

EXPLORING THE CAPABILITIES OF RECONFIGURABLE HARDWARE FOR OFDM-BASED WLANS

Thilo Pionteck, Lukusa D. Kabulepa, and Manfred Glesner

Darmstadt University of Technology, Institute of Microelectronic Systems, Karlstr. 15, 64285 Darmstadt, Germany

pionteck@mes.tu-darmstadt.de

Abstract In this paper, the potential of reconfigurable hardware in wireless communication systems is evaluated. As most of the published works aim at the usage of reconfigurable architectures as a universal platform for different standards, the optimization capabilities of reconfiguration techniques for hardware modules within one standard are often not considered. This work focuses on the hardware optimization within one transmission scheme, namely OFDM (Orthogonal Frequency Division Multiplexing). By making use of special characteristics of packet-based WLANs and standard specifications, functional blocks of an OFDM receiver can be mapped on the same hardware. Even performance enhancements can be achieved without any additional hardware.

Keywords: Reconfigurable hardware, OFDM, digital baseband

1. Introduction

The usage of reconfigurable architectures as a hardware platform for wireless communication systems is often considered to be very suitable for solving the increasing requirements of modern communication systems. Critical design issues like performance, low-power consumption and flexibility seem to be facilitated by reconfigurable hardware. Especially the flexibility of reconfigurable architectures raises the idea of a universal architecture supporting different communication standards. Some works even expand the idea to a support of standards based on different transmission schemes [Helmschmidt et al., 2003]. Thereby designers inevitably encounter the fundamental trade-off between flexibility and efficiency. Flexibility can only be achieved at the cost of performance, chip area and power consumption. As different transmission schemes significantly differ in their transceiver structure, a universal hardware platform results in a significant hardware overhead and performance degrada-

Please use the following format when citing this chapter:

Pionteck, Thilo, Kabulepa, Lukusa, D., Glesner, Manfred, 2006, in IFIP International Federation for Information Processing, Volume 200, VLSI-SOC: From Systems to Chips, eds. Glesner, M., Reis, R., Indrusiak, L., Mooney, V., Eveking, H., (Boston: Springer), pp. 149-164.

tion. Hence, many research efforts are devoted to reconfigurable architectures supporting different standards based on the same transmission scheme. Such architectures require less flexibility as the basic operations of a transceiver for one specific transmission scheme are mostly the same. For example, OFDM-based (Orthogonol Frequency Division Multiplexing) WLAN standards like IEEE 802.11a and HiperLAN/2 show a huge similarity in the physical layer. The usage of reconfigurable architectures for supporting both standards seems to be a promising application field.

The flexibility required for supporting different standards of the same transmission scheme can also be used to improve the performance within one standard. WLAN standards show certain characteristics which can be used to reduce the hardware effort of receivers. Dynamic reconfiguration can be applied as certain functions within a receiver are not required permanently. By mapping these functions onto the same hardware components, the released resources can be used for the sake of optimizing the receiver performance.

The rest of the paper is organized as follows: in section 2, the basic principles of OFDM-based WLANs are presented. Based on these analyses, reconfiguration capabilities of OFDM receivers are shown in section 3. Section 4 gives a design example for a function-specific dynamically reconfigurable hardware design for the blocks depicted in the last section. Finally, a conclusion is given in section 5.

2. OFDM-based WLANs

The usage of OFDM as a transmission scheme for emerging WLAN standards like HiperLAN/2[*Hip*, 2001] and IEEE 802.11a[IEEE802.11a, 1999] has proved its suitability for high data-rate wireless communication systems. The basic principle behind OFDM consists in splitting the available bandwidth into many narrowband subchannels. A high transmission rate can be achieved by sending a large number of subcarriers in parallel as lower rate data streams. The discrete-time equivalent of a complex OFDM signal can be obtained by means of the IDFT (Inverse Discrete Fourier Transform) [Nee and Prasad, 2000]. Therefore the receiver can demodulate the transmitted signal by applying the DFT (Discrete Fourier Transform). Thus the FFT is essential for an OFDM receiver and is also one of the most computation-intensive blocks besides the Viterbi decoder. The complete structure of an OFDM receiver is shown in figure 1.

The first functional block in the baseband processing for an OFDM receiver is the synchronization unit. Its aim is to detect an incoming data packet and to compensate for a potential frequency offset. Therefore special preambles at the beginning of a transmission are used. After removing a cyclic prefix which was introduced at the transmitter side in order to avoid intersymbol interference,

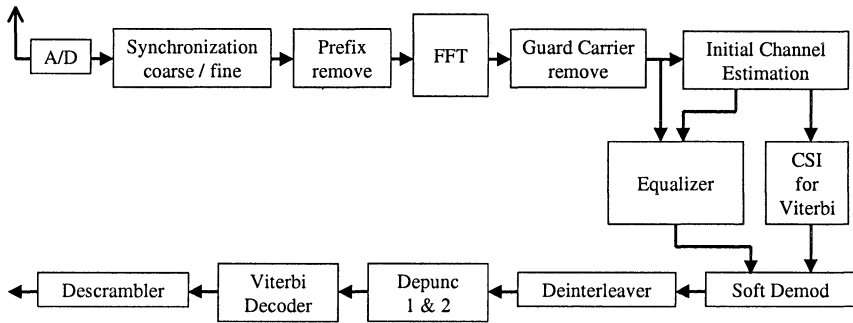


Figure 1. Block Diagram of an OFDM Receiver

the signal is transformed from the time-domain back to the frequency-domain by applying a FFT. For the compensation of the channel distortion, a channel estimation is done at the beginning of each transmission burst. The estimates are used by the equalizer for the compensation of the channel distortion by means of a zero-forcing equalization. In addition, channel state information are generated by using the channel estimates. The channel state information are used to weight the equalized soft values from the equalizer. Prior to the Viterbi processing of the soft values, the interleaving and puncturing have to be reversed. Finally the data bits are descrambled.

Transmission Format

The HiperLAN/2 standard defines a basic frame with a length of 2 ms, which comprises five different burst types. For a downlink connection, two types are defined, a BCB (BroadCast Burst) and a DLB (DownLink Burst) [Hip, 2001]. Figure 2 shows the structures of these two burst types. The preamble of the BCB is divided into three symbol groups. The A and B symbols enable frame synchronization, automatic gain control and frequency synchronization. The C symbols are used for channel estimation as well as for fine frequency synchronization. While the C symbols are included in both burst types, the A and B symbols are only used for the BCB. As a DLB is preceded by a BCB, a new frame and frequency synchronization is not required for the DLB.

For the WLAN standards IEEE 802.11a and HiperLAN/2, a channel raster of 20 MHz is used. With a subcarrier spacing of 312.5 kHz, this results in 64 subchannels. For one OFDM symbol channel, 11 guard-carriers are defined in order to achieve a sufficient adjacent channel suppression. Moreover, the DC-carrier is not used either. The remaining 52 subchannels are used for

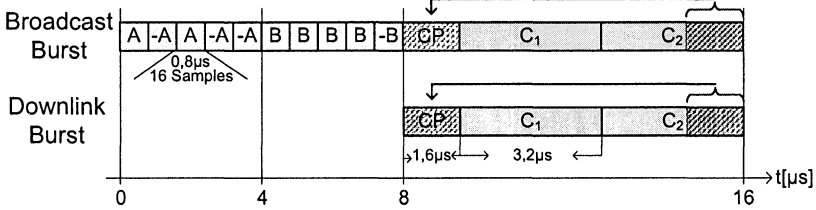


Figure 2. Burst Preambles

transmission. Out of these 52 subchannels, 4 pilot-carriers are scattered among the frequency spectrum and thus 48 subcarriers are used for data transmission. Figure 3 shows the frequency spectrum allocation.

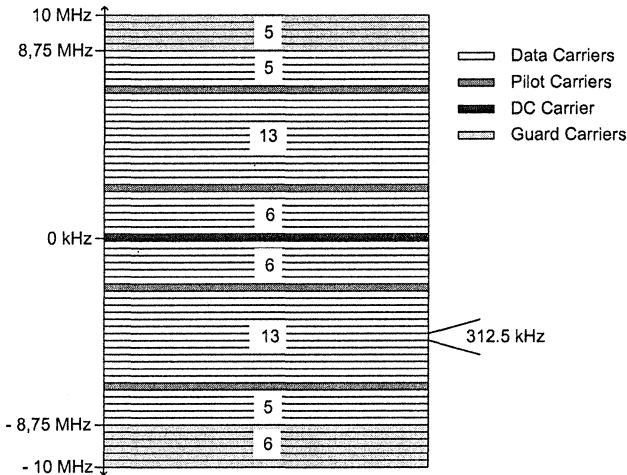


Figure 3. Subcarrier Frequency Spectrum

3. Reconfiguration Capabilities

Reconfiguration always comes at cost of hardware efficiency. Due to the required high computational power of modern wireless communication systems, the use of reconfigurable architectures for wireless systems based on different transmission schemes seems not to be advisable. By restricting the flexibility to function-specific reconfigurable designs, the hardware overhead can be reduced significantly [Zhang and Bodersen, 2000]. Such hardware architectures will be specific to a transmission scheme but not to a special standard. In the

following, reconfiguration capabilities based on WLAN and standard characteristics are analyzed. Notice that computation-intensive tasks like FFT and Viterbi decoding are not considered in the following. Due to the high data rates up to 54 Mbits/s, these tasks are often realized with specialized hardware designs. This specialization hampers the utilization of these hardware blocks for reconfiguration.

Reconfiguration Options based on WLAN Characteristics

The main option enabling reconfiguration is the fact that not all functional blocks within an OFDM receiver are active while processing a burst. During timing and frequency synchronization, no payload processing is done (neglecting a pipelined processing of the last burst). On the other side, while decoding the payload, synchronization has already been achieved. This raises the idea of mapping both tasks onto the same hardware.

Figure 4. Synchronization Block Diagram

Figure 4 shows the block diagram of a synchronization unit. The synchronization process can be divided into two steps. In the first step, the time and frequency values are determined. Therefore the repetition peambles within the received data sequence r have to be detected. This process is based on the computation of the complex correlation

$$S_n = \sum_{m=0}^{N_s-1} r_{n+m} r_{n+m+N_s}^* \tag{1}$$

and the power sum of a synchronization windows of length N_s

$$P_n = \sum_{m=0}^{N_s-1} |r_{n+m}|^2. \tag{2}$$

The hardware structure for the symbol correlator is given in figure 5.

There exist several metrics which use the complex correlation and the power sum to determine the synchronization point and the frequency offset. In this work, a combination of the MNC (Maximum-Normalized-Correlation) scheme [Schmidl and Cox, 1997] and the MMSE (Minimum-Mean-Square-Error) scheme [Chevillat et al., 1987] is used, which was proposed in [Kabulepa et al., 2002]. While the MMSE scheme has shown to be efficient in continuous OFDM applications, it tends to increase false alarm probabilities in burst-oriented OFDM systems when no data frame is transmitted. By applying the MNC scheme to detect the existence of a data frame and using the

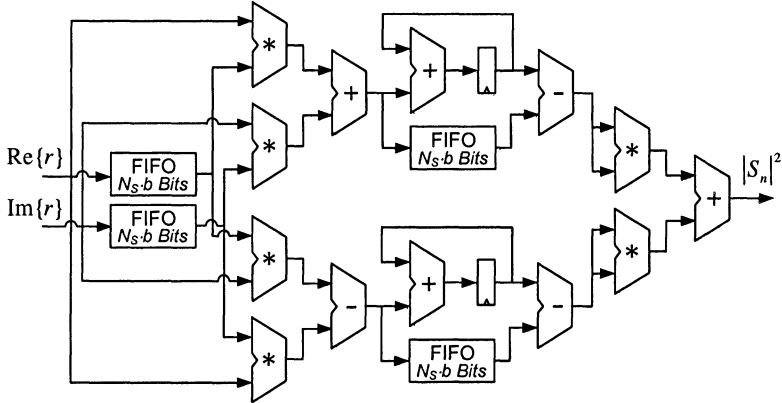


Figure 5. OFDM Symbol Correlator

MMSE scheme to determine the exact synchronization point, it was possible to combine the advantages of both schemes. The hardware structure for this synchronization scheme is given in figure 6.

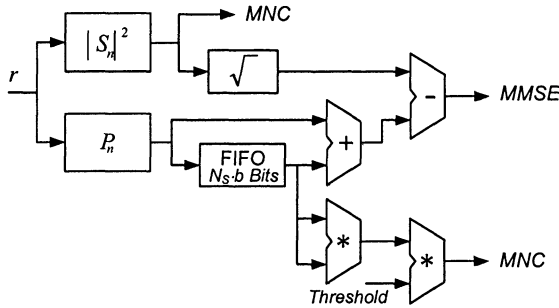


Figure 6. Hardware structure of synchronization scheme

When the synchronization point has been detected, the received data are aligned in a second step. Therefore, FIFOs are used for the time alignment and a CORDIC (COordinate ROTation DIGital COmputer) is used for the frequency correction. The CORDIC is also required for angle calculation during the frequency offset estimation during the first phase. Thus, only the hardware components realizing the synchronization scheme as depicted in figure 4 can be reused as these modules are not required all the time.

Once synchronization has been achieved, payload processing can be started. Focusing on the functional blocks succeeding the FFT, the payload processing

starts with equalization. Equalization (zero-forcing) can be performed by multiplying the received signal $\hat{d}_n(i)$ with the estimated inverse channel transfer function \hat{C}_n^{-1} .

$$\tilde{d}_n(i) = \hat{d}_n(i) \cdot \hat{C}_n^{-1} = d_n(i) + \frac{N_n}{\hat{C}_n} \quad (3)$$

resulting in the transmitted signal $d_n(i)$ and a noise term N_n . The channel estimates \hat{C}_n can be achieved by using the two C-symbols at the beginning of each burst. Equation 4 shows the computation of the channel estimates according to the preamble structure of IEEE 802.11a and HiperLAN/2.

$$\hat{C}_n = \frac{\hat{d}_n^P(0) + \hat{d}_n^P(1)}{2 \cdot d_n^P} \quad (4)$$

In case of a non perfect synchronization, the equalized data may still show a remaining phase rotation. In order to compensate for this impairment, the phases of the pilot-carriers can be monitored. By averaging the phase rotations of the pilot-carriers in an OFDM symbol, a correction value $\tilde{\varphi}(i)$ can be determined which is used to rotate the equalized data. This value can also be used to update the channel estimates, resulting in an improved equalization for succeeding OFDM symbols. Equation 5 presents a method for averaging the phase rotations of the pilot-carriers under consideration of the channel state. N_p gives the number of pilot-carriers, $\tilde{d}_j^P(i)$ are the equalized pilots and $\hat{C}_j^P(i)$ are the channel estimates for the pilot-carriers. A more detailed description of this method can be found in [Pionteck et al., 2003a].

$$\tilde{\varphi}(i) = \frac{\sum_{j=0}^{N_p-1} \arg \left(\tilde{d}_j^P(i) \cdot |\hat{C}_j^P(i)|^2 \right)}{\sum_{j=0}^{N_p-1} |\hat{C}_j^P(i)|^2}. \quad (5)$$

Figure 7 shows the hardware structure of a zero-forcing equalizer with phase rotation compensation. The grey area highlights the components which are used for zero-forcing equalization only. It was possible to avoid the division of the equalized data by the squared absolute values according to equation 3, as the succeeding functional block, the soft value generation, requires a multiplication of the data with this factor.

For the soft value generation, the LLR (Log-Likelihood Ratio) is used as a measure for the reliability of a decision. The sign of the LLR denotes the symbol (± 1) while the absolute value represents the reliability of the decision. Especially for higher modulation schemes, the computation of the LLR function

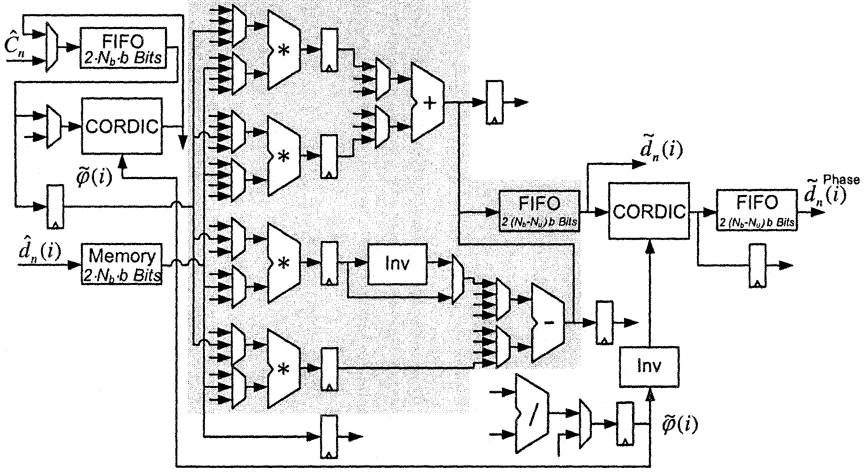


Figure 7. Block structure of the equalizer

is very complex. Therefore, a simplified version is used, which was presented by [Tosato and Bisaglia, 2002]. Let k be the dimension of the modulation scheme and $|G_n(i)|^2$ be the channel state information of subcarrier n . The half distance between the partition boundaries of the constellation for $k > 1$ is given by $a_{k,n}$. According to [Tosato and Bisaglia, 2002], an approximation of the LLR function for a square QAM constellation is given by:

$$D_{k,n}(i) \approx \begin{cases} \tilde{d}_n(i) & k = 1 \\ -|D_{k-1,n}(i)| + a_{k,n}(i) & k > 1 \end{cases} \quad (6)$$

$$LLR(b_{k,n}(i)) = |G_n(i)|^2 \cdot D_{k,n}(i). \quad (7)$$

The corresponding hardware structure for $k \leq 3$ is given in figure 8. The multiplication of $\tilde{d}_n(i)$ with $|G_n(i)|^2$ can be avoided by adapting the equalizer as mentioned before.

The hardware structures of the presented functional blocks show similarities to the symbol correlator and the power sum computation module. Due to the non-overlapping execution times, the use of a common hardware platform for these functional blocks seems to be advisable. The usually introduced hardware overhead through dynamic reconfiguration can be avoided as the re-configuration capabilities are very function-specific. A hardware platform for these functional blocks is presented in section 4.

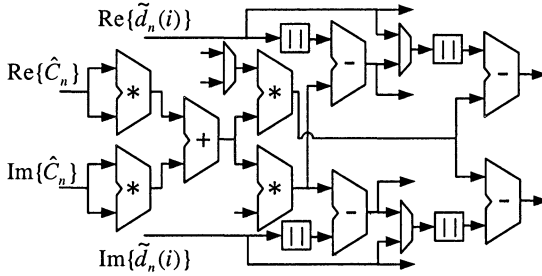


Figure 8. Datapath for generating soft decision values

Reconfiguration Options based on Standard Characteristics

Another way of finding reconfiguration options is the analysis of the standards to be implemented. Focusing on the HiperLAN/2 standard, the different characteristics of the two burst types (BCB and DLB) for a downlink connection can be exploited. The BCB is used to transmit control information to all users while the DLB contains the payload for one specific user. Due to the importance of the BCB, simple modulation schemes are used to make sure that all users are able to decode the control information. In HiperLAN/2, the BPSK modulation scheme is exclusively defined for the BCB, while for the DLB more complex modulations schemes like 64QAM can be used. It is evident that higher modulation schemes require more complex decoding algorithms. As for the BCB only BPSK is used, hardware required for the demodulation of 64QAM is idle. This hardware can be used to improve the performance of the receiver without any costs.

Based on the functional blocks introduced in section 3, the soft values generation can be simplified. For BPSK, the soft information is proportional to the distance from the decision boundary [Tosato and Bisaglia, 2002]. Thus, the hardware blocks for generating the channel state information and soft values are not required for the BCB. All further hardware blocks presented in section 3 are independent of the modulation scheme and thus are required for both burst types.

The freed hardware resources can be used to improve the timing synchronization. OFDM systems are very sensitive to timing synchronization errors [Bölcskei, 2001]. Even a small misalignment leads to a degradation of the system performance. The sensitivity increases with the complexity of the modulation scheme. In the presence of a timing offset Δt , where Δt is an integer multiple of the sampling period, the received signal can be written as

$$\hat{d}_n^{\Delta t}(i) = \sum_{k=0}^{N-1} \hat{s}_i(k + \Delta t) \cdot e^{-j2\pi \frac{nk}{T_s}} \quad (8)$$

$$= \sum_{m=0}^{N-1} \hat{s}_i(m) \cdot e^{-j2\pi \frac{(m-\Delta t)k}{T_s}} = \hat{d}_n(i) e^{j2\pi \frac{\Delta t n}{T_s}} \quad (9)$$

Thus a timing offset results in a phase rotation of the subcarriers, which can easily be detected by monitoring the pilot-carriers. A possibility to estimate a timing offset is to determine the best fit line of the phase rotations of the pilot-carriers. With linear regression, the influence of the channel can be reduced. Figure 9 shows the best fit line for a timing offset of -1. By monitoring the slope of the best fit line during the BCB, a residual timing offset can be detected reliably [Pionteck et al., 2003b]. A receiver can compensate for the estimated timing offset before starting the processing of the DLB. The required hardware effort is marginal. The calculation of the phase of the pilot-carriers, which is the most complex task, is already done in the equalizer when performing phase rotation compensation. For correcting the offset, no additional hardware is required as only memory positions inside the synchronization module have to be updated.

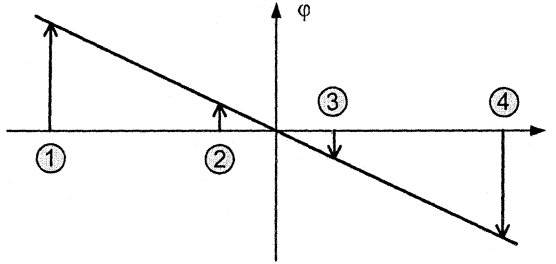


Figure 9. Timing offset estimation with best fit line

The timing offset estimation based on pilot-carriers phase shift can be done according to the following equation:

$$m = \frac{\sum_{i=0}^{N_{BCB}-1} \sum_{j=0}^{N_p-1} (\bar{x} - x_j) \left[\left(\frac{1}{N_p} \sum_{k=0}^{N_p-1} \varphi(i) \right) - \varphi_j(i) \right]}{N_{BCB} \cdot \sum_{k=0}^{N_p-1} (\bar{x} - x_k)^2} \quad (10)$$

where N_{BCB} denotes the number of considered OFDM symbols, N_p gives the number of pilot-carriers, x_j and x_k are the subcarriers at position i, j and

$\varphi_j(i)$ is the phase of j th pilot-carrier of the i th OFDM symbol. As some of the pilot-carriers are rotated, an addition of the phase of these pilot-carriers with π is required before determining the slope of the best fit line. The timing offset estimator can be realized with only a few hardware components. This allows the realization of this optimization method onto the hardware blocks of the channel state information and soft value generation. By means of a dynamic reconfiguration between the BCB and DLB, no additional hardware for the timing offset estimator is required. The division in equation 10 can be avoided by adapting the reference values which are used to determine the size of the timing offset.

4. Hardware Design

The previous section indicates that several possibilities exist where dynamic reconfiguration can reduce the hardware requirements of a receiver within one transmission scheme. These function-specific reconfigurations can be used either as optimizations within a dynamically reconfigurable architecture or for a stand-alone implementation. A possible stand-alone implementation for the usage inside an ASIC is presented in the following.

Figure 10 shows the datapath of the proposed hardware architecture. For the sake of simplicity, not all feedback connections and pipeline registers are shown. Memory blocks which can be used at different positions in the datapath are drawn separately. Each of these memory blocks has a capacity of 52 Bytes, capable of storing the real or imaginary part of a complete OFDM symbol. The remaining memory blocks provide a capacity of 104 Bytes. For the CORDIC elements a word-parallel architecture was chosen. This increases the hardware requirements but leads to a reduced latency for the complete system. The same holds true for the architecture of the divider.

The design comprises all arithmetic blocks and datapaths for realizing the functional blocks discussed in the last section, including the timing offset estimator for the BCB. Dynamic reconfiguration enables the switching between different functional blocks at runtime. By making use of timing and scheduling properties of the different tasks, the functional blocks can be grouped into three different configuration mappings. Figure 11 shows the mapping of the functional blocks onto the proposed hardware design.

mapping A The first mapping realizes the combination of the MMSE and the MNC scheme as described in section 5. According to the numbering in figure 10, the result of the MMSE scheme is provided at position ⑤ and the result of MNC scheme at position ④ and ⑥ of the circuit. Besides memory block 4 and both CORDICs, all hardware and memory elements of the architecture are used.

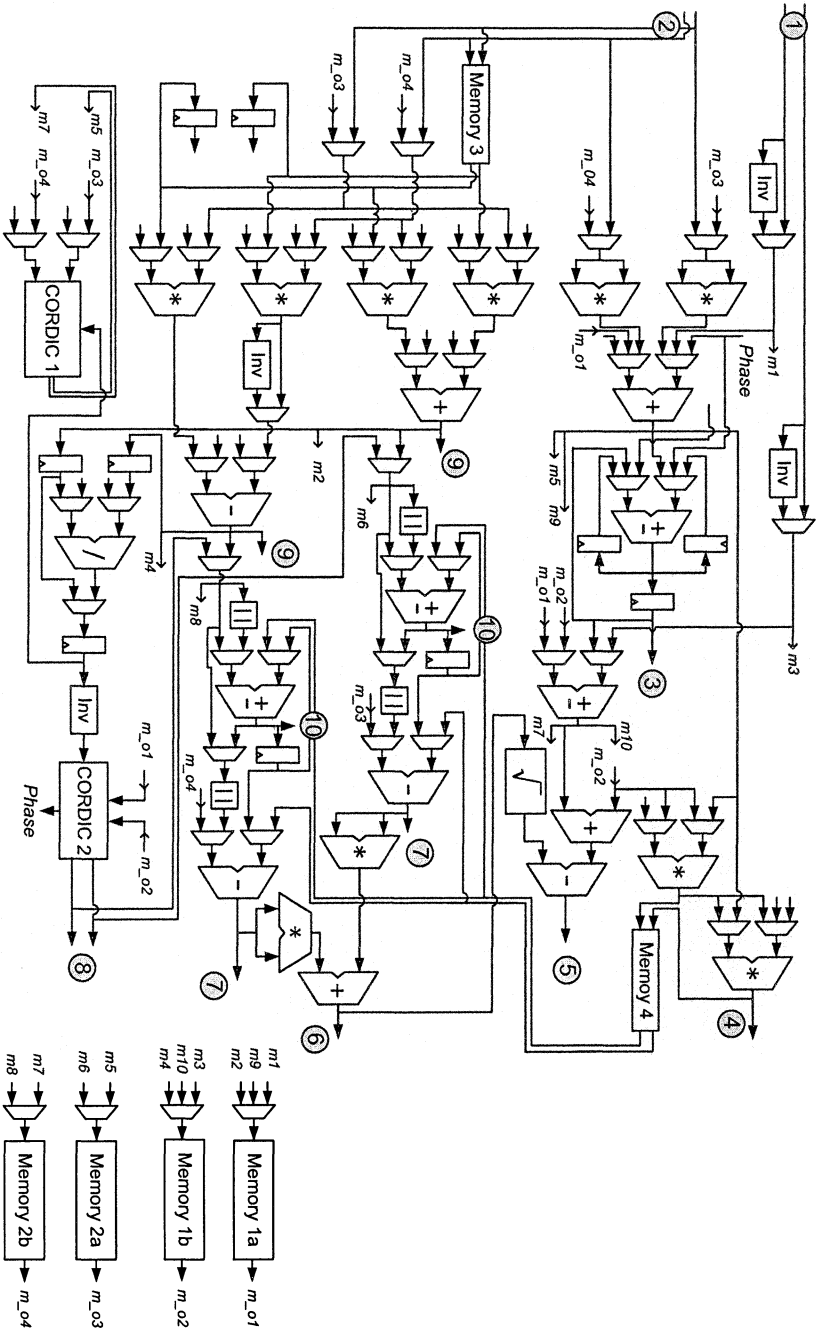


Figure 10. Dynamically Reconfigurable Architecture

mapping B The second mapping realizes the channel estimator, equalizer, channel state information and soft values generator. This mapping can be used for payload processing in HiperLAN/2 (DLB and BCB) and IEEE 802.11a. Modulation schemes up to 64QAM are supported. Input ① provides the C symbols which are used to calculate the channel estimates. These values are stored in memory block 2 for the equalization of the data symbols. The incoming data symbols $\hat{d}_n(i)$ at input ② are stored in memory block 3. In parallel the first data symbols are equalized. As soon as all pilot-carriers have been received, the computation of the correction value for the phase rotation is started. In the meantime, memory block 1 is used as a buffer for the data symbols. The phase compensation of the equalized data is done in CORDIC block 2, while CORDIC block 1 is used to update the channel estimates. The equalized and rotated data symbols are available at position ⑧. If only zero-forcing equalization is to be performed, the result is available at position ⑨. For the soft value generation, the equalized data is fed back to the architecture. According to the used modulation scheme, the soft decision values are provided at position ⑦ or ⑩.

mapping C The third mapping realizes the channel estimator, equalizer and timing offset estimator as described in subsection 3.2. Generation of soft values and channel state information are not required as BPSK is used as modulation scheme. Thus this mapping is specific to HiperLAN/2 as it exploits standard characteristics for the BCB of the downlink connection. The result of the timing offset estimation is available at position ③.

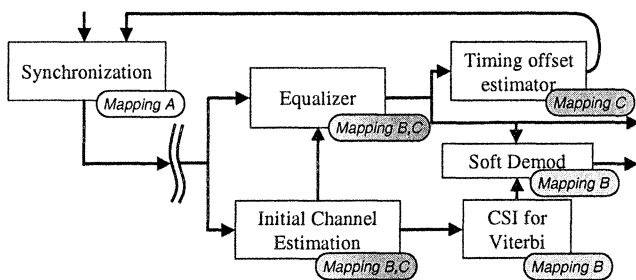


Figure 11. Functional mapping

The configuration memory comprises 48 Bytes and is divided into two sections. In the first section, three different configuration sets can be stored. The

Table 1. Synthesis Results

<i>Modules</i>	<i>Frequency [MHz]</i>	<i>Area (normalized)</i>
synchronization metric	171	15.1
timing offset estimator	31	19.9
channel estimator	534	16.5
equalizer	20	121
channel state inform.	110	3.4
soft value generation	223	2.6
reconfigurable architecture	30	143

second section is used to provide some general settings which are applicable to all of the three configuration sets. These settings include information about the preamble structure (HiperLAN/2 or IEEE 802.11a) and the position of rotated pilot-carriers. The architecture allows a single cycle context-switch between different mappings, as it is required for a continuous data processing during a burst. The hardware architecture is controlled by an external device (e.g., microcontroller), which also manages the reconfiguration process. This external device has to implement functions of the data link layer according to the OSI layer model, as only this layer has knowledge of the type of a burst. The switching between the preamble processing and the payload processing of a burst is controlled by the architecture itself.

The proposed dynamically reconfigurable design was implemented using a $0.35\mu\text{m}$ standard CMOS process with 3 metal layers. Synthesis was done with the Design Analyzer from Synopsys. Table 1 shows the synthesis results of the different functional blocks and of the proposed function-specific dynamic reconfigurable architecture normalized to the area of an eight bits multiplier. The maximum clock frequency of the timing offset estimator, equalizer and the reconfigurable architecture is limited by the critical path inside the unpipelined word-parallel CORDIC. When using a pipelined or word-serial CORDIC, the maximum clock frequency of the reconfigurable architecture is about 100 MHz. Compared to a fixed stand-alone implementation of the presented functional blocks, the proposed dynamically reconfigurable design significantly reduces the hardware effort while providing a higher flexibility than a fixed design. The proposed architecture has a normalized area of about 153. Compared to the normalized area of 178.5 for implementing all functional blocks separately, hardware savings of about 20% could be achieved by using the reconfigurable platform. The hardware overhead introduced by dynamic reconfiguration is very low (about 6%), as only three configurations

have to be stored. Even performance optimizations by improving the timing synchronization were to be realized without increasing the hardware effort.

5. Conclusion

In this work, reconfiguration capabilities for OFDM-based WLANs are explored. By focusing on WLAN and standard characteristics, a set of functional blocks suitable for a dynamic reconfiguration were extracted. The blocks were chosen according to the division of the data processing into preamble and payload processing. The reconfiguration possibilities could be used either for optimization purposes within a dynamically reconfigurable architecture or for a stand-alone implementation. As an example for a stand-alone implementation, the extracted functional blocks were realized onto a function-specific dynamically reconfigurable hardware design, offering significant savings in terms of area. Making use of characteristics in the HiperLAN/2 standard, it was also possible to use the presented hardware design to improve the receiver performance by optimizing the timing synchronization. The work shows that the use of reconfiguration techniques is not only worthwhile for hardware designs for different standards but also for optimizations within a standard. By offering a restricted set of reconfiguration capabilities reduces the overhead introduced by dynamic reconfiguration and eases the use of reconfigurable hardware designs.

6. References

- [Bölcskei, 2001] Bölcskei, H. (2001). Blind Estimation of Symbol Timing and Carrier Frequency Offset in Wireless OFDM Systems. *IEEE Trans. on Communication*, pages 988–999.
- [Chevillat et al., 1987] Chevillat, Pierre R., Maiwald, Dietrich, and Ungerboeck, Gottfried (1987). Rapid Training of a Voiceband Data-Modem Receiver Employing an Equalizer with Fractional-T Spaced Coefficients.
- [Helmschmidt et al., 2003] Helmschmidt, J., Schüler, E., Rao, P., Rossi, S., Matteo, S., and Bonitz, R. (2003). Reconfigurable Signal Processing in Wireless Terminals. In *Date'03 Designers Forum*.
- [Hip, 2001] Hip (2001). Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) Layer, ETSI TS 101 475.
- [IEEE802.11a, 1999] IEEE802.11a (1999). *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*. Institute of Electrical and Electronic Engineers (IEEE).
- [Kabulepa et al., 2002] Kabulepa, Lukusa D., Ortiz, Alberto Garcia, and Glesner, Manfred (2002). Design of an Efficient OFDM Burst Synchronization Scheme.
- [Nee and Prasad, 2000] Nee, R. Van and Prasad, R. (2000). *OFDM for Wireless Multimedia Communications*. Artech House Publishers.
- [Pionteck et al., 2003a] Pionteck, Thilo, Kabulepa, Lukusa D., and Glesner, Manfred (2003a). On the Rapid Prototyping of Equalizers for OFDM Systems. *Design Automation for Embedded Systems*, 8(4):283–295.

- [Pionteck et al., 2003b] Pionteck, Thilo, Kabulepa, Lukusa D., Koppel, Stefan, Garcia, Alberto, and Glesner, Manfred (2003b). Hardware-Efficient Detection and Correction of Timing Offsets in OFDM-based WLANs. *8th International OFDM-Workshop*.
- [Schmidl and Cox, 1997] Schmidl, T. and Cox, D. (1997). Robust Frequency and Timing Synchronization for OFDM. In *IEEE Trans. on Communication*.
- [Tosato and Bisaglia, 2002] Tosato, F. and Bisaglia, P. (2002). Simplified Soft-Output Demapper for Binary Interleaved COFDM with Application to HiperLAN/2. In *IEEE International Conference on Communications*.
- [Zhang and Bodersen, 2000] Zhang, N. and Bodersen, R. W. (2000). Architectural Evaluation of Flexible Digital Signal Processing for Wireless Receivers. In *Asilomar Conference on Signals, Systems and Computers*.