

Reconfigurable Signal Processing in Wireless Terminals

Jürgen Helmschmidt¹, Eberhard Schüler¹, Prashant Rao¹, Sergio Rossi², Serge di Matteo², Rainer Bonitz³

¹ PACT XPP Technologies, ² Accent, ³ STMicroelectronics

Abstract

In this paper, we show the necessity of reconfigurable hardware for data and signal processing in wireless mobile terminals. We first identify the key processing power requirements for realizing a third generation wireless mobile terminal with multi-link and multi-standard capabilities. This is done on the basis of two real-world applications: a flexible mobile rake receiver for UMTS/W-CDMA and an OFDM decoder for high-speed wireless LAN protocols. We present a software-defined concept and a system implementation for the signal processing in these applications. The system is based on a DSP for control-flow oriented tasks, dedicated hardware for predefined data-flow oriented tasks and reconfigurable hardware for software-defined data-flow oriented tasks. A new coarse-grained approach is used to implement the reconfigurable hardware, which is in the form of an array of processing elements and also contains resource management mechanisms. The features and programming concepts of the reconfigurable hardware are emphasized further in the text.

1 Motivation

Next-generation mobile terminals must provide consumers with a wide host of application possibilities to meet the demands of an ever-advancing concept of mobile connectivity.

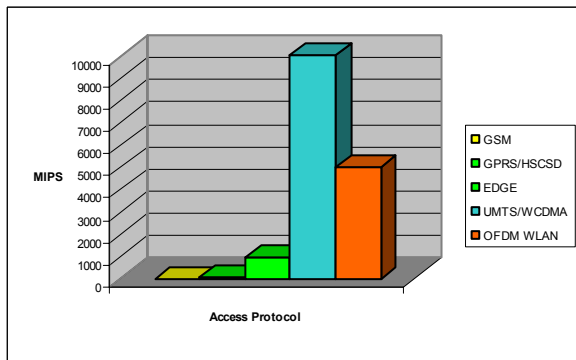


Figure 1: Processing power requirements of wireless access protocols

Technology differentiators such as interactive multimedia applications, video and audio broadcasting, and additional hardware in the form of MP3 players, digital cameras and color displays pose high processing power demands that must be met by the baseband and application processors in mobile terminals. These processing power demands are in addition to the already high demands of next-generation wireless protocols.

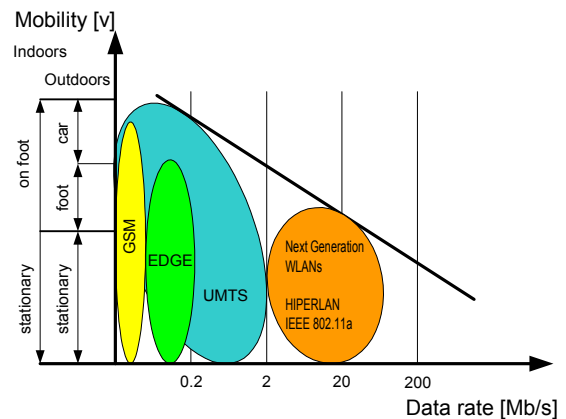


Figure 2: Data rate vs. mobility for wireless access

Through industry consensus, processing power is measured in terms of Millions of Instructions per Second or MIPS. Current GSM phones require approximately 10 MIPS. GPRS/HSCSD functionality requires approximately 100 MIPS. For the implementation of EDGE around 1000 MIPS are required. Potentially up to 10,000 MIPS are required for the implementation of 3rd generation UMTS/W-CDMA. Wireless LAN protocols implementing Orthogonal Frequency Division Multiplexing (OFDM) require around 5000 MIPS. These figures are visualized in the chart in Figure 1.

The infrastructure in the landscape of next-generation wireless telecommunication will also consist of multiple access protocols. These differentiate themselves in the data rate available to the mobile terminal depending on mobility demands, as shown in Figure 2. Third generation cellular UMTS/W-CDMA solutions provide a maximum data rate from a few hundred kbit/s at high mobility up to 2 Mbit/s in

stationary environments. Modern wireless LAN protocols such as IEEE 802.11a and HIPERLAN/2 can provide data rates of up to 54 Mbit/s in stationary and low mobility environments.

To provide consumers with continuously optimal wireless service regardless of the underlying access protocol, next-generation mobile terminals will also have to provide high a degree of flexibility within the baseband processing. This demand for flexibility can be met by Software Defined Radio (SDR) concepts, whereby the radio baseband digital signal processing – which differs according to the access protocol – is implemented using programmable solutions such as DSPs.

Modern high-performance DSPs can provide around 1600 MIPS at clock speeds of 200 MHz. Stringent power consumption demands – as consumers have become accustomed to a battery life of several hours of active usage and several days of standby time – constrain the maximum clock speeds of devices within a mobile terminal. Thus, solutions providing the high processing power demands within mobile terminals involve implementing multiple hardware accelerators for the different high-demand processing tasks, with associated losses in total silicon efficiency, high development costs and limited flexibility, or the use of multiple DSPs, with high demands on silicon area and power consumption.

2 Reconfigurable Hardware

Reconfigurable hardware, in the form of an array with multiple processing elements, presents a solution with the processing power comparable to that of hardware accelerators, with the required flexibility of a DSP. The array-based reconfigurable hardware approaches differentiate themselves in the granularities of the processing elements. In this paper we present an approach based on coarse-grained processing elements such as ALUs, multipliers and RAMs. Especially for the signal processing requirements in SDR, which require a high number of multiply and accumulate operations, such an approach provides a high amount of processing power in a cost-efficient implementation. The approach involves the usage of the reconfigurable hardware as a coprocessor to a standard microcontroller or DSP.

The functionality of the reconfigurable array is defined by software-based configurations, which describe the behavior of the processing elements and the routing between them. Special hardware protocols implemented in the communication and control structures of the array ensure that configurations cannot be overwritten illegally.

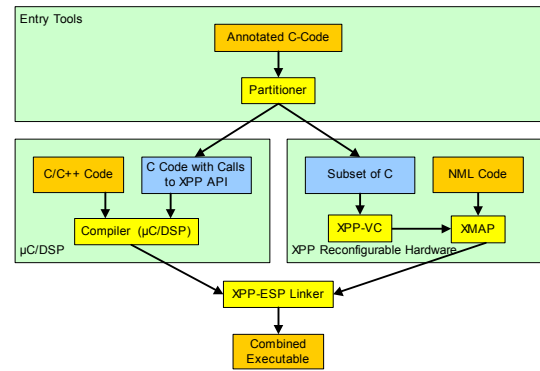


Figure 3: Integrated design flow for reconfigurable hardware and microcontroller

Furthermore, these protocols implement a unique token-based data communication between the processing elements that enable an efficient pipeline-based operation. The combination of these features enables high-level programming of the array using C and a software-based simulation environment. A programming and simulation environment integrating the tool and design flows of the microcontroller/DSP and the reconfigurable hardware can thus be implemented, as shown in Figure 3.

3 Operational Scenarios

The following section describes the operation of a Software Defined Radio (SDR) solution for UMTS/W-CDMA and modern wireless LAN protocols involving a reconfigurable processing array and a DSP/microcontroller. A multi-standard, multi-link wireless terminal must provide the capability of handling at least these protocols simultaneously. By time-slicing the processing of both protocols over the same hardware, a large savings in the resources required can be achieved.

The analysis of the operational scenarios initially involves the partitioning of the processing onto the different hardware resources. Critical computational parts with high data streaming demands are mapped onto the reconfigurable processing array. Algorithmic parts with low criticality, mostly implementing control code, are mapped onto the DSP/microcontroller.

In the following, the first part describes the implementation of a rake receiver used for the detection, tracking, descrambling, despreading, channel correction and combination of CDMA signals. The second part describes the implementation of an OFDM receiver used for the demodulation of the Wireless LAN protocols.

3.1 Rake Receiver

In contrast to the current second generation mobile telecommunications standards, the UMTS/W-CDMA infrastructure is capable of handling a “soft” handover, whereby a mobile terminal is in contact with multiple basestations at the same time. This can occur, for example, when a mobile terminal is at a cell border and the signals from all surrounding basestations have near equal strength. The signals from multiple basestations differentiate themselves in their scrambling code. Thus, to handle the “soft” handover scenario, the rake receiver in the mobile terminal must be capable of correlating the received signal with the individual scrambling codes of all the basestations involved. Furthermore, the rake receiver is capable of collecting and using the energy from multipath components of a signal. Hereby, the rake receiver correlates the received signal with different time-offsets of the spreading and scrambling codes and performs a channel correction to compensate for the differing channel characteristics of the individual multipaths. These steps are performed for all multipaths in the environment. The results of these operations are combined to drive the decision-making process for the value of the received symbol.

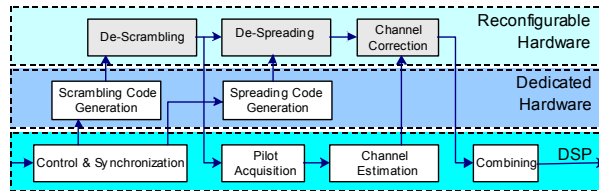


Figure 4: Partitioning of the rake receiver

In addition to the actual signal reception tasks, the rake receiver must also perform a set of timing and synchronization tasks. A path searcher performs a correlation of a fixed set of pilot signals over a sliding window to detect the paths with the strongest signal values in a multipath environment. The offsets of these paths are stored within a control context and are used to generate the required offsets for the individual rake fingers that descramble and despread the chip rate signals. The path searcher divides itself into a coarse and a fine searcher, with differing repetition intervals and accuracies. A path tracker is responsible for the tracking and the resynchronization of the paths that are currently being received. The channel estimator calculates the channel coefficients that are used for the channel correction. The channel coefficients are calculated on the basis of a specific sequence of pilot signals.

The operational scenario for the mobile terminal rake receiver implementation shown in the following

involves a “soft” handover scenario with up to six basestations, with the reception of three multipaths per basestation. The basic partitioning of tasks between a DSP, dedicated hardware and reconfigurable hardware is shown in Figure 4. Dataflow oriented tasks that operate on a word-level granular data stream are executed using the reconfigurable hardware. A DSP is used to execute the control-flow and synchronization tasks. Bit-level data processing tasks that execute continuously are mapped onto dedicated hardware resources.

Basestations	Channels					M u l t i p a t h s
	1	2	3	4	5	
1	18	9	6	4	3	M u l t i p a t h s
2	9	4	3	2	1	
3	6	3	2	1	1	
4	4	2	1	1	-	
5	3	1	1	-	-	
6	3	1	1	-	-	

Table 1: Rake receiver finger scenarios

For this operational implementation, 18 (6x3) rake fingers for the descrambling and despread operations must be realized. As the UMTS/W-CDMA chip rate is 3.84 MHz, a single physical finger is actually implemented. By repeating the descrambling and despread operation on a single chip over multiple scrambling and spreading codes and time multiplexing the resulting data stream, the single physical finger thus corresponds to an implementation of 18 rake fingers. The minimum operational frequency of the single finger to accommodate this maximum scenario is thus $18 \times 3.84 \text{ MHz} = 69.12 \text{ MHz}$. Further possible channel, basestation and multipath scenarios are shown in Table 1. The scenarios that require the full frequency of 69.12 MHz are shaded. The remaining scenarios that are shaded light need not run at the full frequency.

The physical finger is implemented in the form of a pipeline on the reconfigurable hardware. The following assumptions are made in the design:

- Sampling Rate : 3.84 MHz
- Data Representation: 12-bits for I and Q each
- Spreading Factors: 4 to 512
- Symbol Encoding: Space Time Transmit Diversity (STTD)

The individual components of the rake receiver finger are described in the following.

Descrambler. The descrambling operation involves the complex multiplication of the aligned incoming

data with the scrambling codes. The implementation of the descrambler on the reconfigurable hardware is shown in Figure 5. The scrambling code generator, which is realized as dedicated hardware as shown in Figure 4, provides the scrambling code in the form of a two-bit representation. The reconfigurable hardware translates the two-bit representation into the form of $\pm 1 \pm j$ by the use of multiplexers.

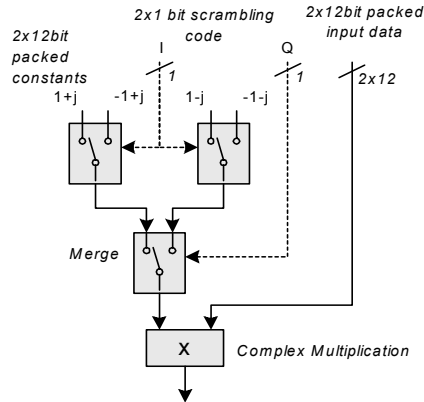


Figure 5: Rake descrambler with scrambling code multiplexer

Despreader. The despreading operation is the multiplication of the corresponding spreading code (OVSF) code with the real and imaginary part of the descrambled data sequence followed by a summation over a length equal to the spreading factor. The spreading factor in the downlink can range from 4 to 512 chips. Figure 6 shows the block diagram implementation of the despreader on the reconfigurable hardware using the word granular processing elements.

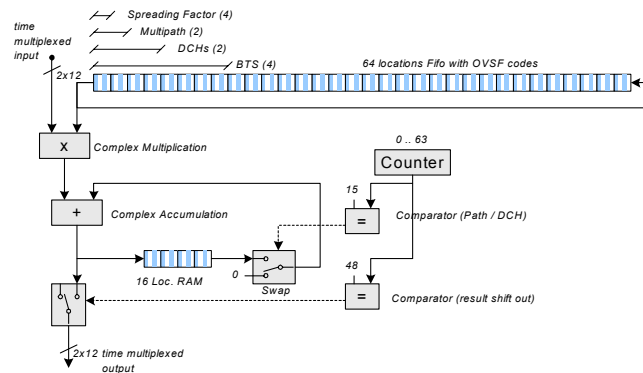


Figure 6: Implementation of the rake despreader on the reconfigurable hardware

Channel Correction Unit. Figure 7 shows the implementation of the channel correction unit. In addition to the actual channel correction, the unit also performs the STTD decoding of the symbols. In STTD encoding, the symbol stream is divided into two

streams each with half the transmit frequency. Each stream is transmitted over a locally separate antenna. The first symbol stream remains unchanged. The second symbol stream is reordered and the conjugate complex of the symbol is transmitted. The antennas are located far away from each other that each stream has its own channel coefficient, but close enough so that both symbols arrive at the receiver at the same time.

Using the pilot sequences, the DSP calculates the channel coefficients, which are then transferred to the reconfigurable hardware. The channel corrector takes two symbols from the despreader at half the symbol rate to then execute the STTD decoding and the channel correction. The upper FIFOs shown in Figure 7 store the channel coefficients for finger. The symbols arrive from the despreader in a time-multiplexed manner. The channel weighting consists of a complex multiplication and the addition of two subsequent symbols for each finger.

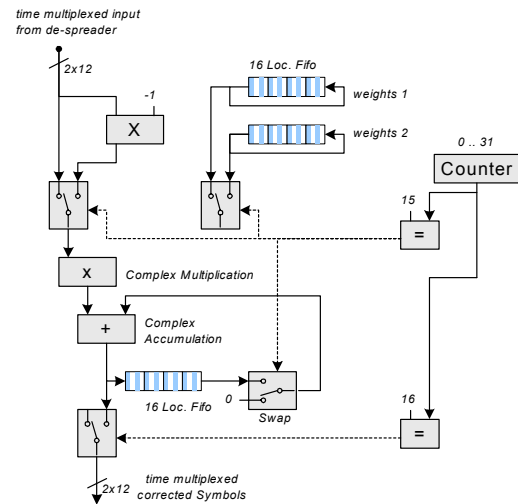


Figure 7: Block diagram of the channel correction unit

3.2 OFDM Decoder

The high-bandwidth wireless LAN standards IEEE 802.11a and Hiperlan/2 are Orthogonal Frequency Division Multiplexing (OFDM) systems. Hereby, symbols are modulated and spread over 48 low-bandwidth carriers, with an additional 4 carriers containing pilot signals. The standards define various modulation schemes and code rates, which specify data rates from 6 up to 54 Mbit/sec. Figure 8 shows the required modules the mapping of the functionalities onto the DSP, the dedicated hardware and the reconfigurable hardware.

The modules that are mapped onto the reconfigurable processor are sketched in Figure 7. The complex input samples are down-sampled, interpolated and then propagated to the preamble detection for framing and synchronization.

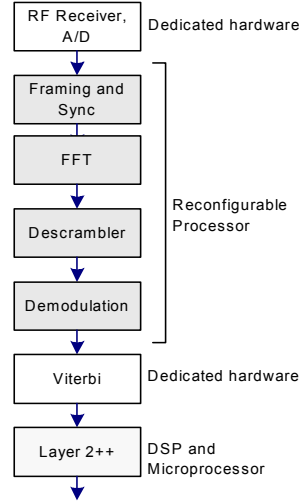


Figure 8: Partitioning of the OFDM decoder tasks

The FFT64 is used to transform the resulting OFDM symbol from the time into the frequency domain. Individual symbols are acquired by demodulating the resulting carrier signals. A Viterbi decoder is used for the forward error correction before propagating the data points for higher layer protocol processing.

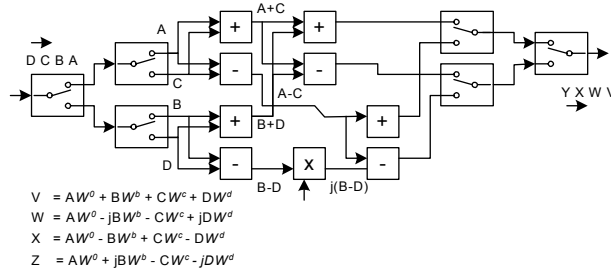


Figure 9: The FFT radix-4 kernel mapped onto complex-arithmetic ALUs

The FFT64 uses the radix-4 approach. The radix-4 computation is performed in a pipeline delivering a result value with every clock cycle. The block diagram in Figure 9 shows the implementation of the FFT64 with the radix-4 kernel. Read and write addresses are stored in circular lookup tables, which are implemented as preloaded FIFOs. Twiddle factors for all 3 stages of the FFT64 are also stored in a lookup table. Initially 64 samples stream into the data RAM. The output of the RAM is multiplied with the twiddles and then streams to the radix-4 module. The output is read back to the dual-ported data RAM for the next iteration. After the third iteration, the transformed data

is available at the output multiplexer. A simple counter and comparator control the multiplexer stages.

The accuracy of the complex input signal is 10 bit. With every stage a scaling (2-bit right shift) is required to prevent overflow. For three stages of the FFT64 we finally get a 4-bit precision in the result.

Figure 10 shows a module-level mapping of resources and configurations on the reconfigurable hardware. Modules contained in Configuration 1 are required to run continuously and thus remain in the hardware. The resources of the preamble detection (Configuration 2a) can be removed after execution. The freed resources are then available for the demodulation tasks contained in Configuration 2b.

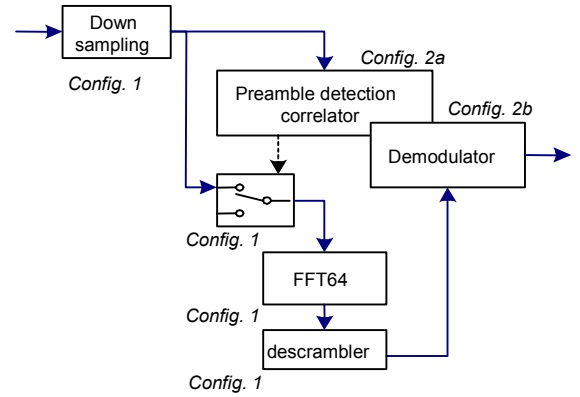


Figure 10: Configurations mapping on the reconfigurable hardware for the OFDM decoder

4 System Implementation

Figure 11 shows a functional block diagram of an SDR evaluation board containing a microcontroller, DSP, FPGA and a reconfigurable array processor. The microcontroller, for housekeeping tasks, is realized with a MIPS 4Kc core contained in the QuickMIPS device. A DSP slot allows the integration of a selection of different DSPs. A Programmable Logic Device, in the form of the streaming FPGA, provides a variety of data routing configurations to allow the investigation of different hardware and software processing trade-offs. Furthermore, dedicated hardware can be implemented in the FPGA. The XPP-64A device provides the reconfigurable computing platform.

The XPP-64A is a unique array-based reconfigurable parallel processing platform. It provides an 8x8 array of computing elements called ALU Processing Array Elements (ALU-PAEs) with a row of 8 storage elements called RAM-PAEs on either side.

Each ALU-PAE processes 24 bit words using a DSP-based instruction set. RAM-PAEs contain 512x24 bits

of dual-ported SRAM and can be configured as standard RAM and FIFO modes. Each PAE also includes individually configurable vertical and horizontal routing resources. Four dual-channel Input/Output ports, capable of functioning in streaming and RAM-addressing modes, handle external communication.

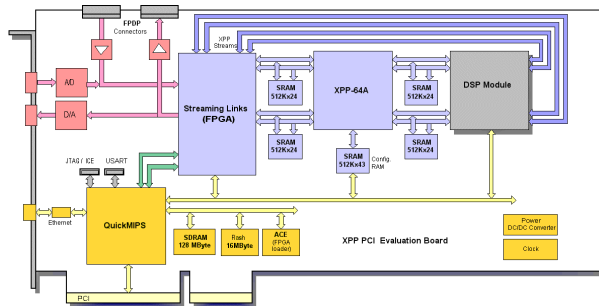


Figure 11: Software Defined Radio evaluation board for mobile terminals

All resources in the XPP-64A execute completely synchronously. A single clock domain is used for the entire device. Handshake protocols implemented in the communication resources maintain a token-oriented data flow. Algorithms programmed on the array are executed by a number of processing elements simultaneously in the form of a pipeline. A configuration manager is responsible for the resource handling on the array. The array is capable of being reconfigured with different tasks during run-time. Individual resources on the array can hereby be independently reconfigured and allotted to the different tasks. This holds for the functional, as well as the communication objects.

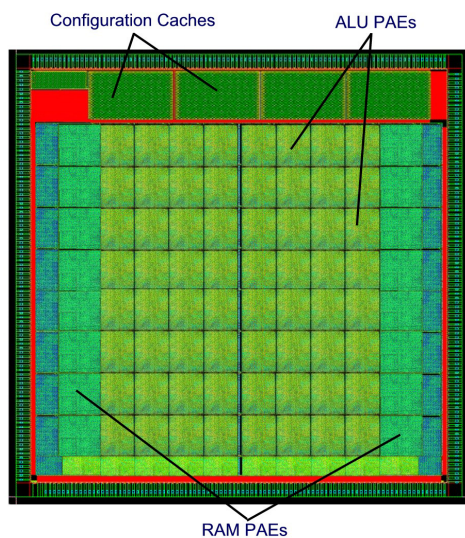


Figure 12: Layout of the XPP64A reconfigurable processing array on 0.13µm CMOS

Figure 12 shows the layout of the XPP64A-1 realized on 0.13 µm CMOS, using the STMicroelectronics HCMOS9 [6] process. Fabricated in 0.13 µm CMOS, the XPP has a drawn feature size of 130nm resulting in a physical gate length of 110nm. A high resistivity STI substrate with retrograde, triple wells is used. Dual Vt technology helps reduce power consumption. Furthermore nitrided dual gate oxides of equivalent thickness are used at sizes of 2, 5.0 and 6.5nm. Contacts to the copper metal layers are realized by CoSi₂ advanced salicide. The metal layers themselves are 6-8 dual Damascene copper metal layers with low -k dielectric plus additional local interconnect layers.

5 Conclusion

The presented combination of reconfigurable hardware, dedicated hardware and a DSP shows a very good fit to handle SDR wireless applications. Especially in the case of enabling mobile terminals with multi-link and multi-standard capabilities that require a high amount of processing power combined with high demands in flexibility, we identified a mapping of two popular 3G wireless receiver standards onto the presented hardware. Special emphasis was put on the new coarse-grained reconfigurable hardware, which is realized in the form of an array of processing elements with resource management. The reconfigurable hardware enables the high throughput required for the signal processing by means of a pipeline-based parallelization, which also results in low overall power consumption. The implicit software-based approach allows also for a high degree of flexibility and shortened design and testing times.

References

- [1] H. Lange, O. Franzen, H. Schröder, M. Bucker, B. Oelkrug, *Reconfigurable Multi-Accumulate-base Processing Element*, Nokia Research Center, Bochum, Germany, IEEE Workshop on Heterogeneous Reconfigurable Systems on Chip, Hamburg, April 2002
- [2] Roke Manor Research, *Roke Manor Research Business & Technology Review 2001*
- [3] IMEC Technology, J-Y. Mignolet, S. Vernalde, D. Verkest, R. Lauwereins *Enabling Hardware-Software Multitasking on a Reconfigurable Computing Platform for Networked Portable Multimedia Appliances*
http://www.imec.be/design/pdf/reconfig/ERSA_2002_enabling.pdf
- [4] Guido H. Bruck, Peter Jung, *Software Defined Radio in drahtlosen Endgeräten*, Univesität Duisburg, 2001
- [5] PACT XPP Technologies AG. www.pactxpp.com
- [6] STMicroelectronics HCMOS9book Rev.1.1