques.

Within this manuscript, the authors present a prototyping platform based on the software-defined radio paradigm. The platform concept is illustrated as well as a cognitive radio realization concept is given.

The document is organized as follows. After this introduction, Sect. 2 gives an overview of the prototyping platform. Since clock generation and distribution is a critical factor in platform design, Sect. 3 primarily addresses this aspect. Modularity and scalability aspects were a crucial design constraint and are discussed in Sect. 4. Sect. 5 presents a realization concept of a cognitive radio demonstrator based on concurrent processes running on the platform. In Sect. 6, selected results of the sensing engine are shown. Finally, a conclusion is given in Sect. 7.

2. PLATFORM OVERVIEW

Following the ideas highlighted in Sect. 1, the prototyping platform developed by the authors of this manuscript is based on a combination of a digital signal processor (DSP) and an FPGA. A block di-

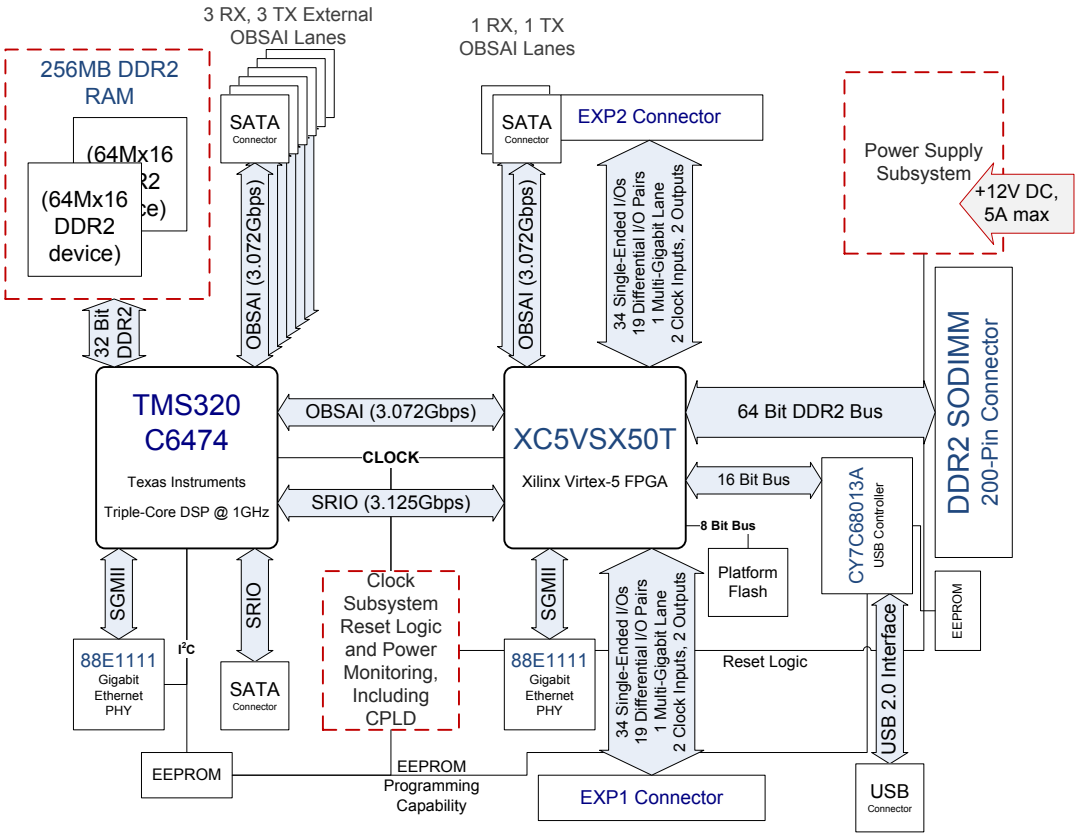

Figure 1: *eFalcon* block diagram

agram of this platform, in the following referred to as *eFalcon*, is shown in Figure 1. The deployed DSP is a Texas Instruments triple-core DSP TMS320C6474 running at 1 GHz system clock. The DSP has 256 MB of DDR2 Random Access Memory (RAM) attached which enables the implementation of memory-intensive applications. The interfaces to external components consist of a Gigabit Ethernet link, three full duplex OBSAI (Open Base Station Architecture Initiative) or CPRI (Common Public Radio Interface) lanes, commonly referred to as antenna interface, and a Serial RapidIO (SRIO) link. The clock generation and distribution network described in Sect. 3 provides a reference clock which is used to generate the system clock by a processor-internal phase-locked loop (PLL). The DSP supports several boot modes using either I^2C, Ethernet or SRIO. It contains very flexible hardware accelerators for Viterbi and Turbo decoding since convolutional and turbo codes can be found in a large number of communication standards. In case of channel codes like LDPC (Low-Density Parity-Check) codes which are employed e.g. in various modern video broadcasting standards [7],[8], the DSP's signal processing capability is not sufficient to decode the incoming signal stream in real-time. In that case, the FPGA which is directly attached to the DSP can be used to design a DSP co-processor dedicated to a specific signal processing task. In addition to the potential DSP co-processing capability, the FPGA, namely a Xilinx Virtex-5 SX50T, implements additional interfaces to external peripheral devices. Besides Giga-

bit Ethernet, a USB 2.0 interface is attached to the FPGA which can be used to communicate with conventional PC hardware. The connection to the DSP is realized using OBSAI, CPRI or SRIO. The FPGA offers the capability of attaching a DDR2 SODIMM which can be used to extend the memory capability of the platform significantly. Spare high-speed serial ports are routed to proprietary connectors which can be used with a variety of protocols.

One major task of the FPGA is to act as a bridge between the high-speed serial interfaces of the DSP and the parallel interfaces of analog-to-digital (A/D) or digital-to-analog (D/A) converters. Therefore, the FPGA offers a full implementation of the Avnet EXP Expansion Connector Specification [9] consisting of two connectors with 34 single-ended I/O ports and 19 differential I/O pairs each. In addition, this interface consists of lines for clock inputs and outputs as well as high-speed serial connectivity. The FPGA can be used to post-process baseband data calculated by the DSP and submit it to the D/A conversion process. Digital up-conversion, upsampling and filtering algorithms are applied in this step. In the A/D conversion process, the FPGA preprocesses the incoming data before transmitting it to the DSP. Signal statistics, which act as an input for an automatic gain control (AGC), are calculated. These statistics consist of long-term averages as well as fast detect features to enable over-range indication and can also be used to remove DC offsets. Further processing blocks are digital down-conversion in case of IF sampling, sampling rate adaptation and filtering. The FPGA is also ideally suited for the application of synchronization blocks which usually consist of autocorrelation or cross-correlation based algorithms, cf.[10].

Due to the large number of different devices, a complex power supply design is mandatory. *eFalcon* works with an input voltage of 12 V. Switching regulators are used to generate voltages less than this input voltage at a high efficiency and directly supply the multitude of digital devices. Voltages which are used to supply analog parts of devices, especially the PLLs (phase-locked loops), are generated using linear low-dropout regulators. All voltages are monitored and reported to a central glue logic which is implemented in a CPLD (Complex Programmable Logic Device). The CPLD is responsible for the power-on procedure of the *eFalcon* platform.

3. CLOCK GENERATION AND DISTRIBUTION

Modern digital systems have strict requirements on the clock references. Both the DSP and the FPGA contain on-chip PLLs which need to be supplied with a suitable reference frequency. These signals are required to derive the core clocks as well as the reference clocks for the high-speed serial links. A general block diagram of the platform clock subsystem is depicted in Figure 2. Most of the reference clocks in the system are generated by Texas Instruments CDCE62005 devices which are fully integrated PLLs including a voltage controlled oscillator (VCO). A programmable loop filter is integrated on the chip as well. All platform clocks generated by the CDCE62005 devices can be split into two general classes. The first class contains the 30.72 MHz derivatives which are used in the OBSAI and CPRI clock domain. The second clock class includes all other clocks which are derivatives of 25 MHz. The reference clock of 125 MHz which is necessary for SRIO is generated by a dedicated crystal oscillator. Both Ethernet PHY chips contain an internal PLL and a 25 MHz crystal resonator as frequency reference. The USB controller uses a 24 MHz external crystal resonator to generate the necessary clock for the FPGA-to-USB data transfer. Besides an internal generation of the PLL reference frequency, all devices have the capability to be fed by an external reference. These external references are routed to coaxial connectors which can be used by an external clock distribution network to synchronize an arbitrary number of platforms. This is an important feature in case that multiple platforms are connected to work on a single signal processing task.

4. MODULARITY AND SCALABILITY

eFalcon is based on a flexible architecture which paves the way towards a rapid implementation of

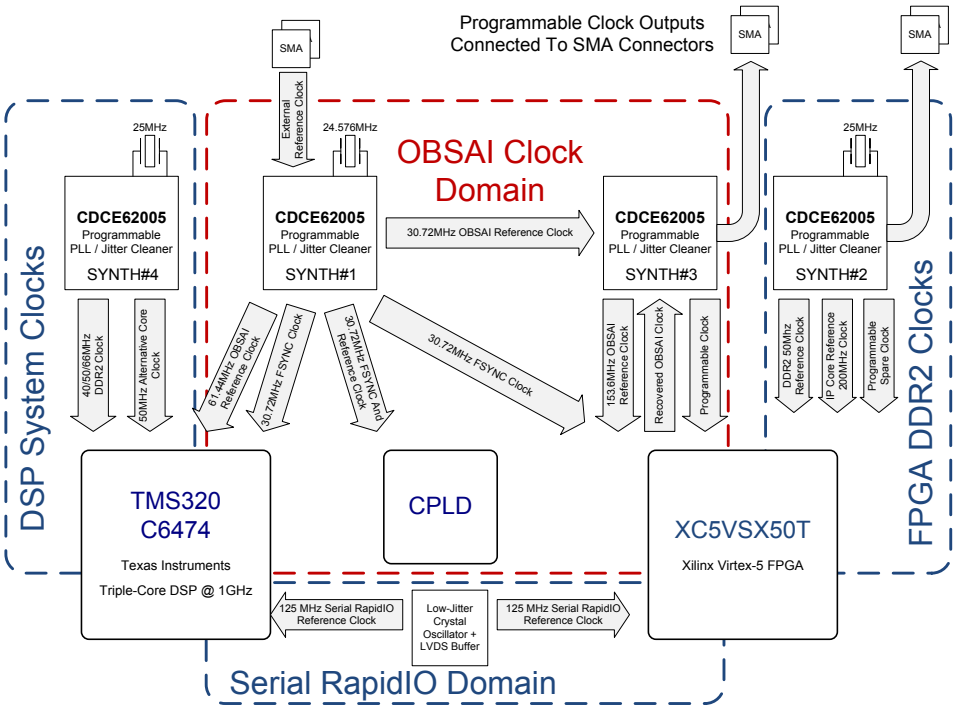

Figure 2: *eFalcon* clock subsystem

algorithms deployed in modern communication systems. It offers the opportunity to directly attach two daughter cards following the Avnet EXP Expansion Connector Specification as well as a multitude of expansion cards using the high-speed serial links.

Crucial aspects during the concept phase of the *eFalcon* platform were modularity and scalability in order to be prepared not only for current communication systems but also for next generation systems with increased complexity requirements. On the one hand, *eFalcon* consists of multiple serial high-speed interfaces such as OBSAI, CPRI and SRIO to allow intra-board communication but also board-to-board communication with data rates beyond 1 GBit/s. Arbitrary daughter cards can be attached providing interfaces to analog front-ends. To allow the deployment of further hardware accelerators, additional FPGA daughter cards can be attached as well.

On the other hand, the elaborate clock distribution concept of this platform allows an accurate synchronization of multiple platforms. Thus, the interconnection of two or more prototyping platforms by a combination of serial high-speed interfaces and a clock synchronization allows even the implementation of multi-antenna systems. In that case, one platform acts as a master device which is responsible for the clock generation and distribution to the attached platforms acting as slaves as well as for the main scheduling parts of the signal processing chain.

5. COGNITIVE RADIO REALIZATION CONCEPT

The prototyping platform presented before is ideally suited for cognitive radio applications. Therefore, a realization concept for TV white space applications is illustrated. The primary communication system considered here is the Chinese mobile television and

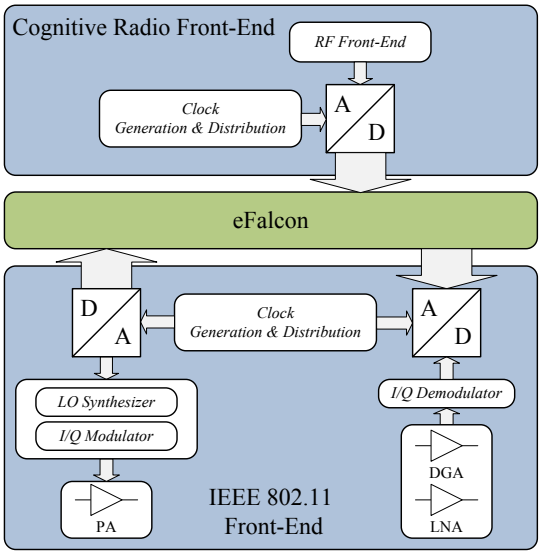

Figure 3: *eFalcon* cognitive radio realization concept

multimedia standard (CMMB) [11] while a modified IEEE 802.11 system is employed as secondary system. The IEEE 802.11 is modified in a way that it operates in the TV band with variable bandwidths.

Before describing the operation concept of this realization, the hardware setup is addressed. As depicted in Figure 3, the prototyping platform is extended by two modules. The first module, denoted as cognitive radio front-end, is intended for the cognitive part of the demonstrator while the secondary front-end is intended for the realization of the IEEE 802.11 system. The cognitive radio front-end consists of an RF tuner which receives a primary signal and down-converts it to an intermediate frequency. This IF signal is then sampled and - after further filtering operations - transferred to the *eFalcon* platform.

For the secondary system, a full transceiver functionality is necessary. The baseband signal in either complex inphase and quadrature (I/Q) or real valued intermediate frequency representation is generated using the DSP/FPGA combination of the platform and is fed into a D/A converter using the expansion interface. After a suitable reconstruction filter, the resulting analog signal is up-converted to a carrier frequency using an I/Q modulator. The local oscillator (LO) signal which is used in the up-conversion process is generated by an LO synthesizer and ranges from 300-4800 MHz. The synthesizer accepts external reference signals which can be used to synchronize the LO phase in case of multi-antenna applications. In addition, the LO signal can be externally fed in or distributed to other platforms. The I/Q up-converter is followed by an optional power amplifier branch which can be used to obtain signal levels up to +30 dBm. Optionally, the up-converter can also be driven by an IF signal.

The receiver branch consists of a gain stage implemented by a combination of a low-noise amplifier (LNA) and multiple digitally controlled amplifiers (DGA) allowing the amplification of signals at very low input levels. The gain vectors which lead to an optimal behaviour in terms of noise figure and linearity are calculated in the *eFalcon* platform according to the signal level detectors implemented at several positions within the amplification chain. The resulting signal is fed into an I/Q demodulator which converts the signal to complex baseband using an external LO signal. The down-converter can also be configured for IF applications where the I/Q branches can then be used for the implementation of a Hartley image reject receiver [12]. The signal is sampled and the resulting signal samples are transferred to the *eFalcon* platform using the expansion interface. The *eFalcon* platform is responsible for processing the physical as well as the media access control layer. While synchronization-related processes are carried out on the FPGA, the majority of the signal processing tasks run on the DSP.

The overall concept of the implementation of a cognitive radio based transceiver fundamentally relies on concurrent processes running on the platform.

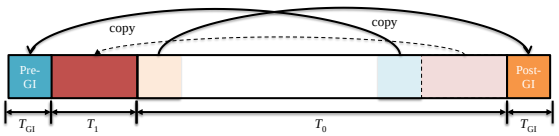

Figure 4: CMMB data symbol structure

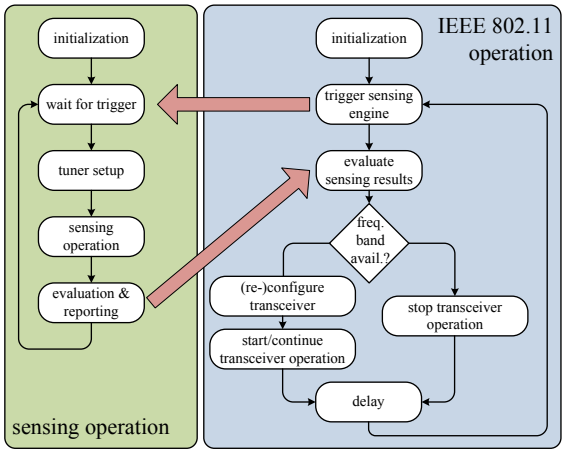

Figure 5: Cognitive radio flow diagram

The deployment of a multi-core DSP in combination with an FPGA facilitates to run the cognitive operation in parallel to the secondary transceiver system. Figure 5 illustrates the two processes running concurrently.

The first process is responsible for the cognitive operation. During the initialization state, the cognitive radio front-end is configured. The cognitive operation is based on spectrum sensing engines which exploit certain characteristics of the known primary system. CMMB, acting as primary system, uses OFDM (Orthogonal Frequency Division Multiplexing) with a data symbol structure given in Figure 4 [11].The spectrum sensing process is based on an autocorrelation metric exploiting the cyclic repetition of the data symbols. After initialization, the cognitive radio front-end waits for a trigger event to start the sensing operation. When the trigger event occurs, the RF front-end is set up according to the desired configuration and its output samples are the fed to the sensing engine where the sensing operation is carried out. The sensing result is evaluated and acts as input to the second process embodying the functionality of the IEEE 802.11 system.

The state machine describing the secondary system's process begins with an initialization state as well for initial configuration of the various hardware modules. Subsequently, the sensing operation is triggered to obtain information about available frequency bands. Based on the evaluation of the sensing results, the IEEE 802.11 front-end is configured according to the available frequency bands. In case no vacant frequency bands are available, the transceiver operation must be interrupted. In case previously available frequency become occupied during the transceiver operation, the transceiver front-end must be reprogrammed to use another carrier frequency. In order to obtain reliable information about the presence of the primary system, the transceiver operation must be paused during data acquisition of the cognitive radio front-end. The sensing operation itself is triggered periodically in a given interval. To guarantee full awareness of the transceiver's environment, not only the currently used frequency band can be scanned but the full frequency spectrum which may be reused by a secondary system.

6. SPECTRUM SENSING RESULTS

In this section, sensing results of the cognitive radio engine are shown based on the autocorrelation metric mentioned above. Before the algorithm was implemented in hardware, software simulations were carried out. These software simulations consider an single-path propagation with AWGN. The software simulations consist of two stages. Initially, the thresholds the correlator output is compared with are determined for a target false-alarm probability P_{fa}. Then, the detection probability P_{d} is simulated versus the signal-to-noise ratio using the previously determined thresholds.

In Figure 6, the simulation results as well as measurement results are depicted. Therefore, the probability of detection is plotted versus the signal-to-noise ratio. The differences between the simulation and the measurement results of 2 to 4 dB mainly stem from impairments in the analog part of the receiver. Nevertheless, in case of a false-alarm probability of 0.1, the autocorrelation-based sensing algorithm allows a detection probability of 0.9 for a received signal power of approximately -111 dBm.

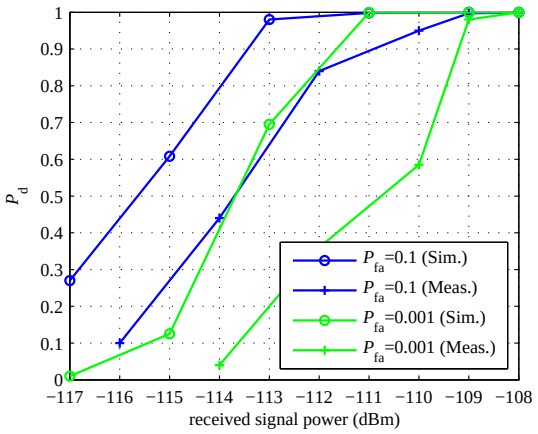

Figure 6: CMMB sensing results

7. CONCLUSION

In this manuscript, the authors presented a software-defined radio prototyping platform for cognitive radio applications. Besides powerful digital signal processing capabilities, the platform provides a high degree of modularity and scalability. The underlying architecture is based on a combination of a triple-core digital signal processor and a field-programmable gate array. The advantages of both digital signal processors and programmable logic devices are merged to a single platform. It paves the way towards rapid implementations of arbitrary communication systems with cognitive radio functionalities filling the gap between theoretical considerations concerning algorithm design and the final system implementation

Exemplarily, a cognitive radio realization concept was presented exploiting the TV white space by a modified IEEE 802.11 system and selected spectrum sensing results were given.

ACKNOWLEDGMENT

The authors would like to thank Chen Lei and Qiu Jing of Huawei Technologies Co. Ltd. for the fruitful collaboration.

References

[1] F. K. Jondral, "Software-Defined Radio: Basics and Evolution to Cognitive Radio," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 3, pp. 275–283, 2005.

[2] FP6 - Integrated Project, "E2R - End-to-End Reconfigurability."

[3] J. Mitola, "The Software Radio Architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26–38, 1995.

[4] C. Kocks, A. Viessmann, A. Waadt, G. Bruck, and P. Jung, "A Cognitive Radio Realization based on a Petri Net Approach," in *Proceedings of the 7th International Symposium on Wireless Communication Systems (ISWCS)*, September 2010, pp. 1031 –1035.

[5] J. Mitola, "Cognitive Radio - An Integrated Agent Architecture for Software Defined Radio," Ph.D. dissertation, Royal Institute of Technology, Kista, Sweden, 2000.

[6] A. M. Wyglinski, M. Nekovee, and Y. T. Hou, *Cognitive Radio Communications and Networks - Principles and Practice*. Academic Press, 2009.

[7] *Digital Video Broadcasting (DVB): Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and other Broadband Satellite Applications (DVB-S2)*, ETSI Std. EN 302 307 v1.2.1, August 2009.

[8] *Digital Video Broadcasting (DVB): Frame Structure Channel Coding and Modulation for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2)*, ETSI Std. EN 302 755 v1.1.1, September 2009.

[9] Avnet, "EXP Expansion Connector Specification Revision 1.4," September 2007.

[10] A. Viessmann, C. Kocks, A. Waadt, C. Spiegel, A. Burnic, G. Bruck, P. Jung, J. Kim,

Y. Lim, and H. W. Lee, "An Implementation-Friendly Synchronization Algorithm for Multimedia HDTV Receivers," in *Proceedings of the 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, November 2009, pp. 1 –5.

[11] *GY/T 220.1-2006: Mobile Multimedia Broadcasting Part 1 - Frame Structure, Channel Coding and Modulation for Broadcasting Channel*, SARFT - The State Administration of Radio, Film and Television Std. GY/T 220.1-2006, 2006.

[12] B. Razavi, *RF Microelectronics*. Upper Saddle River, NJ, USA: Prentice Hall, 1998.