FPGA Implementation of a MIMO Receiver Front-end for UMTS

A. Burg, M. Rupp, M. Guillaud, E. Beck, D. Perels, N. Felber, W. Fichtner

Multiple input/multiple output (MIMO) systems have received great attention in the past years as a means to boost capacity of wireless communication systems. Multiple antennas at the transmitter and receiver are used to exploit the spatial diversity in a rich scattering environment to concurrently transmit multiple data streams in the same frequency band thereby increasing spectral efficiency. Besides the actual MIMO decoder a significant part of the implementation complexity of such a system turns out to be in the receiver front-end. In this paper an efficient implementation of a UMTS-based MIMO receiver front-end based on a new method for the efficient realization of a MIMO channel estimation is presented. We also discuss the implementation of the RAKE receiver and the frequency-offset estimation.

Channel estimation, MIMO, UMTS, Implementation

I. INTRODUCTION

Recently a number of schemes have emerged to boost the capacity and performance of narrow-band [1] and broad-band wireless communication systems through the use of multiple antennas at the transmitter and/or at the receiver side. The usage of more than one antenna at either side of the Tx/Rx chain improves the system performance mainly by mitigating the effects of fading, as it provides multiple mostly independent transmission paths through spatial diversity at a relatively low additional complexity. However except for a gain in average SNR these systems do not provide an immediate solution to increase the throughput/data rate. With multiple transmit(N) and receive(M) antennas as shown in the NxM system in Figure 1, multiple data streams can exist in the same frequency band at the same time and thereby increase the systems spectral efficiency. Such a system opens $M \cdot N$ channels which under ideal conditions are independent. Rich scattering environments tend to work in favor of such conditions, allowing the streams to be separated at the receiver through the inherent spatial diversity inferred in the channel. The capacity of such a system with $M = N$ is bounded by:

$$N \cdot \log_2\left(1 + \frac{SNR}{N}\right)$$

For a large number of transmit/receive antennas this leads to an almost N-fold gain in spectral efficiency under ideal conditions.

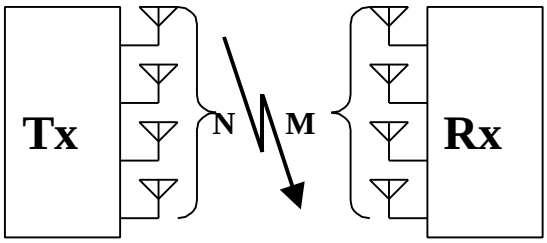

Figure 1: MIMO-System using multiple antennas at the transmitter and at the receiver

The receiver can be partitioned into a front-end and a back-end, where the front-end delivers soft symbols and channel information to the backend, which is responsible for separating the parallel channels through spatial diversity.
In this paper, the efficient hardware implementation of the critical components of a UMTS-based wide-band CDMA 4x4 MIMO-receiver front-end is described. In the next section a brief overview of the entire system is presented with a short introduction of how it is integrated within the UMTS standards. Section three focuses on the efficient implementation of the 4x4 channel estimation, which is the most complex part of the receiver front-end. In this section a new scheme to efficiently perform pilot assisted channel estimation of multiple channels in UMTS is presented. This is the main contribution of this paper. Section four describes the implementation of the RAKE receiver and of the finger search/assignment mechanism. Section five describes the frequency-offset estimation. Section six gives a brief overview of the actual system integration and gives some numbers from our initial implementation of the design in a XILINX Virtex FPGA. The last section concludes this paper with a brief summary.

II. A UMTS BASED MIMO SYSTEM

This section gives an overview of our 4x4 UMTS downlink MIMO system and briefly describes how it is integrated within the UMTS standards. The essential channels of a

A. Burg, D. Perels, N. Felber, W. Fichtner: Integrated Systems Laboratory, ETH-Zurich, apburg,perels,felber,fw@iis.ee.ethz.ch
M. Rupp: University of Vienna, mrupp@nt.tuwien.ac.at
M. Guillaud: Institure EURECOM, France maxime.guillaud@eurecom.fr
E. Beck: Lucent, Wireless Research, ericbeck@lucent.com

regular UMTS system are a primary and secondary sync channel (P/SSCH) for initial synchronization, a dedicated pilot channel (DPCH) and a data channel for each user. So called orthogonal variable spreading factor OVSF-codes derived from a Hadamard matrix are used to separate the channels utilizing code division multiple access (CDMA). The pilot and data channels are each spread with a unique channelization code that is orthogonal to the other codes in the system. The channels are then added together and once more spread with a PN-sequence. The two sync channels are finally added (they are not orthogonal to the other channels) before the signal is transmitted. This procedure is summarized in Figure 2. Extending the transmitter to a 4x4 system with MIMO capability is straightforward. The incoming data stream is demultiplexed into 4 parallel streams. These are spread with identical channelization codes and transmitted, each one on a separate antenna. However, in order to be able to estimate the channels between each transmit and each receive antenna new pilot channels need to be added, one for each transmit antenna. The pilot channels use OVSF code sequences with a length of 256 chips and transmit continuously the symbol (1,1).

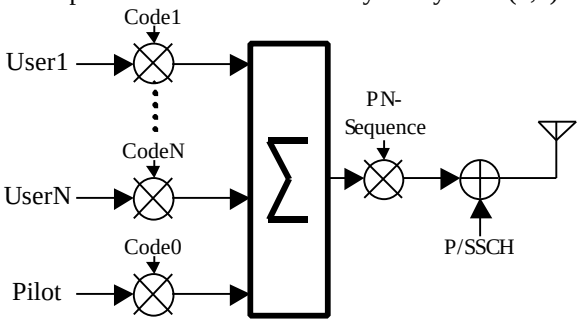

Figure 2 Channelization/Spreading in 1x1 UMTS

The block diagram of the MIMO receiver front-end is depicted in Figure 3. The received signal from each antenna enters the front-end at a digital IF. As a first step it is down-converted (DDC) to baseband. Subsequently the signal levels are adjusted through automatic gain control (AGC), frequency offset from the RF oscillators is removed and the standard UMTS matched filtering with a root-raised cosine filter (RRCos) is applied. The four incoming streams are oversampled at four times the chip frequency. The input signal of one antenna is fed to a synchronization block that performs the cell search and the initial synchronization of the local time reference with the base transceiver station (BTS). This block is not explained in further detail in this paper, however it should be noted that it contains a significant amount of complexity. For the actual decoding all signals are fed to the channel estimation block and to a RAKE receiver. The position of each RAKE finger is constantly determined from the result of the channel estimation. The soft symbols as well as channel coefficients at the finger positions are sent to the actual MIMO decoder [4] through a FIFO, which also provides storage to align the soft symbols with the corresponding channel estimates to improve the MIMO decoders performance in a relatively fast changing environment.

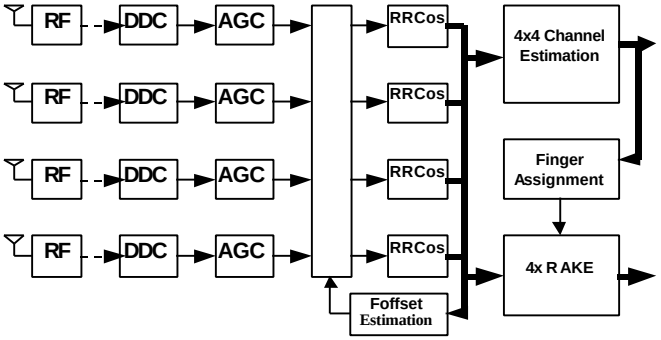

Figure 3 MIMO Frontend Overview

III. CHANNEL ESTIMATION

A. Channel estimation for a single chanel

For the channel estimation the receiver applies matched filtering of the received signal $x_j(t)$ at antenna j with the respective pilot sequence. For all $M \cdot N$ channels this estimation needs to be performed for all received signals and for all pilot codes transmitted from the different transmit antennas. In a 4x4 system this requires a total of 16 channel estimators to be implemented, making area optimization on this part of the receiver a critical issue. The next sections will show two standard methods to implement a single matched filter for the estimation of a 1x1 channel which would then be replicated 16 times to estimate all channels.

Throughout the rest of this section the following basic assumptions will be made:

1. The training sequences are of the form: $S_i(n) = S_c(n) \cdot C_i(n)$ where $C(n)$ are the periodic OVSF-Codes of the i[th] pilot channel with a period of K chips and $S_c(n)$ is the base-station's PN-sequence.
2. The same PN-sequence is used for all pilot signals, while they differ in their OVSF codes.
3. $h_{ji}(n)$ is the sampled impulse response of the channel from transmit antenna i to the receive antenna j with a length of L, so that $h(n > L) \equiv 0$.
4. The length of the channel is short compared to the length of the training sequence. For UMTS this assumption is reasonable, as the pilot OVSF-code has a length of 256 chips, which is equivalent to 64µs, while the impulse response of a typical (urban-) channel rarely exceeds 10µs (40 chips).

The matched filtering is described by the equation:

$$h_{ji}(n) = \sum_{t=0}^{K-1} x_j(n+t) \cdot S^*(t) \qquad (1)$$

1) FIR implementation

A simple direct form or transpose form FIR structure as shown in Figure 4 with the filter coefficients $S_i(n)$ gives the simplest implementation for the channel estimation. The length of the channel's impulse response (IR) is basically limited by the period of the training sequence, which according to our third assumption is substantially larger than

required. The length of the training sequence defines the length of the FIR filter and is fixed in the design. Even though for a binary training sequence the multipliers are degraded to simple adders this structure has a very high complexity, due to the high number of filter taps that are needed for long training sequences. Also the power consumption is extremely high, as the received signal propagates through the filter's delay chain. The resulting high toggling activity as well as the high load on the clock net due to the huge number of registers itself leads to excessive power consumption.

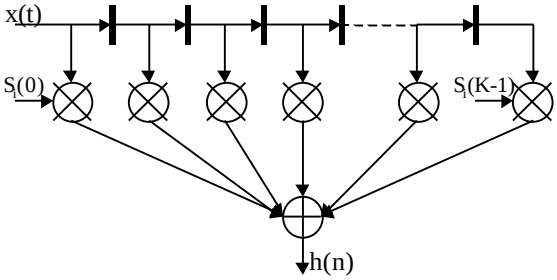

Figure 4 FIR Implementation

2) Correlator implementation

An alternative approach to the FIR implementation is a bank of correlators, each correlator being responsible for estimating one single tap of the channel's impulse response. Their number therefore depends on the length of the impulse response of the channel and is (according to assumption three) much lower than the number of filter taps required for the FIR structure. The complexity of each correlator, however, is only slightly higher than the complexity of one of the FIR taps. In this implementation the length of the channel's IR is fixed in the design, while the length of the training sequence is not predefined and can be easily adjusted at any time within the limits of the mantissa length of the adders and registers. This is an important capability for mobile systems. Longer averaging at low speeds allows a more precise estimation of the current channel condition, while higher speeds resulting in a faster changing channel require faster channel updates and allow less averaging. The delay line in Figure 5 has only the length of the channel's IR and each set of registers is only 2 bits wide to hold the binary complex value of the training sequence and its contribution to the overall complexity can therefore be neglected. Again the multipliers in the schematic are only symbolic and will be substituted by simple add/sub structures.

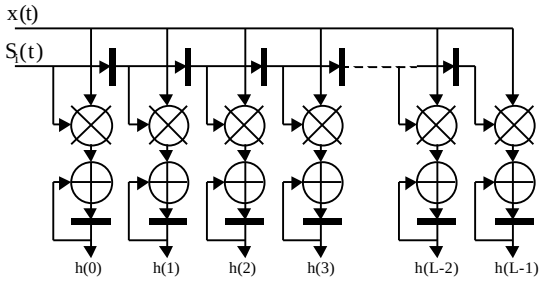

Figure 5 Correlator Implementation

3) Hybrid Correlator/FIR implementation

The two previously outlined approaches can be combined in a hybrid design. This will allow replacing most of the registers from the previous designs with memory blocks and more importantly to concurrently estimate multiple channels from different transmit to a single receive antenna with only very little additional hardware cost.

To achieve this we make the additional assumption that the length of the training sequence K is equal to $L \cdot R$, with R being an integer. The sum of Equation 1 can now be split into two partial sums as follows:

$$h_{ji}^{(r)}(n) = \sum_{l=0}^{L-1} x_j(n + l + r \cdot L) \cdot S^*(l + r \cdot L) \qquad (2)$$

$$h_{ji}(n) = \sum_{r=0}^{R-1} h_{ji}^{(r)}(n) \qquad (3)$$

The first part of the sum (Equation 2) can be interpreted as an FIR filter with the length of the channel's impulse response L. The intermediate results for each tap in the channel at its output are interleaved with a period of L cycles. The second part of the sum is built as a single correlator with a FIFO memory instead of a single register that accumulates the interleaved partial sums from the FIR to obtain the final result after $K = L \cdot R$ cycles. The block diagram of this structure is shown in Figure 6.

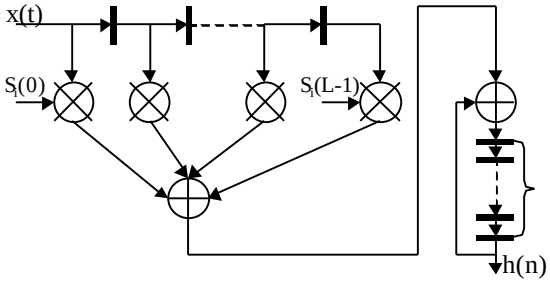

Figure 6 Hybrid Implementation

B. Channel estimation for multiple chanels

In this subsection the efficient implementation for the estimation of multiple channels at a single receive antenna in a UMTS-based MIMO system is described. The approach is based on the hybrid FIR/correlator structure that was outlined in the previous subsections of the paper and on the similarity of the UMTS OVSF channelization codes $C_i(n)$ that are used for the pilot channels. These codes have a period of 2^B and are described by:

$$C_i(n) = \sum_{b=0}^{B-1} n^{(b)} \cdot Code_i^{(b)} \qquad (4)$$

$Code_i^{(b)}$ is the number of the OVSF code used for the pilot channel of transmit antenna i and $xyz^{(b)}$ means the bth bit of xyz in a radix-2 representation. The number of bits needed to represent B is $\#B$. All operations performed on bits are modulo-2 arithmetic. When inserted into arithmetic

expressions a *–1* represents a binary *1*, while a *+1* represents a binary *0*.

After substituting Equation 4 together with $S_i(n) = S_c(n) \cdot C_i(n)$ from our first assumption into Equation 2 one obtains:

$$h_{ji}^{(r)}(n) = \sum_{l=0}^{L-1} x_j(n+l+r\cdot L)\cdot S_c^*(l+r\cdot L)\cdot\left(\sum_{b=0}^{B-1}(l+r\cdot L)^{(b)}\cdot Code_i^{(b)}\right) \quad (5)$$

The length of the channels impulse response is chosen to be a power of 2: $L = 2^{\#L}$. With this choice all bits $b \le \#L$ in $(l+r\cdot L)^{(b)}$ are only determined by l, while the bits $b > \#L$ depend solely on r. Keeping this in mind the channelization code $C_i(n)$ is split into two parts as follows:

$$C_{i,l} = \sum_{b=0}^{\#L-1} l^{(b)}\cdot Code_i^{(b)}, C_{i,r} = \sum_{b=\#L}^{\#B-1}(r\cdot L)^{(b)}\cdot Code_i^{(b)}$$
$$C_i(l+r\cdot L) = C_{i,l} + C_{i,r} \quad (6)$$

substituting into Equation 5 and 3 yields:

$$h_{ji}^{(r)}(n) = \sum_{l=0}^{L-1} x_j(n+l+r\cdot L)\cdot S_c^*(l+r\cdot L)\cdot C_{i,l} \quad (7)$$

$$h_{ji}(n) = \sum_{r=0}^{R-1} C_{i,r}\cdot h_{ji}^{(r)}(n) \quad (8)$$

The FIR is the first stage of the channel estimation. It only depends on the lower $\#L$ bits of the OVSF code that is used for the respective pilot channel. If the codes of the pilot channels in the system only differ in the bits $\#L$ to B the FIR part of the channel estimator needs to be implemented only once, as its coefficients will be identical. Only the correlator needs to be replicated for each transmit antenna. The diagram for the implementation of a 2x1 channel estimation is depicted in Figure 7.

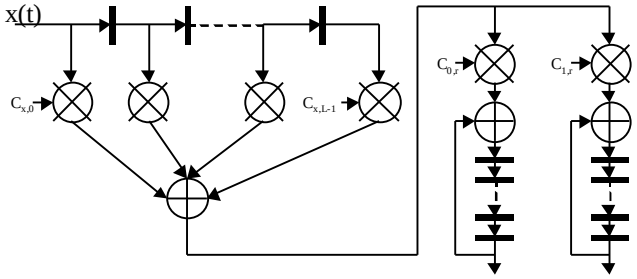

Figure 7 Multiple Channel Estimation

IV. RAKE RECEIVER AND FINGER ASSIGNMENT

The system uses a four finger RAKE receiver at each receive antenna to extract soft symbols from the incoming stream. This provides us with 16 complex values per symbol that need to be delivered to the MIMO decoder. An integral part of the RAKE receiver is the finger assignment unit, which searches the channel profile for the strongest available peaks and adjusts the finger positions of the receiver accordingly to track the channel profile.

A. RAKE receiver

For the implementation of the RAKE receiver two general structures are available, each with its own advantages and disadvantages. Both use correlators to realize the integrate and dump circuit for each finger. However they differ in the way the fingers are synchronized and how the delay of the fingers is adjusted.

Optimizing the RAKE by itself the more efficient implementation in terms of complexity is to use independent correlators for each finger. They start the integration process at their respective assigned delays. This structure only delays the reference signal of the internal time reference that is used for the despreading of the incoming signal by the delay of each finger. This requires only a negligible amount of registers in the delay line for a complex binary spreading sequence. However the disadvantage of this becomes obvious when the finger positions start moving around. Whenever a finger position is being moved to an earlier starting point, part of a symbol or for low spreading factors even entire symbols might be lost. This structure therefore requires a much more advanced finger management to minimize the movement of the fingers and to ensure symbols are not being counted twice or skipped.

The second approach uses a tapped delay line on the incoming signal to adjust the finger positions. All correlators start and finish at the same time instant. Even though this adds a significant amount of registers to the design to build the delay line and the multiplexer to tap it, it greatly simplifies the finger allocation scheme. Virtually no attention needs to be paid to avoid unnecessary movement of the fingers allowing them to be placed at their optimum position at all times without taking a performance penalty through lost symbols or partially decoded soft symbols. Furthermore, the delay line can be implemented as a FIFO memory allowing to save registers and to reduce power consumption, as the high toggling activity that is inherent to register delay lines is avoided. This second approach has been chosen for our implementation due to its higher robustness and due to the fact that it allows to efficiently use the FPGAs built-in memories.

Before the soft symbols are delivered to the MIMO decoder through the FIFO a 2-bit identifier is attached to them that identifies which set of finger positions they belong to. This is important to assure that the MIMO decoder can associate them with the correct set of channel coefficients.

B. Finger assignment

The finger assignment procedure assumes that the peaks of the channel profile appear at the same delay across all the receive antennas. This assumption is based on the fact that the time resolution of the channel estimation at four times oversampling of the chip frequency $F_c = 4MHz$ is only about 60ns which is fairly long compared to the delay variation that might be caused by the antenna spacing. The assumption has also been confirmed through a series of channel measurements.

To decide on the optimum positioning of the RAKE fingers the amplitudes of all 16 channel profiles are approximated by $|h_{ji}(n)| \approx |\Re(h_{ji}(n))| + |\Im(h_{ji}(n))|$ and are added up to a common channel power profile. As this profile is generated sequentially while the channel estimation is performed the finger searcher finds the four strongest peaks for the finger positions. At the same time the channel coefficients that belong to the previous round of finger position assignments are extracted from the channel. This process is also done on the fly as the channel estimation sequentially delivers the channel coefficients and therefore there is no need to store the 16 channel estimates, which would involve a significant amount of additional memory. Just as the soft symbols the coefficients are labeled with a 2-bit identifier to realign them with the soft symbols that were decoded from the same set of delays at the MIMO decoder.

V. FREQUENCY OFFSET ESTIMATION AND COMPENSATION

This part of the design is necessary to remove the frequency offset that is caused by a mismatch between the local oscillators at the transmitter and the receiver. Typical oscillator offset for a relatively high performance oscillator can be expected to be around 1ppm which for a carrier frequency of about 2 GHz results in a residual frequency offset ΔF around 2 KHz. This is well beyond what the channel estimation can compensate for. However as this is a constant offset it can be estimated and can be compensated. It is assumed that at both the transmitter as well as at the receiver one local oscillator is used for all antennas resulting in identical frequency offsets. Based on this assumption the strongest signal from the receive antennas is used together with one of the four available pilot signals to perform the estimation [5]. If it fails it can be repeated using one of the other pilot channels originating from another transmitting antenna. Looking exclusively at the pilot channel the incoming signal with a frequency offset $\Delta w = 2 \cdot p \cdot \Delta F$ can be described as:

$$x_j(n) = S_c(n) \cdot C_j(n) \cdot e^{j \cdot 2 \cdot p \cdot n \cdot \Delta F / F_C} \qquad (7)$$

It is despread and integrated with a correlator that is placed on the channels strongest peak for the duration of a symbol on the pilot channel ($P = 256$ chips). This yields:

$$X_j(l) = (1 + j) \cdot a \cdot e^{j \cdot f} \qquad (8)$$

$$f = 2 \cdot p \cdot l \cdot 256 \cdot \Delta F / F_C \;,\; a = \frac{\cos(2 \cdot p \cdot \Delta F / F_C \cdot P) - 1}{\cos(2 \cdot p \cdot \Delta F / F_C) - 1}$$

The amplitude a as well as the phase f of the symbols depend on the frequency offset and can therefore be used for the estimation. It becomes also apparent that both variables have an ambiguity when $\Delta F \geq 1/2 \cdot F_c / P$, which in the case of our system leads to a maximum tolerated frequency offset of $\Delta F_{\max} = 8KHz$.

A. Structure of the frequency offset compensation

The general structure can be categorized into *feed-forward (a)* and *feedback (b)* schemes, as shown in Figure 8. Algorithms of the first type are easier to analyze and to control, as they can not cause instabilities. The latter might lead to oscillations that need to be analyzed carefully. However our analysis shows that they usually have a better performance especially for high frequency offsets (close to $\Delta F_{\max}$) and in low SNR environments as they reduce nonlinear effects in the estimation caused by noise and by the phase ambiguity.

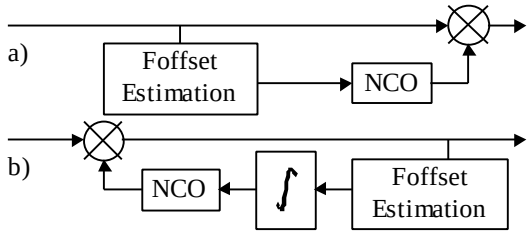

Figure 8 Frequency offset estimation/compensation

B. Estimation methods

1) Power based estimation

Figure 9 depicts the general scheme for a power based frequency-offset estimation. It is based on an early/late type detector scheme. The signal is split up and multiplied with a signal provided by two numerically controlled oscillators (NCOs). One of them operates at $\Delta F + F_0$ while the other operates at $\Delta F - F_0$. The resulting signals are despread independently and the difference of the resulting symbol amplitude is used to update the offset estimate.

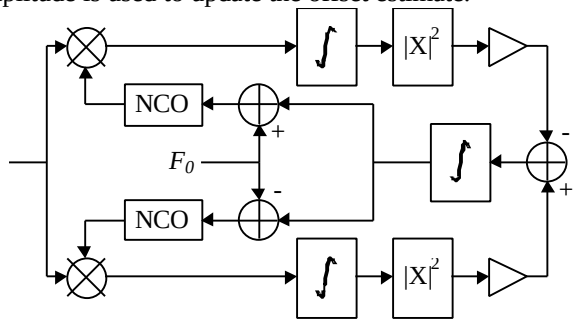

Figure 9 Power based frequency-offset estimation

This circuit exhibits a very good performance over a wide range of frequency offsets and at a low SNR and allows to efficiently trade convergence speed, stability and accuracy. However it has a fairly high complexity mainly due to the two NCOs and the multipliers needed in the signal path and in the feedback path.

2) Phase based estimation

Two variations to estimate the frequency offset through the phase of the despread pilot signals are shown in Figure 10. The first approach computes the phase difference between subsequent symbols in the complex domain, followed by the computation of the phase. The second structure essentially

performs the same operation. However it reduces the complexity of the implementation by avoiding the complex multiplication with the previous symbol. The phase of the despread symbols is instead computed first. Then the phase difference is derived by a simple subtraction of the previous symbols phase in the real domain.

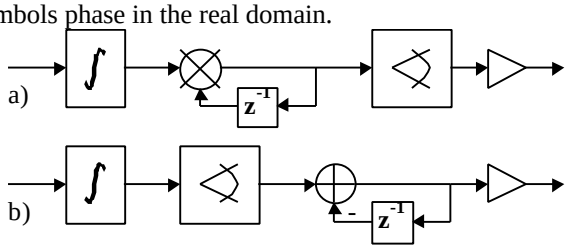

Figure 10 Phase based frequency-offset estimation

The most complexity in the latter circuit is in the phase estimation, which is basically the computation of

$$\Delta f = \arctan\left(\frac{imag(x)}{real(x)}\right) \qquad (9)$$

involving a division and the computation of the *arctan*. The division can be performed sequentially using only an adder and a comparator. The *arctan* function needs to be implemented only in the first quadrant. However the fact that it is defined over a range from 0 to infinity ($[0\cdots\infty]$) and the high nonlinearity make a direct approximation very difficult. To limit the range of the argument a well-known trigonometric relation is used:

$$\arctan(x) = \begin{cases} \arctan(x), 0 \le x \le 1 \\ \frac{p}{2} - \arctan\left(\frac{1}{x}\right), x > 1 \end{cases} \qquad (10)$$

With this expression the *arctan* only needs to be defined in $[0\cdots1]$. Within this range it can be approximated as a linear function of the argument through:

$$\arctan(x) = 0.7918 \cdot x + 0.0493, x \in [0\ldots1] \qquad (11)$$

However combined with either a feedback structure or a complex phase difference computation as shown in Figures 8b and 10a averaging would fail to remove the approximation error, as the argument of the *arctan* function could remain relatively constant over time. For a feedback structure an approximation with the highest accuracy around $x = 0$ using the complex phase difference would lead to better results. In our implementation we use a feed forward structure as shown in Figure 8a with the low complexity phase based frequency offset estimation shown in Figure 10b. In this configuration the input argument of the *arctan* function constantly changes and therefore the error inferred by the linear approximation (11) can be averaged out.

VI. SYSTEM INTEGRATION AND COMPLEXITY

The receiver front-end has been synthesized for XILINX Virtex FPGAs. The basic clock frequency of the system at a 4x oversampling of the 4 MHz chip rate is relatively low. To use the capabilities of today's technologies more efficiently the four receive antennas are being multiplexed into a single stream and processed sequentially. This requires a clock rate of 16 times the chip rate, which is equal to 64 MHz.

Table 1 Implementation in a XILINX Virtex-1000 FPGA

	LUTs	Registers	Memory
AGC	15%	-	-
RRCosine	16%	7%	-
Foffset Compensation	5%	3%	-
4x4 Chennel Estimation	20%	5%	8/32
4x RAKE	3%	1%	2/32

A brief review of the overall complexity of the system is given in Table 1. It shows the percentage of available lookup tables (LUTs), registers and memory for the implementation of our design in a XILINX Virtex-1000 FPGA. The numbers that were derived from initial synthesis runs suggest that the entire system can be implemented in such a programmable device with the optimizations that were applied. The most significant improvement was achieved with the optimization of the channel estimation, which reduced the area from about 300% of the FPGA down to 50%. The further reduction down to the numbers given in the table was mainly achieved through multiplexing all four receive antennas. The system was implemented and simulated in C and then automatically translated to VHDL using Lucent's rapid prototyping methodology BLADE as described in [2] and [3].

VII. CONCLUSION

In this paper we presented our implementation of a UMTS-based 4x4 MIMO receiver front-end in XILINX Virtex FPGAs. We described briefly how such a system might be integrated within the UMTS standards. Its major building blocks are the channel estimation for the 4x4 channels, the 4xRAKE receiver and finger management as well as the frequency offset compensation. We presented an efficient architecture to greatly reduce the complexity of the channel estimation with only minor restrictions on the choice of channelization codes for the pilot channels. We reviewed our choice for the implementation of the RAKE receiver and the channel estimation and described different approaches for the implementation of frequency offset estimation with respect to their complexity and outlined the approach that was actually implemented in our design.

VIII. REFERENCES

[1] G.D. Golden, G.J. Foschini, R.A.Valenzuela, P.W. Wolnianski, "Detection Algorithm and Individual Laboratory Results Using V-BLAST Space-Time Communication Architecture", Electron Letters, Vol. 35, No. 1, pp.11-4, Jan. 1999

[2] A. Burg, B. Haller, M. Guillaud, M. Rupp, E. Beck, L. Mailaender: "A Rapid Prototyping Methodology for Algorithm Development in Wireless Communications", Proc. of Design Automation and Test in Eruope, Designer's Forum, DATE'01, Munich, 13-16 March 2001

[3] M. Guillaud, A. Burg, L. Mailaender, B. Haller, M. Rupp, E. Beck: "From Basic Concepts to Real-Time Implementation: Prototyping WCDMA Downlink Receiver Algorithms – A Case Study", Asilomar Conf., Monterey, California, Oct. 2000

[4] M. Rupp, S. Das, M. Guillaud: "on MIMO Decoding Algorithms for UMTS", submitted to Asilomar Conf. 2001

[5] O.C. Maus, F. Classen, H. Meyr: "Carrier Frequency Recovery for a Fully Digital Direct-Sequence Spread-Spectrum Receiver: A Comparison", Proc. 43rd IEEE VTC Conf. 1993, New Jersey, USA, pp. 392-395