Efficient Baseband Modem Physical Implementation for Fixed Broadband Wireless Access Networks

Labros Bisdounis
Electrical Engineering Department
Technological Educational Institute (T.E.I.) of Western Greece
Patras, Greece

***Abstract*—This paper presents the physical implementation of the digital part of an OFDM-based baseband modem for point-to-multipoint fixed broadband wireless access (FBWA) solutions. It is compliant with the corresponding IEEE 802.16 standard and compatible to a fixed WiMAX profile. The adopted realization approach is based on an array of processing elements belonging to a case of computing systems characterized by having hundreds of embedded processing elements and memories (massively parallel processor arrays). The approach offers the performance, the computational density and the programmability needed for the implementation of modern wireless communication systems.**

Keywords—baseband modem; broadband access networks; wireless communication systems; IEEE 802.16; WiMAX; OFDM; massively parallel processor arrays.

I. INTRODUCTION

Fixed wireless broadband is a competitive alternative to DSL, providing a set of services similar to those of traditional fixed-line broadband but using wireless as the medium of transmission. Fixed broadband wireless access (FBWA) solutions are usually being specified for point-to-multipoint applications, i.e. for use by residential customers and small to medium sized enterprises (Fig. 1) [1]. They will support a wide range of voice and data services, using a wireless system to connect the customer premises to other users and networks.

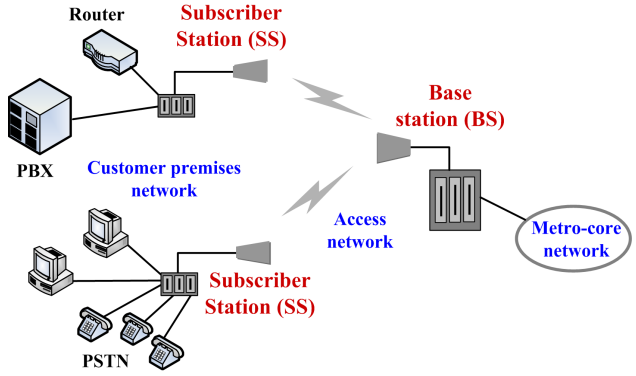

Fig. 1. Point-to-multipoint FBWA network topology

The WiMAX (worldwide interoperability for microwave access) industry forum [1]–[2] certifies broadband wireless solutions mainly based on the wireless metropolitan area networking (WMAN) standard developed by IEEE 802.16 group [3]. In 2004, this group has completed and approved the IEEE 802.16d or 802.16-2004 version of the standard [2],[4], targeted fixed applications (usually referred as fixed WiMAX). Next versions of the standard concern nomadic and mobile applications [1],[5] and referred as mobile WiMAX. The IEEE 802.16d version of the standard covers fixed non-line-of-sight (NLOS) applications in the 2GHz–11GHz band, using an orthogonal frequency division multiplexing (OFDM) physical layer [2],[6] (i.e. baseband modem). For practical reasons of interoperability, fixed WiMAX uses subsets of the standards' options (certification profiles), including 3.5GHz and 5.8GHz systems operating over 3.5MHz up to 10MHz channel and based on a 256-points Fast Fourier Transform (FFT) OFDM physical layer with a point-to-multipoint or mesh MAC layer. The profiles use frequency division duplexing (FDD) or time division duplexing (TDD) [1]–[2].

In this paper, an efficient implementation and the validation of the digital part of an OFDM-based physical layer is presented, which is compatible to an approved fixed WiMAX profile. It is based on a 256-points FFT, uses TDD and operates over 10 MHz channel bandwidth, however it is parametric also supporting 5 MHz. A brief description of the baseband modem algorithmic entities is provided in Section II.

Regarding the implementation, the adopted approach is based on an array of processing elements that belongs to a particular case of computing systems characterized by having hundreds of processing elements and embedded memories and referred as massively parallel processor arrays (MPPAs) [7]–[9]. Traditionally, designers had to choose between rapid time-to-market with microprocessors, DSPs and FPGAs, and tight size, high-performance and low-power consumption using ASICs. MMPAs help to bridge this gap by offering programmability coupled with the ability to increase the computational density, accelerate the performance and reduce the power consumption of the designed system.

The afore-mentioned baseband modem is implemented by using the picoArray architecture [9] that contains several hundreds of heterogeneous processing elements connected through a compile-time scheduled interconnect. The picoArray devices are designed to be connected together enabling multi-chip solutions to be easily realized for DSP applications which require additional processing. The devices are supported by a complete tool suite in order to implement, debug and verify systems on them. Details for the implementation platform, the realization, as well as the validation of the baseband modem are provided in the concluding Section III.

This work was performed within a subproject of the project 'International Cooperation for the Industrial Research and Development Activities in Pro-competitive Phase', supported by the Operational Programme 'Competitive-ness & Entrepreneurship I' (NSRF) and Intracom Telecom S.A. (prime partner of the subproject).

978-1-4673-9680-6/16/$31.00 ©2016 IEEE

II. Baseband Modem Description

As mentioned in Section 1, the baseband modem (physical layer) under consideration is based on the OFDM scheme that offers good resistance to multipath, allows operation in NLOS conditions and is capable of supporting very high data rates [2], [6]. The block diagrams for both transmit and receive paths of the baseband modem's digital part are shown in Fig. 2(a) and (b), respectively.

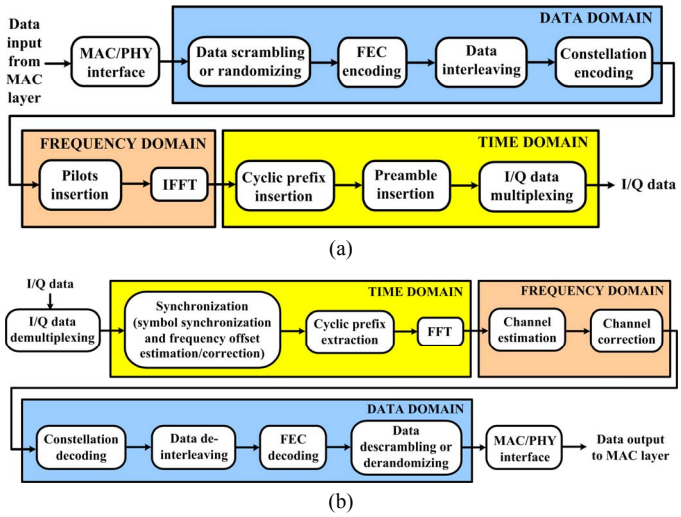

Fig. 2. Baseband modem's block diagrams: (a) transmit path, and (b) receive path

In the transmit path, initially the binary input data are randomized. The purpose is to prevent a long sequence of 1s and 0s, which will cause timing recovery problem at the receiver. The input sequence becomes more disperse so that the dependence of input signal's power spectrum on the actual transmitted data can be eliminated. The randomized data are then fed to the Reed Solomon (RS) encoder, which is a part of forward error correction (FEC) encoder. RS coding is an error-correction technique, in which input data is over-sampled and parity symbols are calculated and appended with original data [6]. In this way, redundant bits are added to the actual message providing immunity against severe channel conditions. The error coded bits are further coded by convolutional encoder with a coding rate of ½ and a constraint length of 7 [4]. Since six different data rates are supported by the adopted standard, proper puncture is applied after the convolutional encoding. The encoded data is passed through a bit interleaver; the bit interleaver size is set to the size of one OFDM symbol and the size of the symbol depends on the used modulation (384 bits for QPSK, 768 bits for 16-QAM & 1152 bits for 64-QAM) [4]. The bits within a symbol are rearranged in such a fashion so that adjacent bits are placed on non-adjacent subcarriers. Interleaving is performed to protect the data from burst errors during transmission. The constellation encoder maps the input bits onto different subcarriers. Different map (modulation) schemes are supported (BPSK, QPSK, 16-QAM, 64-QAM). The interleaved bits are divided in groups of 1, 2, 4 or 6 bits (data subcarriers) and mapped into complex numbers representing BPSK, QPSK, 16-QAM or 64-QAM values, respectively. The basic unit in data transmission is the OFDM symbol, and in the case of the adopted standard each OFDM symbol consists of 192 constellation points (data subcarriers). To facilitate channel estimation at the receiver, eight pilot subcarriers are added to each 192 data subcarrier and by applying the 256-points IFFT, the output is transformed into a time-domain signal [4]. Zero bits are padded equally at the beginning and the end of each OFDM symbol to perform the 256-points IFFT. To make the transmit system robust to multipath propagation, a cyclic prefix (CP) is added to each OFDM symbol, by simply coping the last portion data in an OFDM symbol to the beginning. The CP duration depends on the used channel bandwidth (3–25% of the symbol duration). Using a 10MHz spectrum, the symbol duration is 22.2μs and the adopted CP duration is 2.8μs (12.5%), leading in a total duration of 25μs.

The symbols are packed into frames before sending. The adopted frame duration is 5ms. In the case of TDD, where the uplink and downlink transmissions occur at different (complementary) times while sharing the same frequency, each frame consists of the downlink and the uplink subframes. In each subframe proper preambles are inserted. The preambles are structured as either one of two symbols [4]. Each of the preamble's symbols contains a cyclic prefix, which length is the same as the CP for data OFDM symbols. The preambles are transmitted before the data symbols and their purpose is to facilitate the synchronization task at the receiver. After the preambles, for each subframe, a frame control header of one symbol is transmitted. The last action of the transmit path is the multiplexing of the produced I/Q data, in order the digital output signals to be converted to the transmitted analog signal.

In the receive path after the demultiplexing of the received I/Q data, and before the receiver can demodulate the subcarriers, it has to perform the required synchronization tasks. Achieving synchronization involves detecting the symbol boundaries (to avoid inter-symbol interference) and estimating/correcting any frequency offsets in the received signal [6]. In an OFDM link, the subcarriers are orthogonal only if the transmitter and the receiver use exactly the same frequencies. Any frequency offset results in inter-subcarrier interference. The timing and frequency synchronization is achieved by utilizing the repetitive nature of the preambles [10]. The time domain of the receiver is continued with the CP extraction in order the remaining samples of each symbol to be modulated by the FFT. The transition of the received signal from the time to the frequency domain is achieved by applying a 256-points FFT. The pad bits added at the transmitter are removed after the application of the FFT. During reception, in order to recover the original signal at the receiver, the transfer function of the channel has to be estimated.

Channel estimation is carried out by exploiting known data (pilot subcarriers), which are scattered at predefined locations inside OFDM symbols during transmission. The principle is to estimate the channel transfer function at pilot locations and then to achieve estimation for all data locations by performing a two-dimensional interpolation [6],[11]. Once the channel estimates in each subcarrier position are obtained, channel correction has to be carried out. This task is performed by a frequency domain equalizer that is an array of complex multipliers (one for each subcarrier). In the data domain of the receive path, initially the decoding methods for the four

modulation schemes are applied by the constellation decoder. Then, the data are deinterleaved and inserted to the FEC decoder that is a soft decision Viterbi decoder followed by an RS decoder, which recovers the original data. Finally, the decoded data are derandomized, by using the same structure with that used at the transmitter.

III. Physical Implementation and Validation

As shown in Fig. 1, a fixed wireless access point-to-multipoint network consists of network units, such as base station (BS) and subscriber stations (SS). Fig. 3 presents a general diagram of a network unit, in which the digital part of the baseband modem is included as a separate block.

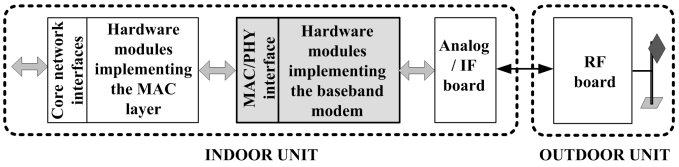

Fig. 3. Wireless access network unit

As mentioned in Section 1, the implementation of the baseband modem's digital part is based on the picoArray MPPA architecture (PC102 device) [9]. In this, 322 processing elements are organized in a two dimensional array, and communicate over a network of 32-bit unidirectional buses and programmable bus switches (Fig. 4). The processing elements contain three 16-bit RISC processor variants, which have varying amounts of memory and additional instructions to implement digital signal processing functions. They use a 3-way LIW scheduling [9]. More specifically, the array contains 240 standard processors with their own memory (768 bytes each) and multiply-accumulate unit, 64 memory processors with an 8704 bytes memory each and a multiply unit, and 4 control processors with a multiply unit and larger amount of memory (65535 bytes each) to implement control functionality. Each processor contains communication ports, which allow access to the interconnect bus for exchanging data with other processors, and is programmed using a configuration bus. In addition to the general purpose RISC processors, the array contains a column of 14 function acceleration units that include configurable hardware for accelerating a number of computational intensive tasks including correlation and trellis processing. Several interfaces are included, such as a host interface to communicate with an external microprocessor, an SDRAM interface and four data interfaces, which can be configured as asynchronous data interfaces or as bidirectional interfaces allowing the array to exchange data with others.

The baseband modem described in Section 2, has been implemented by using the PC102 devices hosted in the PC7218 hardware platform [12]. As shown in Fig. 5, in addition to the MPPAs, the platform hosts an MPC8560 processor (containing a high-performance embedded PowerPC core), an encryption engine, Ethernet and digital I/Q interfaces, as well as memory modules supporting the MPPAs and the MPC8560 processor. The programming of the MPPAs is performed through the host processor. The algorithmic entities of both transmit and receive paths of the baseband modem have been implemented in the two MPPAs of the platform (Fig. 6).

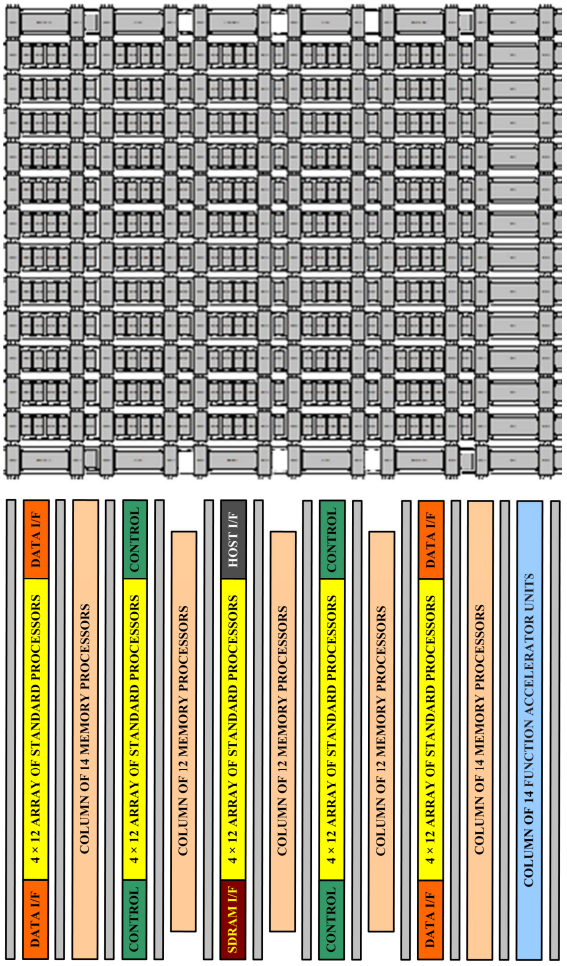

Fig. 4. Top-level diagram of picoArray MPPA [9]

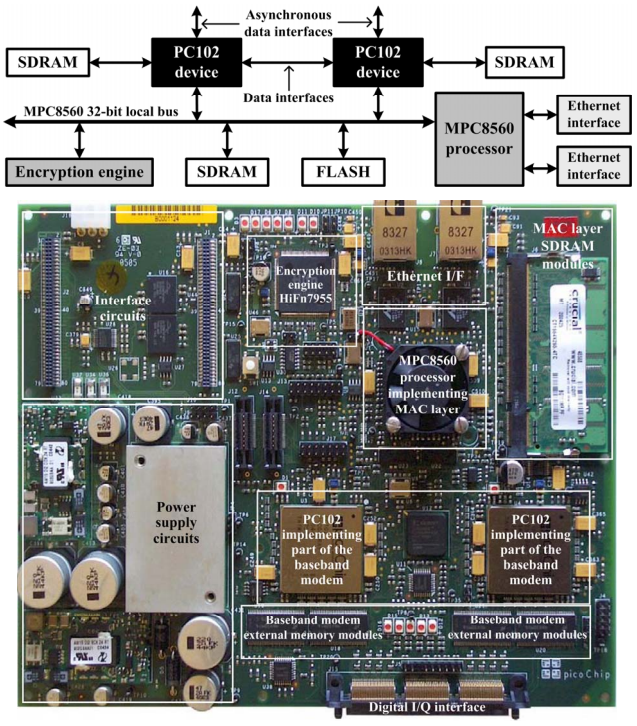

Fig. 5. Implementation platfrom [12]

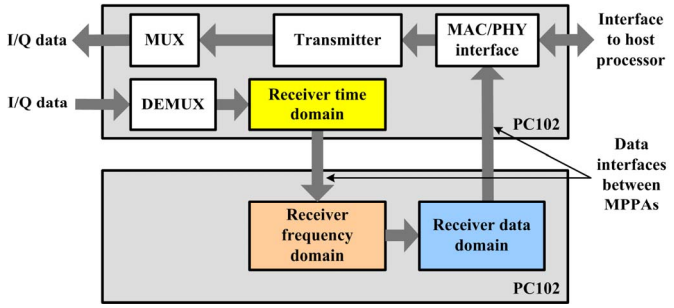

Fig. 6. Mapping of the baseband modem's functionality

For the implementation, the tool chain [13] supporting the used MPPA device has been used. Initially, the algorithmic entities were described in C. Structural VHDL was used to describe the structure of the overall system, including the relationship between entities and the signals which connect them together. The tool chain converts the input code into a form suitable for execution on the MPPA. It comprises a C compiler and a VHDL parser. The parser coordinates the compilation of the code and creates an internal representation for each entity and its signals. Whenever a block of C code is encountered the C compiler is invoked. The available design browser, which allows the design to be viewed graphically, as well as the cycle accurate simulator, which allow the testing and the verification of the entire design before moving to the hardware were used. After the mapping of the entities to the couple of MPPAs (Fig. 6), the available partitioning tool was used to handle the off-chip communication by using a pair of the available data interfaces. The last steps were the final assignment of the devices to each entity, along with the routing of all the signals linking the entities, as well as the debugging of the design on the real hardware.

For the validation of the implemented modem, the setup presented in Fig. 7 was used, in which the communication of the digital transceivers of a base station and a subscriber station has been analyzed. For the purposes of this task, an elementary MAC layer version running at the host processor, providing a bridge for the data packets between the Ethernet interface and the digital transceivers, was developed. In addition, the digital interface of the two stations was implemented in an FPGA device, hosted in a development board [14] that ensures access to the inputs and the outputs of the device. Since, the channel bandwidth was set to 10MHz, operating using TDD scheme with a 3:1 downlink-to-uplink ratio, the peak data rate for 64-QAM modulation is about 34.5Mbps and 8.5Mbps for the downlink and the uplink, respectively. The receive capability is reduced (about 8Mbps), mainly due to the presence of the Viterbi decoder. The reachability between the base and the subscriber stations was firstly tested through the Ping application. Transmit and response times were about 10ms and 25ms for the downlink and the uplink, respectively. After that, a file transfer process were take place between the stations, and according to the file size and the transfer time, it was found out that the average data rate was about 7.5Mbps. During the streaming of a video file by the base station, that was watched at the subscriber station with a typical media player, a data rate between 1Mbps and 3Mbps was observed. This was due to the fact that the video file rate was lower than the data rate that can achieve the developed system. Finally, the time needed for the connection setup between the stations was found almost equal to the duration of 100 frames (0.5 s).

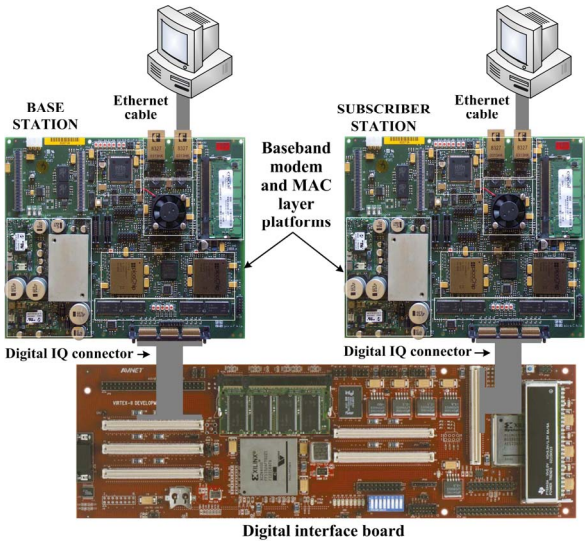

Fig. 7. Validation setup

REFERENCES

[1] L. Wu, and K. Sandrasegaran, "Overview of WiMAX Standards and Applications," in WiMAX Applications, S. Ahson, and M. Ilyas, Eds. Boca Raton, FL: CRC Press, 2008, pp. 15-33.

[2] J.G. Andrews, A. Ghosh, and R. Muhamed, Fundamentals of WiMAX: Undrerstanding Broadband Wireless Networking. Westford, MA: Pearson Education, 2007, pp. 1-46, 113-145.

[3] C. Eklund, R.B. Marks, K.L. Stanwood, and S. Wang, "IEEE Standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," IEEE Communications Magazine, vol. 40, pp. 98-107, June 2002.

[4] IEEE 802.16 Working Group, IEEE Standard for Local & Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems. New York: IEEE, 2004.

[5] A. Bacioccola, C. Cicconetti, C. Eklund, L. Lenzini, Z. Li, and E. Mingozzi, "IEEE 802.16: History, Status and Future Trends," Computer Communications, Elsevier, vol. 33, pp. 113-123, February 2010.

[6] A.R.S. Bahai, B.R. Saltzberg, and M. Ergen, Multi-Carrier Digital Communications: Theory and Applications of OFDM. Boston, MA: Springer, 2004, pp. 99-188.

[7] F. Hannig, and J. Teich, "Design Space Exploration for Massively Parallel Processing Arrays," in Proc. Int. Conf. Parallel Computing Technologies, September 2001, pp. 51-65.

[8] D.B. Thomas, L. Howes, and W. Luk, "A Comparison of CPUs, GPUs, FPGAs, and Massively Parallel Processor Arrays for Random Number Generation," in Proc. ACM SIGDA Int. Symp. Field-Programmable Gate Arrays, February 2009, pp. 63-72.

[9] G. Panesar, D. Towner, A. Duller, A. Gray, and W. Robbins, "Deterministic Parallel Processing," Int. J. Parallel Programming, Springer, vol. 34, pp. 323-341, August 2006.

[10] T.M. Schmidl, and D.C. Cox, "Robust Frequency and Timing Synchronization for OFDM," IEEE Trans. Communications, vol. 45, pp. 1613-1621, December 1997.

[11] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems," IEEE Trans. Broadcasting, vol. 48, pp. 223-229, September 2002.

[12] picoChip Designs Ltd., PC7218 Baseband Reference Design Product Brief. Bath: picoChip Designs Ltd., 2005.

[13] G. Panesar, A. Duller, A. Gray, and D. Towner, "Multi-Core System-on-Chip in Real World Products," in Multi-Core Embedded Systems, G. Kornaros, Ed. Boca Raton, FL: CRC Press, 2010, pp. 369-398.

[14] Avnet Design Services, Xilinx Virtex-II Development Kit User's Guide. Phoenix, AZ: Avnet Inc., 2002.