A REAL-TIME PRE-MIMO-LTE SOFTWARE RADIO TESTBED

Ronald Chen[1], *Qipeng Cai*[1,2], *Karsten Alecke*[1], *Onoriu Lazar*[1] *and Thomas Kaiser*[1,2]

[1]mimoOn GmbH, Bismarckstrasse 120, 47057 Duisburg, Germany
Phone: +49 203 306 4500, Fax: +49 203 306 4554, Email: info@mimoOn.de
Web: www.mimoOn.de
[2]Leibniz Universität Hannover, Institute of Communications Technology,
Appelstr. 9A, 30167 Hannover, Germany
Web: www.ikt.uni-hannover.de

ABSTRACT

The contribution describes the implementation of a MIMO (Multiple Input - Multiple Output) OFDM (Orthogonal Frequency Division Multiplex) online demonstrator with certain parameters for the 3rd Generation Partnership Project (3GPP) Long-Term-Evolution (LTE) system, by employing our scalable multi-standard Software Defined Radio (SDR) Testbed [1][2], which is built up with Sundance modular hardware components [3]. In comparison to prior MIMO test-bed installations, the entire digital signal processing is achieved in real-time using Texas Instruments DSPs and Xilinx FPGAs. It indicates that our MIMO Testbed structure can accelerate the development and evaluation of the techniques for the next generation wireless communication systems.

1. INTRODUCTION

In the past years the demand on high data rates in mobile communication is increasing dramatically. Since the available spectrum used for such communication systems is limited by regulations, another possibility has to be found to handle very high data rates. A very powerful approach is Spatial Multiplexing (SM) MIMO (Multiple Input - Multiple Output). By the usage of multiple transmit (Tx) and receive antennas (Rx), the achievable data rate compared to a Single Input - Single Output (SISO) system can be increased by a factor of about Tx, while the occupied bandwidth remains constant. Currently MIMO is considered in IEEE802.11n and IEEE802.16 and will be an inherent part in future wireless communication systems like 3GPP LTE.

While MATLAB MIMO simulation chains are well suited for theoretical MIMO research activities, the achievable results can not reflect the performance of MIMO in real world environments and system integration aspects. The task of our activities is the implementation of a 2x2 Spatial Multiplexing MIMO OFDM real-time demonstrator with certain parameters for 3GPP LTE system to verify MIMO algorithms under real world conditions. Furthermore, the platform installation should help us to figure out how to handle occurring challenges on the way from theory to prototype.

With the standardization progress of IEEE 802.16e, which extends the WiMAX from Fixed Wireless Access (FWA) to Mobile Wireless Access (MWA), current Universal Mobile Telecommunications System (UMTS) will be confronted with fierce competition from the upcoming mobile WiMAX. Starting from the end of 2004 3GPP works on the evolution of the 3G Mobile System, which involves UTRA-UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE) [4], in order to increase service provisioning, reduce user and operator cost, improve coverage and system capacity and increase data rate, reduce latency, so as to ensure the competitiveness of 3G technology during the next 10 years and beyond.

UTRA-UTRAN LTE adopts OFDMA as its downlink air interface technique instead of Code Domain Multiple Access (CDMA) in current UMTS UTRAN system to achieve higher spectrum efficiency in multipath wireless channel environment. By employing different number of subcarriers the OFDMA system can easily change the system bandwidth so as to ease the deployment in different RF bands in the radio regulation and more efficiently utilize the frequency resource. Furthermore, by dividing the frequency selective fading channel to multiple frequency flat fading channels, OFDMA has inherent property to integrate with MIMO technique to further enhance the system performance. For the LTE downlink, a 2 × 2 configuration for MIMO is assumed as baseline configuration, i.e. 2 transmit antennas at the base station and 2 receive antennas at the terminal side. The MIMO modes can be adjusted between spatial multiplexing and transmit diversity, depending on the channel condition.

In our implementation we focus on realizing a real-time 2 × 2 spatial multiplexing ”pre-LTE” demonstrator. This article describes our demonstrator platform with its digital signal processing and RF part. In Section 2 the system architecture is given, which involves the hardware setup, software configuration and the important interfaces. The software development, including development environment, DSP programming and host pc programming, is described in Section 3. Section 4 presents our new MIMO RF Front-end *MORFAN*. Finally, we give the conclusion in Section 5.

2. SYSTEM ARCHITECTURE

The real-time platform gives a demonstration of a wireless video stream transmission system. It consists of one transmitter as the base station and one receiver as the terminal. The digital hardware setup of the whole system is based on Sundance Software Defined Radio (SDR) modules. Basis at the transmitter and receiver side are Sundance SMT310Q carrier boards, plugged via Peripheral Component Interconnect (PCI) bus to a standard PC. The system architecture is depicted in Fig. 1.

At the transmitter side, the carrier board is equipped with an SMT395 DSP module and an SMT370 ADC/DAC module. The SMT395 exhibits a Texas Instruments (TI) 6416T fix-point Digital Signal Processor (DSP) running at 1000 MHz and a Xilinx Virtex II Pro FPGA. The stream server program encapsulates the video data into Internet Protocol

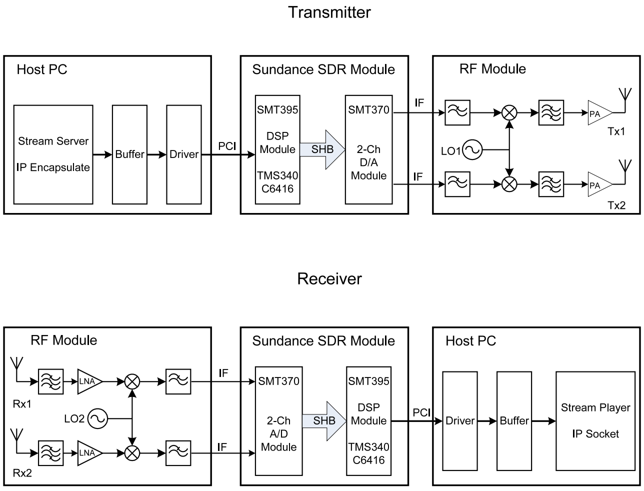


Figure 1: System Architecture

(IP) stream and saves the IP stream in the buffer allocated in the main memory of the host pc. The DSP module fetches the data in the buffer through the PCI interface provided by the abovementioned carrier board and then executes the entire digital baseband and Intermediate Freqeuncy (IF) signal processing algorithms of the transmitter. The driver of the carrier board offers the DSP module the methods to access the main memory of the host pc through PCI interface by providing C/C++ Application Program Interface (API) functions. The Xilinx FPGA on the DSP module takes care of the Sundance Highspeed Bus (SHB) interfacing between the DSP module SMT395 and the 2-Ch D/A module SMT370. The SHB interface is is able to transfer 32- bit data by a 100 MHz clock (400 MBytes/s). Via SHB the digital Intermediate Frequency (IF) signal is forwarded to the SMT370 to generate the analog signal using its integrated Digital to Analog Converter (DAC). The analog IF signals go through the low-pass filter and then the frequency spectrum is modulated to 2.4 GHz Industrial, Scientific and Medical (ISM) band by the linear mixers. The lower sideband spectrum of the modulated signal is rejected by the band-pass filter and only the power of the upper sideband signal is amplified and transmitted.

The hardware setup of the receiver is similar with the transmitter as shown in Fig. 1. However, SMT370 is configured as 2-Ch A/D module and the signal experiences the reciprocal of the transmitter. The signal is received by the receive antennas and then the RF front-end of the receiver converts them down to the low intermediate frequency. The IF signals coming from the two receiver antennas are sampled synchronously by the Analog to Digital Converters (ADC) on SMT370 module. The DSP module receives the digital samples from the SHB interface and accomplishes the entire digital signal process algorithms of the receiver. The final received IP packets are saved to the buffer in the main memory of the host pc through the PCI interface. The network layer program fetches the IP packets from the buffer and emits them to the certain IP port by IP socket programming. The video stream player always listens to the IP port and plays the video back.

3. SOFTWARE DEVELOPMENT

3.1 Development Einvironment

The whole system is based on a software defined radio platform. The main development efforts are therefore put into software programming, which can be classified to real-time algorithms realization and the host pc software programming as well as the interfacing between them, such as video buffer controlling.

For software development there are two different hardware platforms. The host pc softwares are running on a normal personal pc which has x86 architecture. However, the real-time digital signal processing programs are running on the TI TMS320C6416T processor. It is a dedicated fix-point digital signal processor with eight highly independent functional units and meanwhile has two high-performance embedded coprocessors [Viterbi Decoder Coprocessor (VCP) and Turbo Decoder Coprocessor (TCP)] that significantly speed up channel-decoding operations on-chip [5].

As shown in Fig. 2, The host pc softwares mainly consist of Graphic User Interface (GUI) program, video streaming software and the video buffer controlling program. Refering to the Open Systems Interconnection (OSI) model they belong to the application layer. Microsoft Visual C++ is chosen to develop the Graphic User Interface (GUI) and the video buffer controlling program. VLC media player , a free software stemmed from the software project-VideoLan, supports various audio and video formats (MPEG-1, MPEG-2, MPEG-4, DivX, mp3, ogg, ...) as well as DVDs, VCDs, and various streaming protocols. Furthermore, it can not only play back the video stream, but also stream all the video formats that it supports. Thus VLC media player is used as a server and as a client to stream and receive network streams in the transmitter and the receiver respectively. The communication pathes between the host pc and the DSP module, such as the host comport, are carried by the PCI interface. Sundance drive package provides the API functions for both the host pc and the DSP module to access the resourcs of each other through the PCI interface.

The programs running on the DSP module handle the functions belonging to the data link layer and physical layer. Because of the real-time requirement the programs should be optimized to reduce the calculation complexity and utilize

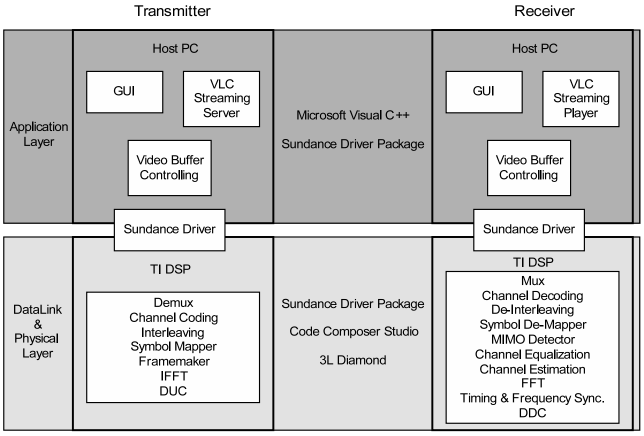


Figure 2: Development Environment

the adventage of parallel processing by felicitously programing the eight highly independent functional units in the DSP processor. The DSP program tool *Code Composer Studio* (CCS) from TI company and the real-time operation system *Diamond* from 3L company constitute the development environment for the DSP module. CCS is a dedicated integrated development tool for TI DSPs. With supporting programming with C/C++ language, linear assembly language and manual assembly language, it provides great flexibility for the demands from fast and easy programming to high efficient and parallel programming. It is natural to choose high level language such as C/C++ to program the tasks which are not time-critical because high level language is more understandable and much easier for debugging so as to accelerate the development progress. However, many tasks of digital signal processing, especially in the physical layer program, are time-critical. Even with the help of high efficient compiler and optimizer integrated in CCS the execution efficiency of the programs written in C/C++, such as Fast Fourier Transform (FFT), MIMO detector and Viterbi channel decoding, may not fulfill the timing requirement because the binary code generated from C/C++ language by the compiler and optimizer cannot fully utilize the parallel architechture of the DSP processor [6]. Therefore such kind of tasks have to be written in assembly language and multiple independent instructions can be manually assigned to different function units at the same clock cycle so as to achieve high parallel processing improvement [7]. CCS also provides some common digital signal processing functions in the library which have been already highly optimized for the parallel architecture of the DSP processor, such as autocorrelation, finite impulse response filter, FFT and so on [8]. Diamond, provided by 3L company, is another important portion for developing the DSP program, which is a real-time operation system specially for the Sundance SDR platform equipped with TI C6000 series DSP processor. It helps the programmer to manage the on-board resource, such as memory, SHB interface. More important, Diamond offers the methods for synchronizing the multiple tasks running on the DSP module as well as synchronizing the tasks running on the DSP module and the program running on the host pc [9]. After compiling the object files are linked with the standalone library of Diamond to generate stand-alone application program for the DSP module. The host pc program calls the API function provided by Sundance driver to load the standalone application program to the DSP module in the system booting phase.

3.2 DSP programming

All the digital signal processing functions including transmitter and receiver, which are shown in Fig. 3, are implemented on the DSP module, some functions can also be implemented on FPGA, which is up to the peak data rate of the system. In our implementation, we put all the digital signal processing functions on the DSP module for simplicity.

A common DSP programming technique, PING-PONG-buffering scheme, shall be used to process the data stream between the DSP module and external devices such as ADC module, DAC module or other DSP modules. Fig. 4 shows the PING PONG processing pseduo-code for the receiver.

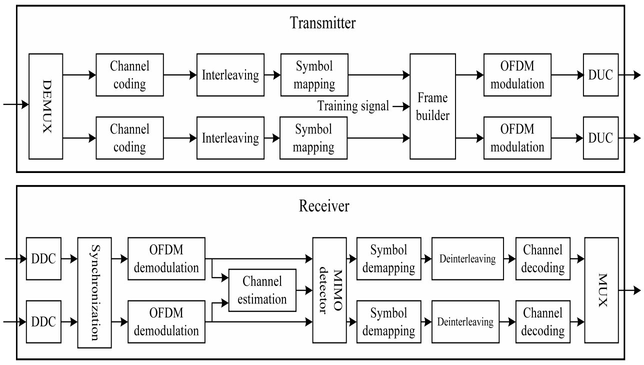


Figure 3: Digital signal processing

In order to run the digital signal processing algorithms in real time, some digital signal processing modules shall be simplified. For DUC and DDC module, the intermediate frequency (IF) is selected as the $1/4$ of the sampling frequency, and then for up converter, the mathematical operation is given as

$$\begin{aligned}&\text{Real}\{(i(n)+\mathrm{j}q(n))\mathrm{e}^{\mathrm{j}(\pi n/2)}\}\\&\quad=i(n)\cos(\pi n/2)+q(n)\sin(-\pi n/2)\end{aligned}\tag{1}$$

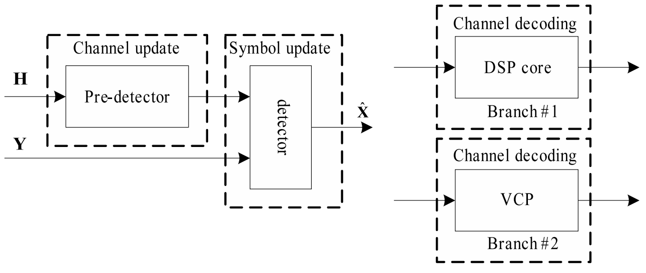
where $i(n)+\mathrm{j}q(n)$ is the complex baseband signal. It's shown that only four values $(1,0,-1,0)$ for both $\cos(\pi n/2)$ and $\sin(-\pi n/2)$. Furthermore, the interpolating filter for DUC can be implemented using polyphase filter. For transversal filter implementation, if there are 32 taps for example, and then for polyphase filter implementation, only 8-taps filters are needed. It's shown that half of the values for $\cos(\pi n/2)$ and $\sin(-\pi n/2)$ are 0, so half of the filtering operations can be eliminated. For DDC, we can use the similar simplification like DUC.

FFT/IFFT, MIMO detector and channel decoding are the most complex parts among the digital signal processing functions. For FFT/IFFT as well as FIR filter, we can use the functions in the DSP library, which are already highly optimized for the parallel architechture of the DSP processor. For MIMO detector, the module can be divided into pre-detector part and detector part which are shown in Fig. 5(a). The op-

```
Read first block from ADC to internal PING input buffer and wait for finish
while(1)
{
    while(1)
    {
        Read next block from ADC to internal PONG input buffer
        DDC operation is performed on internal PING buffer
        if (have not found timing) time synchronization
        if (found timing) copy data excluding cyclic prefix to a new buffer
        Wait for data reading finish for PONG input buffer
        if (got one whole frame data) break
    }
    OFDM demodulation
    Channel estimation
    •
    •
    •
}
```

Figure 4: Processing pseudo-code for the receiver

erations for pre-detector are only calculated when channel parameters are updated, and operations for detector are calculated along with each data symbol. We can assume the channel parameters are constant over several OFDM blocks or over one whole frame, and then computational complexity for MIMO detector can be reduced. Convolutional code is chosen for channel coding in our demonstration. The TI DSP processor TMS320C6416T integrates a programmable peripheral Viterbi Coprocessor (VCP), which can be used to accelerate the decoding of convolutional codes. 2 × 2 spatial multiplexing MIMO scheme is implemented in our demonstrator, so there are 2 data streams which need channel decoding. In order to improve data throughput of channel decoding module, we also implement one Viterbi decoder written in assembly language, which runs on the DSP core. Thus, as shown in Fig. 5(b), the first branch of the two data streams is processed by the Viterbi decoder running on DSP core and the second branch is handled by the VCP. These two decoders execute in parallel.



(a) (b)

Figure 5: MIMO detector and channel decoding

3.3 Host PC programming

We implement GUIs on the PCs with Windows XP operation system, which is connected to Sundance hardware platform, to set some parameters for the testbed and control the running of the testbed. We also implement one video transfer and display application, by which, it's easy to observe the packet lost and packet error to indicate the link quality of data transmission. Since the timing of the PC program shall be synchronized to the timing of the signal processing on DSP, besides GUI and video application thread we open another thread to implement the communication between PC and DSP.

VLC media player runs on the host pc of the transmitter as a video streaming server, on the contrary, it is deployed as a video player on the host pc of the receiver. The GUI of the transmitter accesses the video file on the demand of the user and calls VLC media player, which is running in background, to stream the video. At the receiver side, the host pc program sends the received IP packets to the certain IP port through IP socket programming. VLC media player, configured as the client mode by the GUI, always listens the certain IP port and fetches the stream data from it so as to play the video back.

4. MIMO RF FRONT-END

The RF front-ends seen in Fig. 1 are based on the test-bed described in [1]. The front-ends in the transmitter and receiver are built up using Mini-Circuits components that allow a modular and cost-efficient implementation.

The 7.5 MHz intermediate frequency signals received form the DAC are low-pass filtered and upconverted to the 2.4 GHz ISM-Band using a carrier signal generated by a Local Oscillator (LO). Both Tx branches are using the same LO (LO1) with the help of a power splitter. The signals are then band-pass filtered and sent through a Power Amplifier (PA) to the two Tx antennas.

The front end in the receiver is performing the reverse tasks, the signals received by the two Rx antennas are band-pass filtered, amplified using a Low Noise Amplifier (LNA) and dowconverted to the same intermediate frequency as used in the transmitter. The receiver uses a different LO (LO2) than the transmitter for generating the 2.4 GHz carrier signal. After downconversion the signals are sent to the AD converters on the SMT370 board.

Our current front-end configuration is replaced by a TIM standard compatible board *MORFAN* developed by mimoOn company. MORFAN is a daughter card based on the Sundance FPGA module SMT368 so as to extend the Sundance SDR platform to real world world-band MIMO transmission in the 2.4-2.5 GHz & 4.9-5.875 GHz ISM bands with a possible 3-dB signal bandwidth. The block diagram of MORFAN and its base module SMT368 is shown in Fig. 6.

Each MORFAN implements 2-antenna tranceiver system based on a three chips solution, which yields a compact design RF Front-end and an improved performance compared to other solutions. A single-chip MIMO RF front-end module SE2545 provided by SiGe is adopted, which contains basically all circuitry between the tranceiver and the antenna. SE2545 implements all the functionality of the power amplifiers, power detector, T/R switch, diplexers and associated matching. Two tranceiver chips of type Maxim MAX2829 are used to up- and down-convert signals between the ISM bands and the base band. The MAX2829 is specifically designed for MIMO/Smart Antenna application, which provides a fully receive path, transmit path, VCO, frequency synthesizer and base-band control interface.

MORFAN not only focus on the RF solution, but also in digital area where MORFAN is equipped with 2 two on-board signal converters AD9861, each of them integrates two 10-bit ADC as well as two 10-bit DAC. No further ADC or DAC modules are required for signal tranforming between analog and digital. This capability results an efficient MIMO RF front-end solution. The MORFAN daughter card fits onto the Sundance SMT368 FPGA base module, which has a responsibility to setup and maintain the control of MORFAN during the operation. The base-band samples for the both

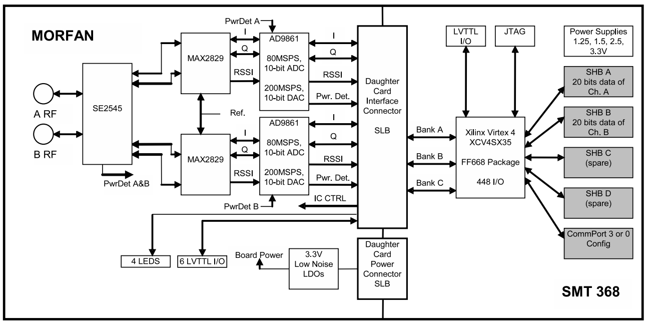


Figure 6: MORFAN and SMT368 Block Diagram

antennas as well as all control possibilities of the used chips are connected with the FPGA base module through the SLB connector. Additionally Receive Signal Strength (RSSI) and Power Detection Signals are also given in digital form to the underlying FPGA base module, so that digital Automatic Gain Control (AGC) and Power Control (PwC) algorithms can be implemented.

5. CONCLUSION

We have tried to exploit the existing resources on our previous off-line MIMO test-bed and realized a real-time ”pre-LTE” MIMO demontrator. Because the entire system functionality is mainly defined by the software development the applicability of our test-bed is less limited to any special standard.

The lack of MIMO RF front-end with full functionality, enough flexibility and high performance is one main impedient standing between the software simulation and the real MIMO system. Our MORFAN RF front-end module, fully covering 2 × 2 MIMO RF chain, is trying to overcome this obstacle and brings developers freedom to implement real MIMO systems. With the highly integrated design developers don’t need to worry about RF design and the converting between analog signals and digital signals any more.

In this contribution we describe the software development enviornment, system constitution and some special considerations for real-time algorithm development as well as the architechture of our new MIMO RF front-end. Taking advantage of the software-defined property we achieve to realize the fully real-time MIMO system in one month. It indicates that our MIMO Testbed structure can accelerate the development and evaluation of the techniques for the next generation wireless communication systems.

REFERENCES

[1] A. Wilzeck, M. El-Hadidy, Q. Cai, M. Amelingmeyer, and T. Kaiser, “MIMO Prototyping Test-bed with Off-the-Shelf Plug-in Hardware ” in *International ITG - IEEE Workshop on Smart Antennas 2006 (WSA 2006)*, Ulm, Germany, March 13-14 2006.

[2] Q. Cai, A. Wilzeck and T. Kaiser, “Evaluation of Synchronization and Fractionally Spaced Equalization in a MIMO SC-FDE Test-Bed ” in *Proc. RWS 2007*, Long Beach, CA, USA, January 09-11. 2007, pp. 527-530.

[3] Sundance Multiprocessor Technology Ltd., UK, website: http://sundance.com

[4] http://www.3gpp.org/Highlights/LTE/LTE.htm

[5] Texas Instruments: TMS320C6414T, TMS320C6415T, TMS320C6416T Fixed-Point Digital Signal Processors (Rev. J).

[6] Texas Instruments: TMS320C6000 Optimizing Compiler User’s Guide.

[7] Texas Instruments: TMS320C6000 Assembly Language Tools User’s Guide.

[8] Texas Instruments: TMS320C64x DSP Library Programmer’s Reference.

[9] 3L Ltd.: Diamond User Guide (Sundance Edition V3.1).