MIMO Prototyping Using Sundance's Hardware and Software Products

Andreas Wilzeck, Matthias Amelingmeyer, and Thomas Kaiser
University Duisburg-Essen
Faculty of Engineering, Department of Communication Systems,
Bismarckstr. 81, 47057 Duisburg, Germany
Email: andreas.wilzeck@uni-duisburg-essen.de

Version December 9, 2005

Abstract

Future wireless communication systems require new approaches to enable higher data rates and/or larger coverage. Traditionally designed wireless communication systems employ diversity techniques in frequency, time and code, so that the remaining, mainly unused, physical dimension is space. To exploit this spatial dimension an array of antennas, either at the transmit-side, at the receive-side, or at both, is required. Such multi-antenna systems are generally called MIMO systems (Multiple-Input Multiple-Output). The MIMO technology is widely accepted as one of the few promising key technologies that will serve the demands of future wireless communication systems. The aim of this contribution is to provide an understanding of basic MIMO techniques and to introduce the concept, the setup, and the usage of our MIMO testbed based on Sundance's hardware and software platform in more detail.

Index Terms

Multiple-Input Multiple-Output, MIMO, Multiple Antenna, Communications, Testbed

I. Introduction

Current and future wireless communication system aim to provide high mobility, high coverage and high data rates for mobile multimedia applications.
Typical applications are:

- High-Quality Video Streaming (Video On-Demand),
- Enhanced Reality Multi-User Online Games,
- Video-Telephony (Point-2-Point or Conference Call), and
- High-speed wireless networks in home, in office and in outdoor environments.

Traditionally designed communication systems require additional signal bandwidth in frequency domain or must employ higher-order modulation techniques to fulfill the data rate demands of such applications. Additional signal bandwidth is an expensive commodity. Numerous parts of the frequency spectrum are already occupied by certain services, and licensed bands are usually expensive (recall the auctions of the UMTS frequency bands in Europe). Higher-order modulations (e.g. 128-QAM, 256-QAM, 512-QAM and above) suffer from technology limitations, because of very low-noise and high linearity requirements of analog components leading to expensive production costs.
The usage of multiple antennas at transmitter- and receiver-side in communication systems improves the communication link quality (range) and/or the data-rate of a communication link without occupying additional bandwidth in the frequency domain and still using commonly employed modulation techniques (e.g. BPSK, QPSK, 16-QAM, 64-QAM). Such systems are called Multiple-Input Multiple-Output (MIMO) systems.
Since more than a decade the MIMO technology is a so-called hot reasearch topic in communications. Today MIMO techniques are substantial part of emerging communication standards like IEEE802.11n and

3GPP/HSDPA. Recently several pre-standard products employing MIMO techniques entered the market for Wireless Local Area Networks (WLAN).
The fundamental research in MIMO communications is of theoretical nature, so its theoretical results, which are based on simplified models of reality, can be validated by employing dedicated testbeds and demonstrators for assumption-free testing. This allows deeper insight into the practical challenges of MIMO algorithms and systems. It also proves wheter an algorithm or approach is suitable for implementation and deployment in practice.
In this *application note* we briefly introduce the basics of MIMO communications and then focus on the concept, setup and usage of a MIMO testbed based on Sundance's hardware and software platform.

II. Basics of MIMO Communications

In general, high data rate wireless communication systems suffer from Inter-Symbol-Interference (ISI) and fading effects due to the multi-path propagation. Multi-path propagation is caused by reflections of the transmit signal for example by walls, buildings and mountains (Fig. 1). Therefore, the received signal is a superposition of multiple echoes of the originally transmitted signal. Either by training or by blind approaches, equalization techniques can be employed to reconstruct the transmitted signal.

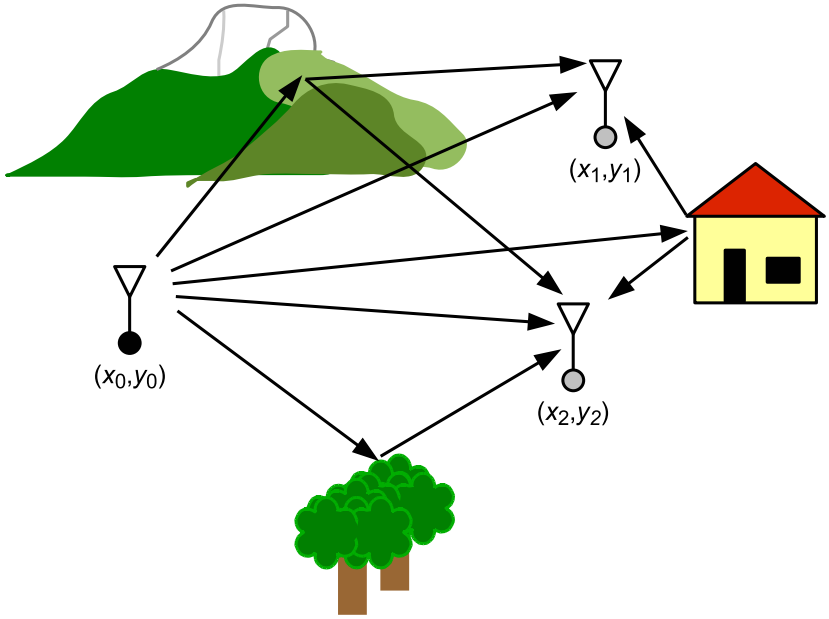

Fig. 1. Multi-path Propagation Scenario

Fading is a fluctuation of the received signal power. This effect can be divided into two groups: Long term fading is caused by shadowing of signal paths by objects (e.g. walls, mountains, buildings). Short term fading is caused by constructive or destructive superposition of the multi-path signals received, where movements of only a few centimeters are required to observe a significant change in the received signal power.
The MIMO technology makes use of the spatial characteristic of multi-path propagation and thereby also relaxes the fading effects.
The latter can be explained as follows: The signal transmitted by an antenna in position (x_0, y_0) is received by an antenna in position (x_1, y_1). The received signal is the superposition of all echo signals arising due to multi-path propagation. A receiver antenna being in position (x_2, y_2) will receive other echoes, because of other reflections. Therefore, a signal transmitted from position (x_1, y_1) or (x_2, y_2) will result in different responses at (x_0, y_0). From this we can conclude that a different spatial signature is induced to the received signal, which basically depends on the multi-path propagation arising between the transmit antenna and the receive antenna.

These different spatial signatures are emploited in a MIMO system (Fig. 2) by using multiple transmit and receive antennas. The aim is to improve the communication link quality and/or the data-rate of the communication link. Hence, a MIMO system needs multi-path propagation to make best use of its multiple antennas.

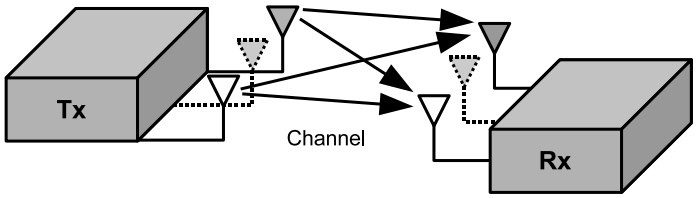

Fig. 2. MIMO System with N_{T} Transmit Antennas and N_{R} Receive Antennas

We will now introduce two basic MIMO concepts demonstrating the twofold use of the spatial signatures. *Spatial Multiplexing* (SM) is a MIMO technique for improving the data-rate. As shown in Fig. 3 a data stream with doubled data-rate is multiplexed to two transmit antennas, so that each stream in a single antenna branch shows the commonly used data-rate, for example 54 Mbps. Therefore, this system can provide overall 108 Mbps without occupying additional bandwidth. Note, that there are also non-MIMO products on the market doubling the bandwidth to provide 108 Mbps.
The SM receiver picks up at each of the two antennas a superposition of the two transmitted signals, so the main tasks of the receiver are to separate the transmitted streams and to equalize the signals. Basically spatial multiplexing can be seen as a virtual parallel cable in air.
The separation of the streams is feasible if the spatial signatures of both received transmit signals differ to some extent. This means that the channel impulse responses $\mathbf{h}_{q,p}$ between the p-th transmit antenna and the q-th receive antenna should be uncorrelated to each other. Note that in a communication system employing N_{T} transmit antennas and N_{R} receive antennas, N_{R} must be greater or equal to N_{T} in order to separate the transmitted streams.

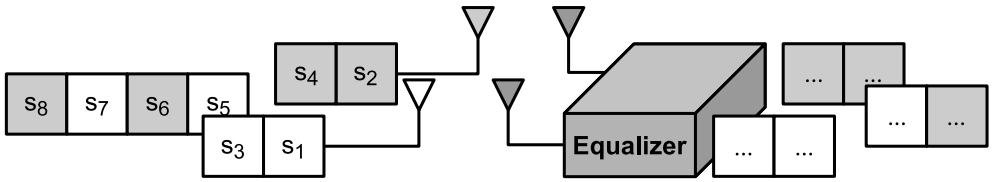

Fig. 3. Spatial Multiplexing in a MIMO System with $N_{\mathrm{T}} = 2$ Transmit Antennas and $N_{\mathrm{R}} = 2$ Receive Antennas

Opposite SM, *Space-Time Coding* (STC) employs a coding of the transmit data stream - in space and time - in order to overcome distortions by the fading channel. Fig. 4 presents the so-called Alamouti scheme, which uses $N_{\mathrm{T}} = 2$ transmit antennas and $N_{\mathrm{R}} = 1$ receive antennas. Each of the two transmit antennas is used to transmit a differently space-time coded signal. This is done in such a way that the stream coming from each antenna basically contains the same information. The main receiver tasks are to equalize the signals and to decode the transmitted data. If the spatial signatures of the received transmit signals are not equal, then they undergo a different fading in the channel. In the extreme case, one transmitted signal might be completely suppressed. Nevertheless, in such a case the receiver will still be able to reconstruct the transmitted data by exploiting the other transmit stream.
In MIMO communications more advanced techniques are developed in order to make the best use of the over-air channel. Of special interest are multi-user scenarios with certain channel knowledge at the transmitter gained from channel reciprocity (TDD mode) or by a separate pilot channel. Such knowledge can be used to reduce Multiple Access Interference (MAI) by spatial pre-filtering techniques (e.g. Space Division Multiple Access, beamforming). Other current topics are systems that adaptively combine STC and SM depending on the instantaneous channel characteristics.

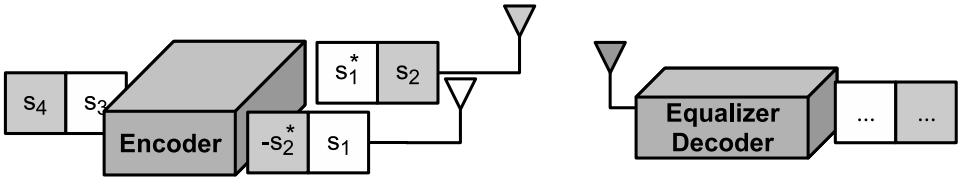

Fig. 4. Alamouti Scheme in a MISO System with $N_{\mathrm{T}} = 2$ Transmit Antennas and $N_{\mathrm{R}} = 1$ Receive Antenna

The interested reader should refer to the book [1] for a good introduction to MIMO communications. For an overview on the mainly European MIMO research activities see [4].

III. Concept

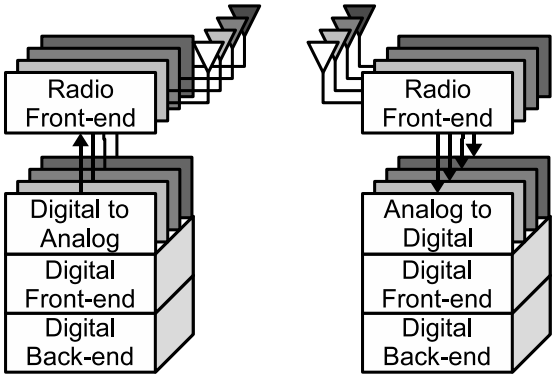

Fig. 5. Concept for a MIMO System with $N_{\mathrm{T}} = 4$ Transmit Antennas and $N_{\mathrm{R}} = 4$ Receive Antennas

In Fig. 5 the concept for a MIMO System with $N_{\mathrm{T}} = 4$ transmit antennas and $N_{\mathrm{R}} = 4$ receive antennas is presented. It is organized in four major horizontal stages.
Starting at transmitter-side (left) the digital back-end processing stage is taking care of coding, interleaving, mapping of bits to complex symbols and other base-band processing tasks. The digital front-end processing stage is used for higher rate processing like I/Q modulation to an Intermediate Frequency (IF), filtering, and other tasks that require high processing power and must be done in parallel for a MIMO system. Afterwards the Digital-to-Analog stage converts the digital signals into analog signals. The Radio Frequency (RF) front-ends up-convert the analog signals from IF to RF (e.g. 2.4 GHz) and also perform some analog processing like power amplification, filtering and pre-distortion. The RF signals are sent by the transmit antennas.
At receiver-side (right) the antennas pick up the transmitted signals. The RF front-ends are used for down-converting the signals to IF and also contain filtering, gain control and low-noise amplification of the received signals. In the Analog-to-Digital stage the analog signals are sampled with an appropriate sampling rate. The digital front-end processing stage is used for higher rate processing like carrier frequency offset estimation and correction, I/Q demodulation to base-band, filtering and other parallel higher rate tasks like time synchronization. In the digital back-end the typical MIMO base-band processing for equalization, de-mapping, de-interleaving and decoding takes place in order to recover the transmitted bit stream.
One can note from this concept, that MIMO technology requests a large amount of high-rate parallel processing in the digital front-ends. This parallel processing and the required high data-rate only allow using a Field Programmable Gate Array (FPGA) for the digital front-ends. After processing in the FPGA, a Digital Signal Processor (DSP) might be used for back-end processing at lower data-rates. Nevertheless back-end processing units like the MIMO equalizer and decoder (e.g. Viterbi Decoder) may especially overstrain the power of a DSP, so that they are required to be implemented in FPGA as well.

During development of a wireless communication system, the partitioning of an algorithm (implementing on FPGA or DSP) and the verification of the final real-time implementation of an algorithm in a system is required. For the mathematical algorithm development and for studying the behavior using abstract models Mathwork's MATLAB Suite is typically used. Afterwards the implementation of an algorithm is evaluated typically using the C/C++ programming language, and is implemented for the first time on a DSP. Once it turns out that this algorithm is valid for implementation, an FPGA implementation using VHDL or Verilog becomes feasible.
The testbed should assist this development chain as much as possible, so that the testbed and the algorithms are developed side-by-side. This is facilitated with the help of a modular platform that allows to mitigate step-wise to a full real-time system and to enhance the system, if for example more processing power is required. It is therefore reasonable to develop the testbed in phases like those shown in Fig. 6.

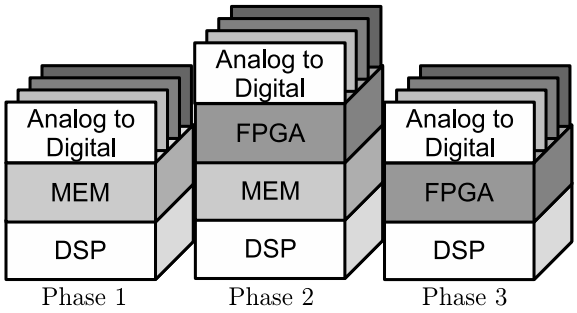

Fig. 6. Development Phases for the MIMO Receiver; Phase 1: Offline, Phase 2: Hybrid, Phase 3: Online

Phase 1 is the so-called "offline-mode". It allows the real over-air transmission chain to be used in order to understand the channel effects and the effects induced by the digital-to-analog conversion, the RF hardware and the analog-to-digital conversion. Pre-processed data is transmitted over-air from the testbed's memory and the data coming from the analog-to-digital converters is stored to memory for later analysis on the host PC. The user can deploy his already developed simulation chains by replacing his channel model with the channel offered by the testbed. Note that this channel is the real over-air channel, which includes all the hardware effects. During the development and the evaluation of an algorithm the offline-mode is especially reasonable, because it allows for repeated testing on real data. Such an offline-mode also allows for higher sampling rates, whereas a real-time system has to limit the sampling-rate and data-rates within the system to a manageable amount.
Phase 2 is the so-called "hybrid-mode". Parts of the system are already operating in real-time (e.g. the digital front-end), whereas the others are still part of offline processing. The data logging ability can also be employed here for repeated testing of a digital front-end implementation and detailed analysis.
Phase 3 is the final "online-mode" or real-time system, where the whole system is running in real-time. The testbed is now a demonstrator or prototype of a wireless communication system. It is now possible to study the entire behavior of the system, the interoperation with other systems, throughput studies and long-term studies.

IV. MIMO TESTBED

Our first MIMO testbed STARS[1] was developed starting in summer 2003. It was finally exhibited at the European Signal Processing Conference (EUSIPCO) in September 2004 showing a SIMO OFDM transmission. STARS[1] is based on Sundance's modular digital signal processing platform, which offers a rich set of TI DSP modules, Xilinx FPGA modules and Analog-to-Digital (A/D) & Digital-to-Analog (D/A) modules [2]. In order to transform the base-band signals to the radio frequency of 2.4 GHz, we incorporated evaluation boards of the WLAN transceiver chips (AT86RF240) of ATMEL. A 3-wire bus between DSP and transceiver chip was employed for control purposes. It is required to distribute a common

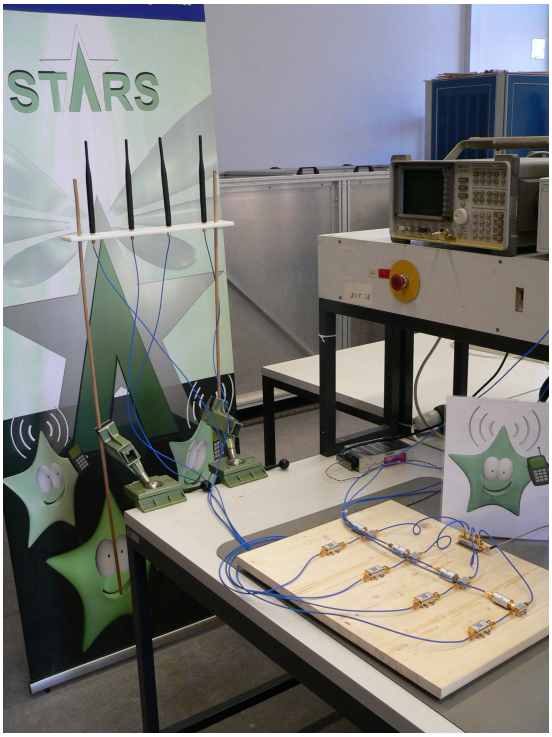

Fig. 7. Pre-Version of the Radio Frequency (RF) MIMO Front-End for the 4-Antenna Receiver of STARS^2.

reference clock between the two transmitter-boards and also between the two receiver boards to form a 2×2 MIMO System. Note, that we did not deploy a common clock between transmitter and receiver, as usual in practical wireless systems.
The Digital-to-Analog and Analog-to-Digital conversion rates are fixed at 40 MHz and the amount of memory for storing snapshots is limited to 128 MBytes (64 M samples) per receive antenna. The used Zero-IF architecture of the RF modules introduces challenges due to DC offsets and to I/Q imbalances, but the major drawback is the required dual channel ADC per antenna, which occupies additional resources and space. The detailed technical properties, abilities and concepts of STARS and of the here described STARS^1 demonstrator can be found in [3] and [4].
STARS^2 (see Fig. 7 and Fig. 11) aims to be a more powerful and more universal tool than it was possible for STARS^1.
It provides a memory that is four times larger (256 M words per antenna) for storing snapshots at high sampling rates (up to 100 MHz) and much faster data-transfer rates to the host. The overall hardware architecture is simplified by allowing the sampling at a Low-IF, which eliminates the DC offsets and the I/Q imbalances in the antenna branches, because the conversion to base-band is now done digitally.
Beside first multi-user tryouts, by employing a 4-antenna receiver and two 2-antenna transmitters, STARS^2 aims to study MIMO front-end effects and architectures in more detail. This is assisted by the free choice of potential high sampling frequencies. Like STARS^1 it is based on Sundance's modular digital signal processing platform, which allows us to extend the system and to employ partial or full real-time processing at later stages of our research projects as previously described.
In the following we describe the hardware setup of STARS^2 and present the software flow of the transmitter and receiver application. Our contribution in [5] reviews all MIMO testbeds developed in our department, including also our educational XIXO Audio Channel Testbed (eXACT), which employs audio signaling.

A. MIMO System Setup

The system setup of the STARS^2 system is based on the following Sundance modules:

- *SMT310Q:* The SMT310Q carrier board features four sites for Texas Instrument Modules (TIM) as well as the connection to the host PC via PCI bus,
- *SMT365_8_2:* The SMT365_8_2 is a DSP/FPGA module. The DSP is a TMSC320C6416 running at 600 MHz. The Field Programmable Gate Array (FPGA) is a Xilinx Virtex II, which is exploited to implement fast communication (e.g. SHB, ComPort). Additionally the module is populated with 8 MBytes of zero bus turnaround RAM (ZBTRAM).
- *SMT351-G:* The SMT351 memory module offers 1 GByte (256 M$\times$32bit) of RAM memory, which can be used as independent buffers to log or playback data on the SHB bus. The maximum data rate for write and read procedures is up to 400 MBytes/s.
- *SMT370:* The SMT370 provides a 2-Channel Digital-to-Analog-Converter (DAC) with up to 160 MSPS and a 2-Channel Analog-to-Digital-Converter (ADC) with a maximum sampling frequency of 105 MHz.

For an easy access, the DSP's General Purpose Input/Output (GPIO) Pin 15 and the SMT370's connectors for the external triggers and external clocks are available at the PC's back side in SMA.

1) Transmitter: The hardware setup for the MIMO transmitter is presented in Fig. 8. Basically it implements a signal generator with two output channels IF 1,2.
The setup involves one SMT310Q to carry the two modules, one SMT370 for 2-channel DAC, and one SMT365_8_2 for control and host communication.
The connection named with SHB in Fig. 8 represents the 32-bit wide Sundance High-Speed Bus (SHB). It interconnects the SMT365_8_2 module with the SMT370 and allows maximum data rates of 400 MByte/s. The connection labeled ComPorts serves as an 8-bit wide Communication Port to send commands for configuration from the SMT365_8_2 to the SMT370 or vice versa. In this setup the SMT370 expect its configuration on ComPort 0. The SMT365_8_2's ComPort 4 is therefore interconnected with it.
In addition, the following external connections are provided:

- SMT365_8_2 provides a trigger output using one of its GPIO pins,
- SMT370 features an external trigger input,
- SMT370 provides an external clock input,
- SMT370 provides two analog output channels (labeled with IF 1,2).

The analog output channels can be used to transmit an Intermediate Frequency (IF) of around 15 to 20 MHz, so a 3-dB signal bandwidth of 5 to 20 MHz is possible.

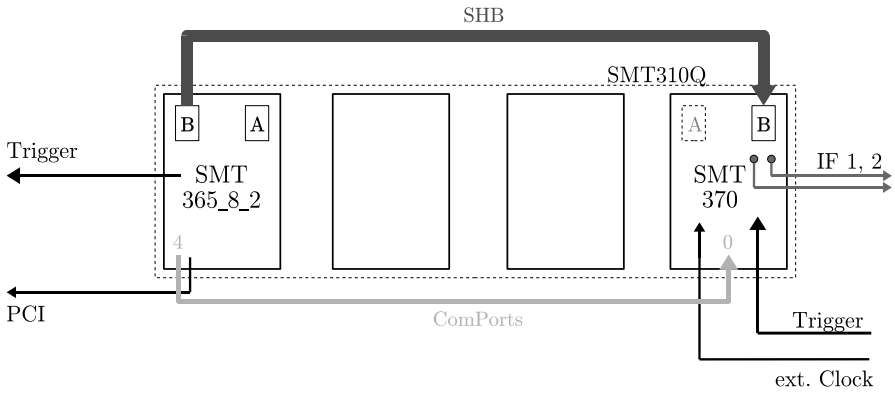

Fig. 8. Block Diagram of Transmitter's Digital Part with Two Antennas.

2) Receiver: The MIMO receiver is presented in Fig. 9 and Fig. 10. Basically the receiver is a 4-channel data logging unit providing an adjustable sampling rate up to 100 MHz and a 4 $\times$ 512 MByte deep memory.

In Fig. 9 the interconnection of ComPorts for configuration tasks is sketched. The setup involves two SMT310Qs to carry the five modules, two SMT370 for 4-channel ADC, two SMT351-G for data logging, and one SMT365_8_2 for control and host communication.
The Fig. 10 shows the four analog inputs (IF 1,2,3,4) and the SHB interconnection of the modules.
The following external connections are provided:

- SMT365_8_2 provides a trigger output using one of its GPIO pins,
- Each of the SMT370s feature an external trigger input,
- Each of the SMT370s provide an external clock input,
- Each of the SMT370s provide two analog in channels (overall four channels, labeled with IF 1,2,3,4).

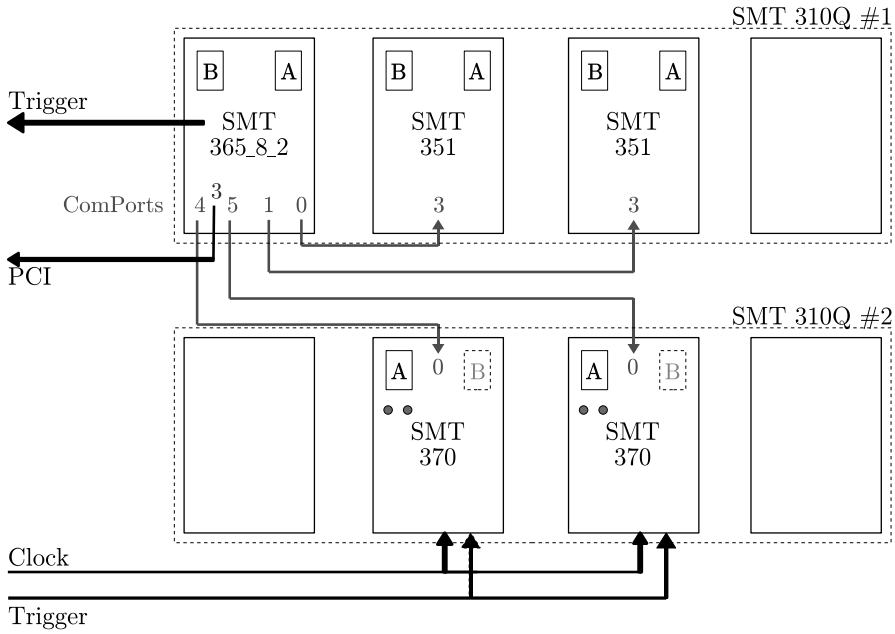

Fig. 9. Block Diagram of the Receiver: Trigger, ComPorts.

As shown in Fig. 9 the SMT365_8_2 is connected via the ComPorts 0, 1, 4 and 5 to all other modules. SMT365_8_2's ComPort 3 is reserved for interaction with the host PC (via the PCI bus), which can be contacted by the first SMT310Q board only.
Fig. 10 shows the setup of the digital part of the receiver, the SHB connections and the four IF input channels. Each of the TIM modules offer two SHB connectors, labeled **A** and **B**. The analog signals on the four input channels IF 1,2 and IF 3,4 are converted to digital data in the SMT370 modules. Both SMT370s transmit the digital data using their SHB **A** connectors, these are connected to the corresponding SHB **A** connectors of the SMT351s. For reading back the data, both SHB connectors of the SMT365_8_2 are necessary to receive the digital data from the two SHB **B** connectors of the SMT351s.
Fig. 11 delivers insight into the ADC part and digital processing hardware employed in the receiving host PC. As one can notice it is possible in the current setup to deploy some additional modules for real-time processing or two additional receive antennas, e.g. one additional SMT370 & SMT351 pair and one additional DSP module.

3) Synchronization of ADCs/DACs: Because of the use of two SMT370 modules and two SMT351 memory modules in the receiver, a synchronized operation must be guaranteed. The two channels of ADC or DAC of a single SMT370 can be easily made synchronous by standard configuration. In the case where two or more SMT370s are involved, a common external clock and a simultaneous start and stop of data conversion for all SMT370 ADCs or DACs is required. The start/stop event is provided by an external trigger given by the SMT365_8_2. Therefore, the GPIO-Pin 15 of the SMT365_8_2 is employed.

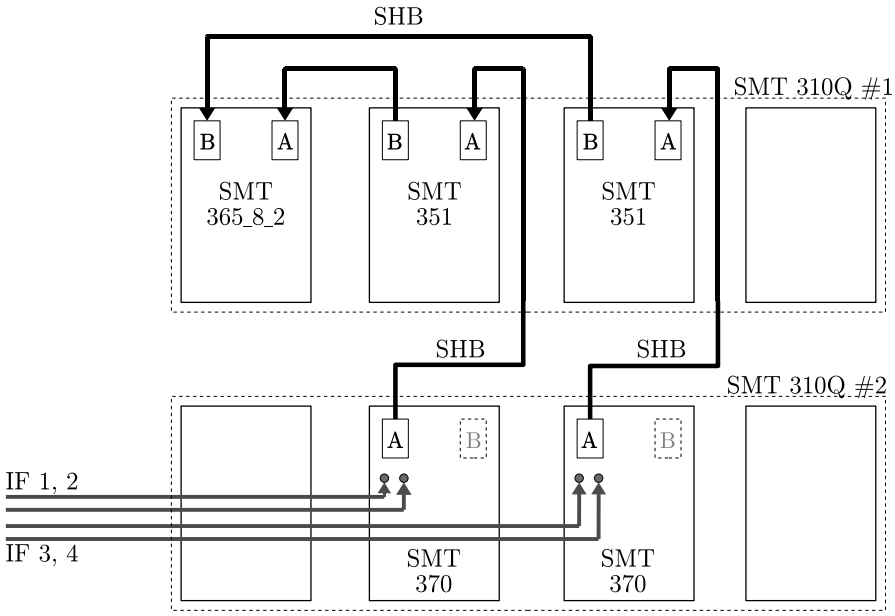

Fig. 10. Block Diagram of the Receiver: Input Signals, SHB.

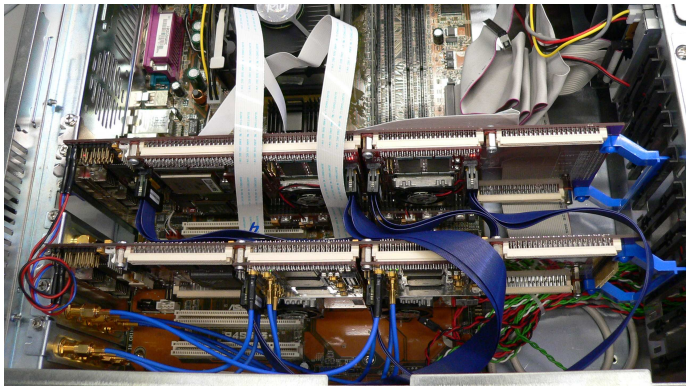

Fig. 11. A Closer Look into the Receiver Host PC.

The GPIO pin 15 is programmed in such a way that it can possess a ’high’ and a ’low’ LVTTL state. ’High’ means a voltage of 3.3 Volts, ’low’ means a voltage of 0 Volts.

B. Software

The whole software is developed using 3L Diamond 3.0 and TI’s Compiler Tools within Code Composer Studio (CCS) 3.0. 3L Diamond is a multi-DSP Real-Time Operating System (RTOS), which makes advanced C-based software development with Sundance C6x-based modules quite easy. It delivers built-in drivers and functions for inter-processor and inter-module communication, and allows the DSPs in the DSP access to host services and C standard I/O functions. For running a 3L Diamond-based program (.app) on a host PC, the software 3L Diamond Server V3.0 is necessary. The server is a Windows program running on the host computer. This server is used to first load a C6x executable file from the host onto the target board and it handles I/O requests from the DSP system.
In the following we will explain the program flow of the transmitter and receiver application.

1) Software Flow Chart of Transmitter: Fig. 12 represents the program flow of the transmitter application. In the beginning of the program a declaration of variables is necessary and the trigger is set to the defined

’low’ state. The DSP claims the SHB and set it in transmitter mode in order to be able to transfer the digital data to the SMT370’s DAC or on-board memory.
The SMT370 configuration is stored in a configuration file, which is generated by a GUI-based external application. The DSP application uses the standard I/O functions to virtually read this file from the host PC’s harddrive. The configuration is stored temporarily and then it is written across the corresponding ComPort to the SMT370 under configuration.
In our setup we make use of the pattern generator mode of the SMT370.
When loading data into SMT370’s on-board memory for pattern generator mode use, one has to make sure that a Start operation is not selected, i.e. the pattern generator must be stopped first.
In the next step, the host PC has to read an existing binary data file which consists of the signals to be transmitted. These samples are stored in a previously allocated memory space. Its data size is essential for reserving the correct amount of memory in the SMT370’s on-board ZBTRAM. After all this is done, the previously loaded samples can be sent from the DSP module to the DAC module across the SHB. Once the transfer is completed, the pattern generator can be started by using a ’Start’ command.
The DSP sets the trigger signal to a ’high’ state, which is connected to the external trigger input of the DAC. All DACs start to generate a periodic signal simultaneously. Driving the trigger to a ’low’ state will stop the DACs.

2) Software Flow Chart of Receiver: In the following the 4-channel receiver application is presented. For convenience the software flow chart is split into four stages

- Configuration and Setup (Fig. 13),
- Trigger Control (Fig. 14),
- Data Logging (Fig. 15), and
- Read Back and Storage on Hard Disk (Fig. 16).

In the first stage the configuration and setup (Fig. 13) of the receiver is described. After a declaration of variables it is necessary to claim both SHBs of the SMT365_8_2 TIM module as receivers. Then the DSP accesses the configuration file on the hard disk to set up the Analog-to-Digital-Converters (ADC) on-board the SMT370 module. To stop any data logging, the trigger is set in a well-defined ’low’ state. The SMT351 memory module has to be configured by loading firmware into its FPGA.
The DSP sends a command via a ComPort to the SMT370’s ADCs to get ready for Data Acquisition, but the ADCs will not start running until the trigger is in ’high’ state. The memory modules (SMT351) receive a capture signal from the DSP, which includes the information about how much data has to be stored. Note, that this capture signal is given before the ADCs start outputting data, so the trigger is still in a ’low’ state. Therefore, the SMT351 will wait for data.
We will explain this in more detail.
Fig. 14 represents the trigger control to start and stop the Data Acquisition (DAQ). When the configuration is finished, the ADCs await a ’high’ trigger signal to start the DAQ. Therefore, the DSP has to set the trigger in a ’high’ state by user command, which is received by the SMT370 and then all ADCs start logging data synchronously. To stop DAQ of all ADCs at the same time the trigger is pulled to a ’low’ state.
As shown in Fig. 15 the storage of data is performed at the same time as the analog-to-digital conversion. The SMT370 delivers data via the SHB to the SMT351. Before the ADCs can transmit their output data to memory, a control signal to request the SHB has to be sent by the SMT370, which is acknowledged by the SMT351. This is the Write ENable (WEN) control signal. After pulling the WEN signal to a ’low’ state, the SMT370 can deliver data over an SHB to the SMT351, which receives and stores the data.
As shown in Fig. 16 the DSP reads the data captured by the SMT351 memory module and stores it into two files on a hard disk (one file per SMT370). To do so, the DAQ is stopped by pulling the trigger to a ’low’ state and a play back control command is sent from the DSP to the memory module over the ComPort.

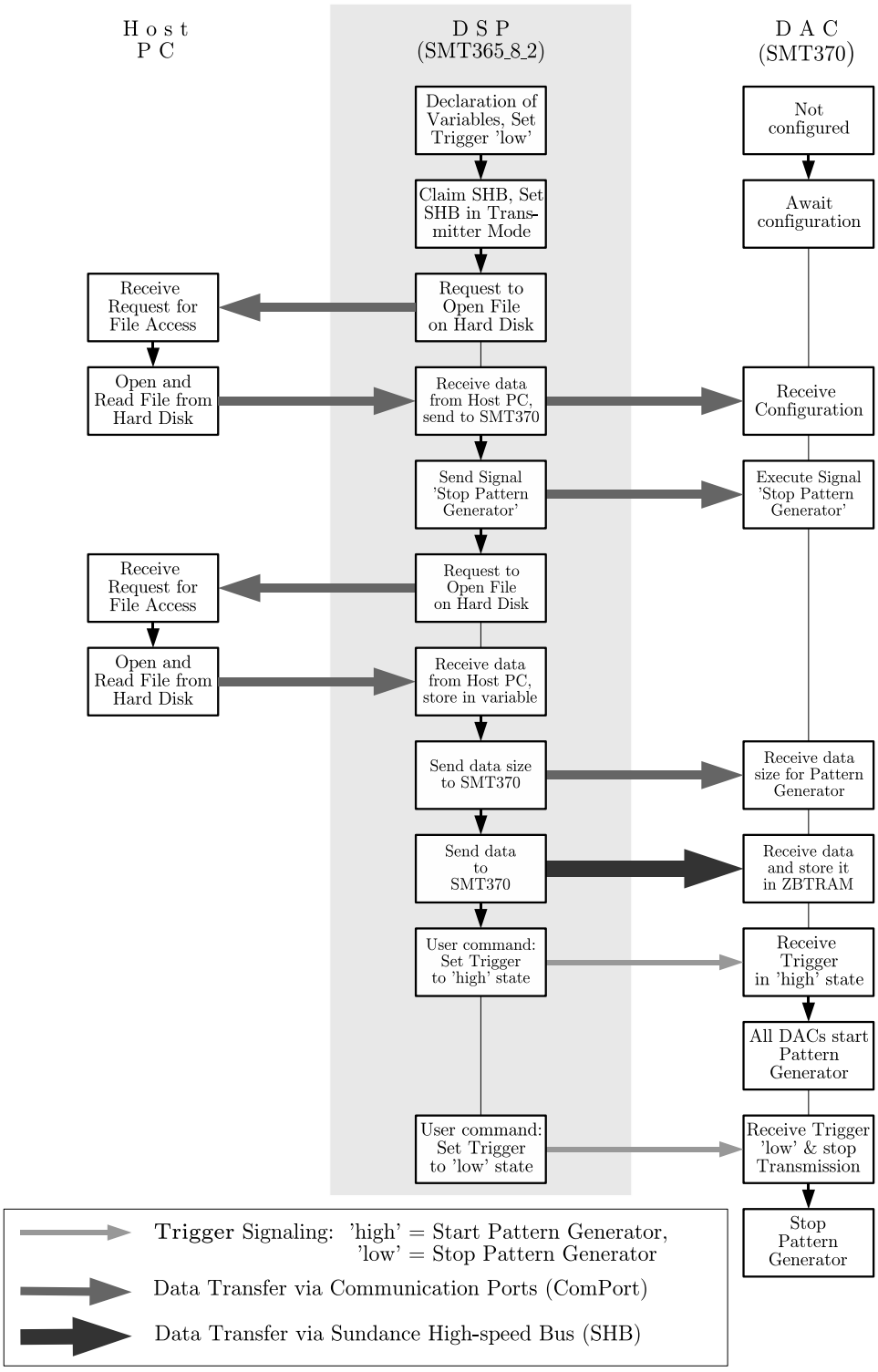

Fig. 12. Block Diagram of Transmitter's Digital Part with Two Antennas.

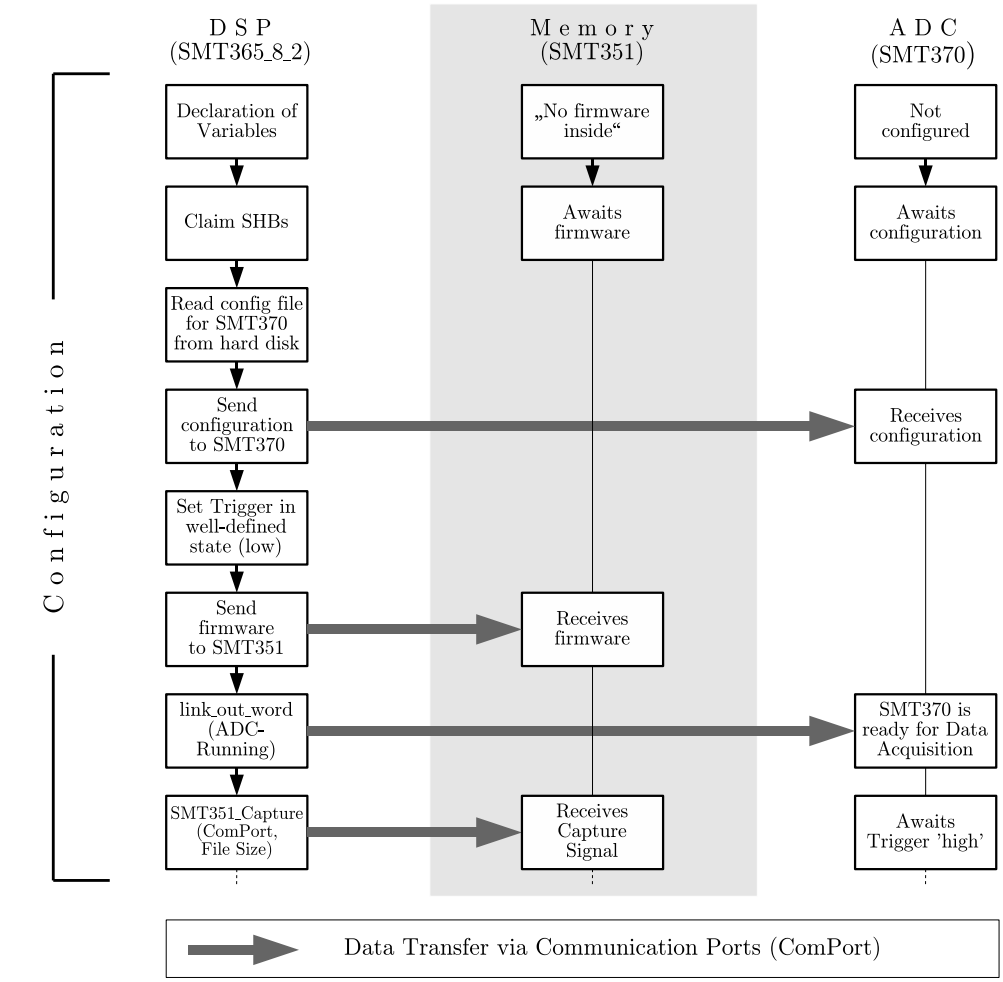

Fig. 13. Configuration and Setup of SMT370 and SMT351.

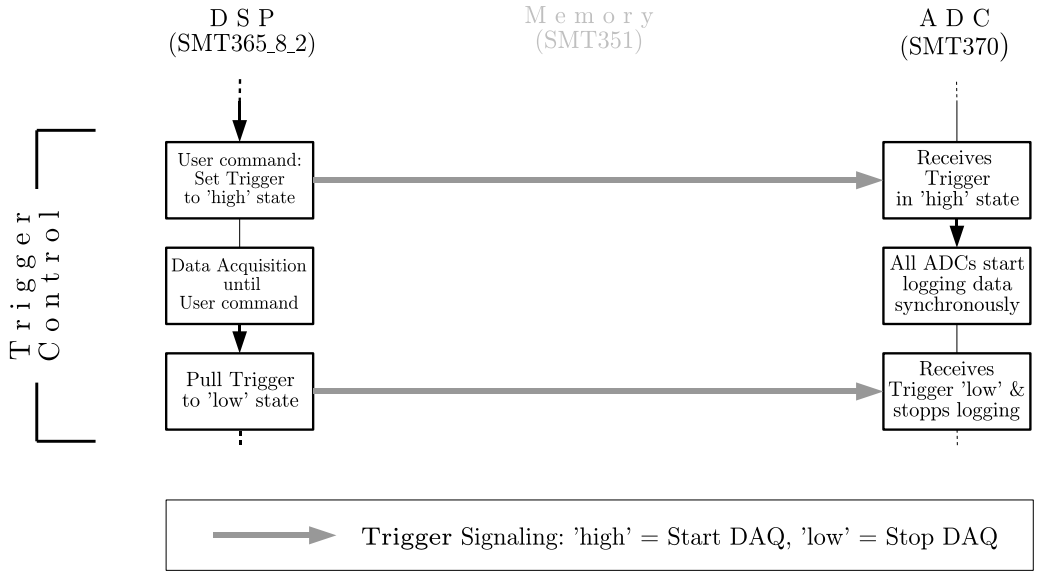

Fig. 14. Trigger Control for Synchronous Start and Stop of Data Acquisition.

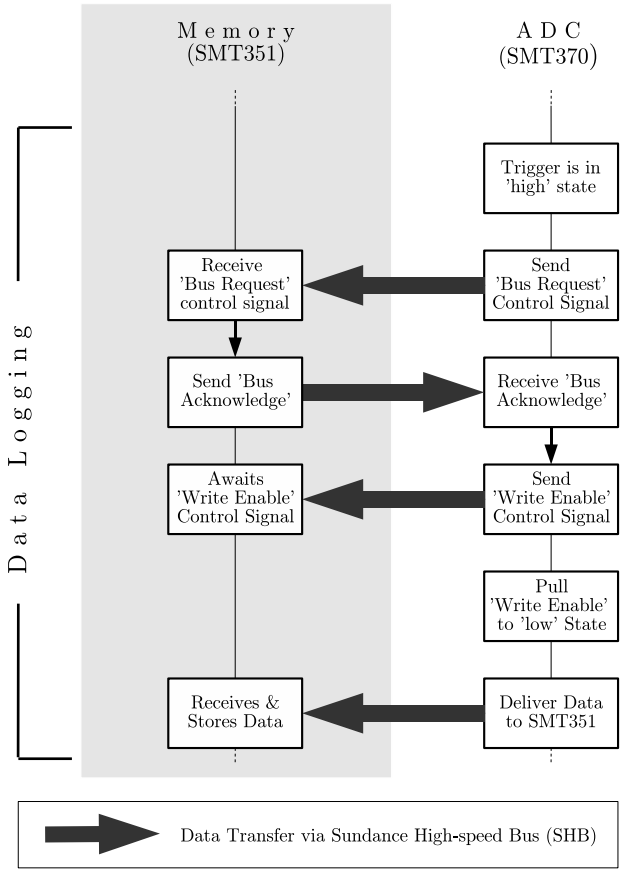

Fig. 15. SHB Control Signaling and Data Logging.

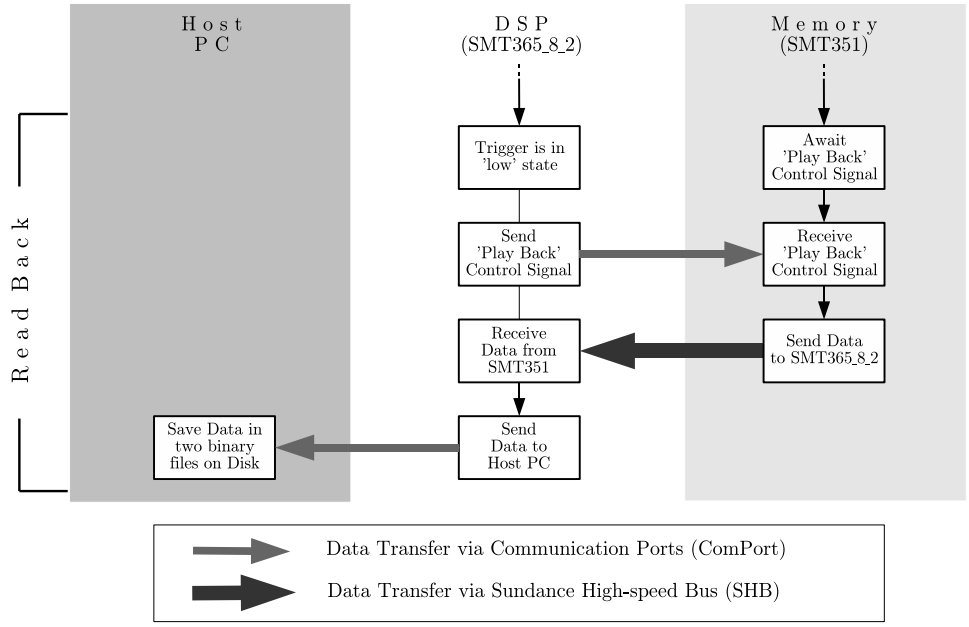

Fig. 16. Read Back and Storage on Hard Disk.

C. MIMO Radio Frequency Hardware

The STARS[1] testbed employs evaluation boards of recent commercial wireless transceiver-chips [4], which can be seen as more practical in terms of latter MIMO products. Nevertheless it reduces the possibility of influencing the behavior of the RF hardware, which is restricted to the requirements of specific wireless communication standards.
The new STARS[2] uses a discrete and modular setup based on plug-in RF building blocks to form a MIMO RF front-end, so that each component can be exchanged easily. Fig. 7 shows a pre-version of this 4-antenna RF front-end. The 4-antenna array can be seen in the upper left corner of the figure.

V. Outlook

After intensive study of algorithms our final aim is a fully or partly real-time solution of a Multi-user wireless MIMO communication system, which also involves feedback techniques or cooperation between the users. Sundance's Simulink tools combined with the Xilinx System Generator may be used to reduce the development time noticeably.
This application note will be updated from time to time to reflect the current state of our testbed and projects around.

References

[1] A. Paulraj, R. Nabar and D. Gore: "Introduction to Space-Time Wireless Communications", Cambridge University Press, 2003.
[2] Sundance Multiprocessor Technology Ltd., website: www.sundance.com.
[3] A. Wilzeck, T. Kaiser, R. Tempel, "A Modular Multi-User MIMO Testbed" (invited), Session on MIMO Implementation Issues, IEEE Radio and Wireless Conference 2004, Atlanta, USA, Sept. 19-22, 2004.
[4] T. Kaiser, A. Wilzeck, M. Berentsen, X. Peng, L. Haering, S. Bieder, D. Omoke, A. Camargo, A. Kani, R. Tempel and F. Ancona: "A MIMO Platform for Research and Education", in the book "Smart Antennas in Europe - State of the Art", EURASIP Book Series, 2005.
[5] A. Wilzeck, M. Amelingmeyer, T. Kaiser: "A Pre-Filtered MIMO Single-Carrier System with Frequency Domain Equalization and Its Prototyping Platform", RadioTecC, Berlin, Germany, October 26-27, 2005.