Dynamic Spectrum Access AAF Platform

Shreyas Bhargav Raghunathan*, Maarten van den Oever*, Rahman Doost-Mohammady*, Przemysław Pawełczak*,
Ibrahim Budiarjo*, Marnix Heskamp†, Qiwei Zhang†, André Kokkeler†, Homayoun Nikookar*, Zhen Qin*,
Ramin Hekmat*, and Leonard P. Ligthart*
*Department of EEMCS, Delft University of Technology, Mekelweg 4, 2600 GA Delft, The Netherlands
†Department of EEMCS, University of Twente, Zilverling 4096, 7500 AE Enschede, The Netherlands
Email: {s.b.raghunathan, maartenvandenoever, r.doost, z.qin}@student.tudelft.nl, {p.pawelczak, i.budiarjo, h.nikookar,
r.hekmat, l.p.ligthart}@ewi.tudelft.nl, {m.heskamp, q.zhang, a.b.j.kokkeler}@ewi.utwente.nl

***Abstract*—The article provides basic information on the Dynamic Spectrum Access platform developed by AAF Freeband project sponsored by the Dutch Ministry of Economic Affairs.**

I. Introduction

The AAF Freeband project [1], started in 2004, is one of the first European national projects that focused solely on Dynamic Spectrum Access (DSA) application to contemporary emergency networks. One of the objectives of the project was to develop a proof of concept platform that would show the DSA capabilities, implementing some of the theoretical developments within AAF. This article will provide a brief description of the developed platform.

Briefly speaking, the AAF demo is composed of three independent components:

P1 Adaptive OFDM carrier selection based on a P25M [2] board (without RFE) with cooperative spectrum sensing based on the USRP platform [3];

P2 A complete ODFM receiver based on the Montium platform [4];

P3 An independent enhanced spectrum sensing platform based on the USRP written without GNU Radio components [5].

P1 is the main DSA platform, while P2 and P3 serve as independent and complementary platforms to P1.

This paper is structured as follows. In Section II we briefly describe each of the above components of the AAF demo and in Section III we summarize the paper.

II. Brief Platform Description

We now describe each of the Px platforms independently. We start with P1 description.

A. P1: Main DSA Platform

The main DSA proof of concept platform was developed by Delft University of Technology between 2007 and 2008 by graduate students and PhD researchers of the Wireless and Mobile Communications Group and Telecommunication and Remote Sensing Technology Group at the Department of Telecommunications.

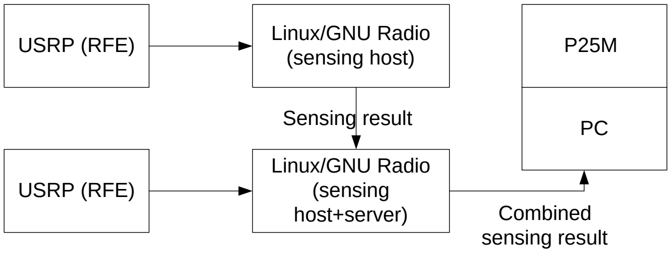

USRP (RFE)
Linux/GNU Radio (sensing host)
P25M
Sensing result
PC
USRP (RFE)
Linux/GNU Radio (sensing host+server)
Combined sensing result

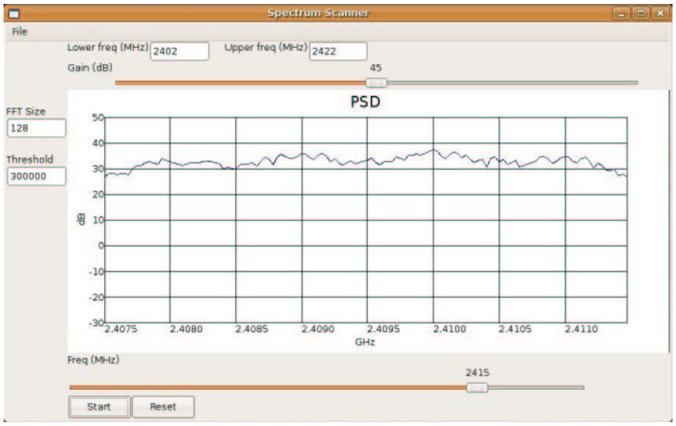
Fig. 1. Schematic representation of P1 platform.

The whole P1 platform is constructed of two main components: a cooperative spectrum sensing platform based on USRP devices, and OFDM adaptive carrier selection based on the P25M SDR platform. The spectrum sensing part sends real-time signals on the activity of the Primary Users (PUs) (in this case WLAN activity) to SDR P25M platform, which adapts its waveform based on information obtained, i.e., deactivates carriers on the channels where a PU was detected. The whole setup is schematically presented in Fig. 1, while the photograph of the whole setup is given in Fig. 3.

Specifically, the spectrum sensing part is constructed of two PCs running Ubuntu Linux with complete GNU Radio installed, connected to individual USRP with 2.4 GHz daughterboard. Each laptop runs its own spectrum sensing programme in GNU Radio. Both laptops are connected via the Samba protocol to exchange sensing data, while one of the laptops serves as a sensing server and combines spectrum sensing information from both of GNU radio programs. In the setup a simple measurement combining is performed. The outcome of the measurement is sent, again via the Samba protocol, to the PC hosing the P25M SDR platform as a vector of zeros and ones, where ones symbolize carrier occupancy by PU. An example GUI of the spectrum scanner (server side) is shown in Fig. 2.

The second part of the P1 platform consists of the P25M base board and a server computer. The P25M board consists of a TI C6713 DSP processor and a Xilinx Spartan 3 FPGA.

This work has been supported by the AAF Freeband program of the Dutch Ministry of Economic Affairs.



Lower freq (MHz) 2402
Upper freq (MHz) 2422
Gain (dB) 45
FFT Size 128
Threshold 300000
Freq (MHz) 2415

Fig. 2. GUI of the spectrum scanner of the P1 platform.

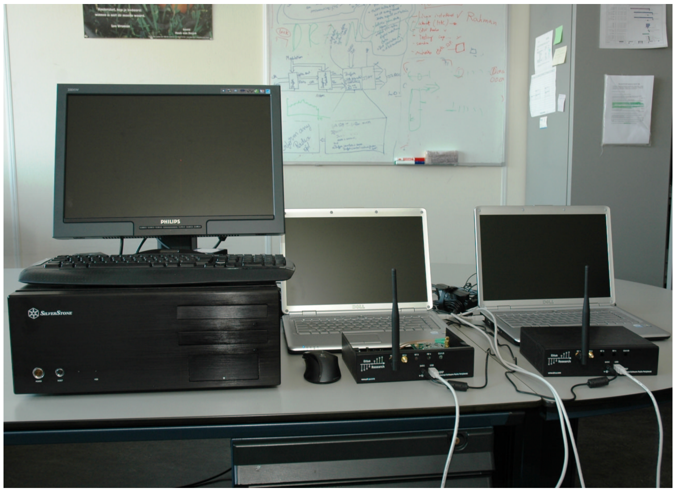


Fig. 3. Photograph of the P1 platform. Stationary PC contains P25M platform, while two laptops are connected to two USRP devices (black boxes in front of laptops), serving as spectrum scanners.

The computer runs a server that enables communication with the DSP processor. The DSP processor is capable of communicating with the FPGA.

The DSP Processor is used for feeding the FPGA with data. The P25M runs FPGA logic that enables adaptive modulation. This enables the usage of a different modulation scheme for each carrier. The modulation depth can vary from BPSK up to 64-QAM. The data about the spectrum occupancy is loaded to the board as a vector. With this vector the carriers are adaptively switched on and off.

The implementation is done using a mixture of VHDL and C++. The software used for the server computer is created in C++ using Borland Builder. The programs for the DSP Processor are also in the C++ language using Code Composer Studio. The logic for the FPGA is created using Xilinx ISE.

Xilinx ISE is also used for the simulation of the logic. In the demo the intermediate results of the FPGA are shown using Matlab plots. Plots of the constellation, the output spectrum and the deactivated spectrum are generated.

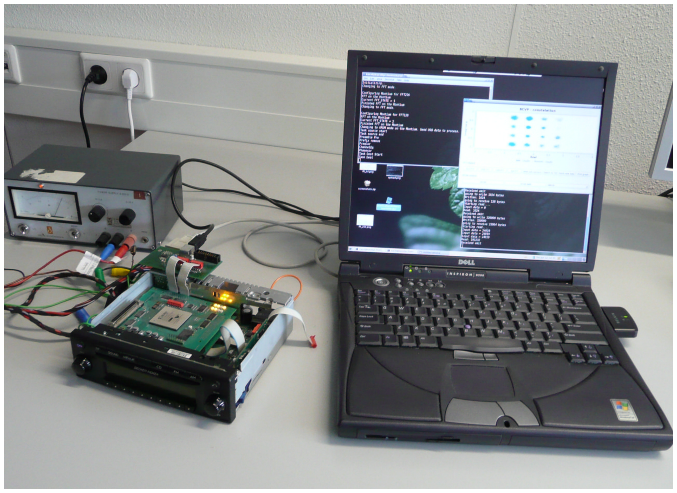


Fig. 4. Photograph of the P2 platform. Laptop is connected via USB and Serial port to Montium-based BCVP platform (opened black box), for the purpose of visualization of the OFDM modulation. Grey box behind BCVP contains power source for BCVP.

Two limitations of the presented setup is the lack of RFE at the P25M, and artificial spectrum sensing measurement result. Since at the time of P1 development university had access to 2.4 GHz daughter board only, spectrum sensing results using USRP were not reliable. It was due to the fact that USRP at was simply too slow to perform accurate measurement in real-time in 2.4 GHz range.

B. P2: Montium-based OFDM Implementation

The second platform, called symbolically P2 and shown in Fig. 4, was developed by University of Twente's Computer Architecture for Embedded Systems Group at the Department of Computer Science. P2 is composed of PC connected to the, so called, Basic Concept Verification Platform (BCVP) via USB and Serial port. PC sends reconfiguration data to the BCVP and retrieves results from the BCVP. A graphic user interface runs on the PC to interact with the BCVP and display the results.

Major components in the BCVP are two ARM processors, ARM920 and ARM946, where ARM946 is the processor used by default. The ARM946 use two tightly coupled memories, one instruction memory of 32 Kbytes and one data memory of 64 Kbytes. Whole BCVP has access to external Memory of 3 Mbytes in total. Another major part of BCVP is FPGA, which emulates three Montiums connected by a circuit switched Network-on-Chip. BCVP also includes USART. External constant 12 V power supply for the BCVP is needed. Schematic representation of BCVP is given in Fig. 5.

Some brief description is needed on the Montium architecture. Montium targets the 16-bit DSP algorithm domain [4]. At first glance the Montium bears a resemblance to a Very Long Instruction Word processor. However, the control structure of the Montium is optimized to minimize the control overhead which is imperative for energy efficiency. It consists of two major parts: the Communication and Configuration Unit (CCU) and the reconfigurable Tile Processor (TP). The CCU

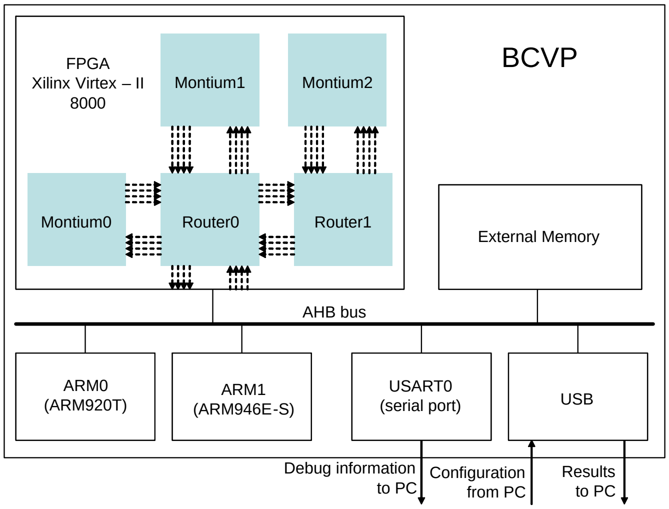


Fig. 5. Schematic representation of BCVP.

implements the interface for off-tile communication. The off-tile interface depends on the interconnect technology that is used in the Sysytem-on-Chip. The TP is the computing part that can be configured to implement a particular algorithm.

The five identical Arithmetic Logic Units (ALUs) in a tile can exploit spatial concurrency to enhance performance. The data path of the ALUs has a width of 16 bits and the ALUs support both signed integer and signed fixed-point arithmetic. The parallelism demands a very high memory bandwidth, which is obtained by having 10 local memories in parallel. The local memories imply a good locality of reference. A relatively simple sequencer controls the entire tile processor. The sequencer selects configurable tile instructions that are stored in the decoders.

Each one of four 16-bit inputs to an ALU has a private input register file that can store up to four operands. The input register file cannot be bypassed, i.e., an operand is always read from an input register. Input registers can be written by various sources via a flexible interconnect. Two 16 bit outputs from an ALU are connected to the interconnect. Neighboring ALUs can also communicate directly, i.e., the West-output of an ALU connects to the East-input of the ALU neighboring on the left.

Each local SRAM is 16 bit wide and has a depth of 1024 positions, which adds up to a storage capacity of 16 Kbit per local memory. A reconfigurable address generation unit (AGU) accompanies each memory. The AGU can generate the most frequently used address patterns, but when needed also an ALU can generate address patterns. It is also possible to use the memory as a lookup table for complicated functions that cannot be calculated using an ALU, such as sine or division (with one constant). A memory can be used for both integer and fixed-point lookups.

In the demonstration a complete OFDM receiver running on the BCVP platform is presented. Constellation plot of received signal is shown by a graphic user interface on the PC retrieving the results from the BCVP. The most computational intensive task, the FFT, in the OFDM receiver runs on the Montium[1]. However, the FFT can be switched to the ARM at run-time, see an example of such switching in Fig. 6. The small changes in the constellation by switching from the Montium to the ARM can be noticed due to the different precisions of different processors. Constellation plots from different channel model can also be generated.

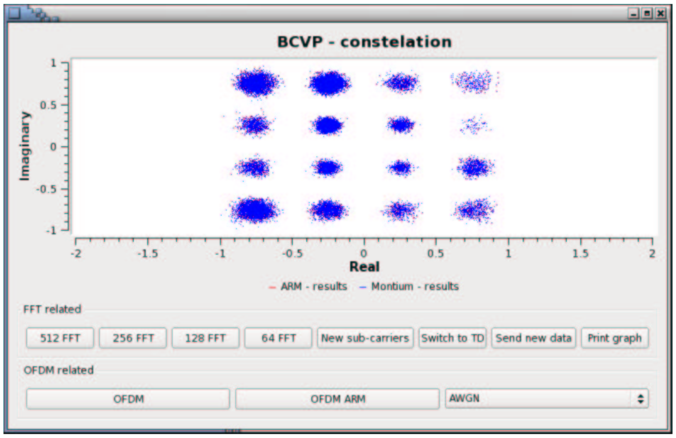


Fig. 6. GUI of P2. In the figure switching the FFT execution from the Montium to the ARM is shown.

C. P3: Enhanced Spectrum Sensing Platform

Finally, the goal of the P3 platform, see Fig 7, is to demonstrate physical layer issues involved in spectrum sensing. P3 was developed by University of Twente's Signals and Systems Group at the Department of Electrical Engineering. It consists of a PC application running on Windows XP, which combines several system level simulations with a GUI. It can also connect to a USRP to receive live radio signals, which can be captured to file, or processed at real time. Computational intensive algorithms like cyclostationary feature detection can be done offline.

The platform is designed as a hybrid between simulation and implementation. The reason behind such approach is as follows. The main problem with a full simulation is that it is difficult to model all relevant aspects of reality. It is for example difficult to simulate a transmitter and receiver that work on sampling rates that have a true non-integer ratio between them. Especially when analyzing detection problems in which many samples are correlated and averaged, such implicit synchronization has to be avoided, as it can result in artificial correlations. On the other hand, the problem with working only with live signals is that it is difficult to control all relevant aspects of reality, like fading and primary user behavior. Furthermore, results obtained from real world signals are difficult to reproduce. In simulation mode, an ensemble of

[1] Dynamic reconfigurable FFT that runs on the Montium can switch to different sizes. It can also switch from a radix-2 FFT to a novel sparse FFT that only produces the required FFT bins. The sparse FFT is efficient in the context of OFDM based DSA and multi-resolution spectrum sensing. For the detail of the sparse FFT see [6].

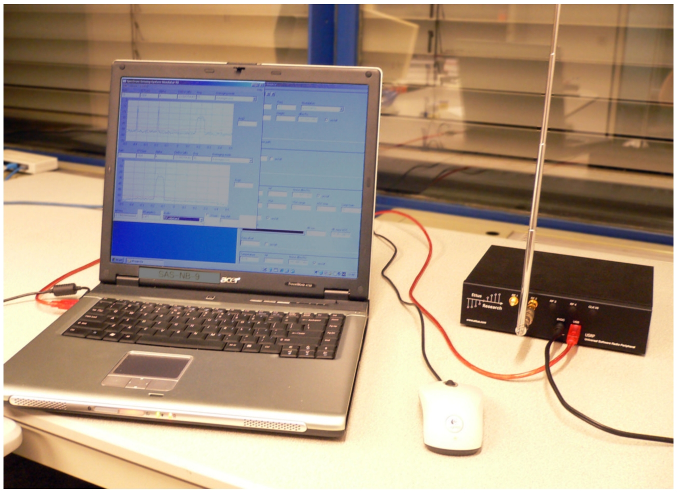


Fig. 7. Photo of P3 platform. Laptop running the simulator and signal processing blocks is connected to USRP.

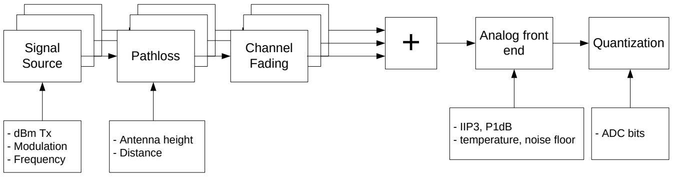
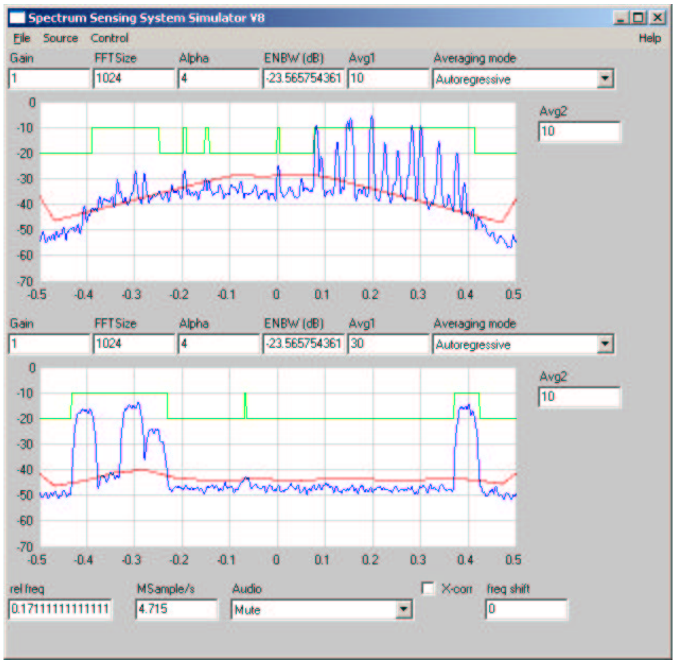


Fig. 8. Block scheme for generating simulated signals in P3.

signals is generated in software by a system as depicted in Fig. 8. Various parameters can be adjusted through graphical user interface windows. When connected to a USRP the demonstrator can receive live radio signals, see Fig. 9.

III. Summary

This article described a briefly Dynamic Spectrum Access platform developed by the AAF project. A work is ongoing to improve the capabilities of presented hardware. For example in case of P1 platform, RFE installation is required, and tests of cooperative spectrum sensing platform are needed, especially for non-UNII frequency range.



Spectrum Sensing System Simulator V8

Gain: 1
FFTSize: 1024
Alpha: 4
ENBW (dB): -23.565754361
Avg1: 10
Averaging mode: Autoregressive
Avg2: 10

Gain: 1
FFTSize: 1024
Alpha: 4
ENBW (dB): -23.565754361
Avg1: 30
Averaging mode: Autoregressive
Avg2: 10

rel freq: 0.17111111111111
MSample/s: 4.715
Audio: Mute
☐ X-corr
freq shift: 0

Fig. 9. Developed GUI of P3 showing spectra of TETRA downlink signal received through a television tuner frontend connected to USRP.

References

[1] AAF Program Home Page. [Online]. Available: http://aaf.freeband.nl

[2] Innovative Integration P25M Memo. [Online]. Available: http://www.innovativedsp.com/products/p25m.htm

[3] Universal Software Radio Peripheral Home Page. [Online]. Available: http://www.ettus.com

[4] G. J. M. Smit, P. M. Heysters, M. Rosien, and B. Molerkamp, "Lessons learned from designing the MONTIUM – a coarse-grained reconfigurable processing tile," in *Proc. International Symposium on System-on-Chip*, Tampere, Finland, Nov. 16–18, 2004, pp. 29–32.

[5] GNU Radio Home Page. [Online]. Available: http://www.gnu.org/software/gnuradio

[6] Q. Zhang, A. B. J. Kokkeler, and G. J. M. Smit, "An efficient FFT for OFDM based cognitive radio on a reconfigurable architecture," in *Proc. IEEE CogNet'07*, Glasgow, Scotland, June 24–28, 2007, pp. 6522–6526.