

# OFDM Datapath Baseband Processor for 1 Gbps Datarate

Miloš Krstić, Maxim Piz, Marcus Ehrig, Eckhard Grass  
Wireless Communication Department  
IHP, Im Technologiepark 25,  
15236 Frankfurt (Oder), Germany  
{ krstic, piz, ehrig, grass }@ihp-microelectronics.com

**Abstract—** In this paper we describe the hardware architecture of an OFDM datapath baseband processor capable to support datarates up to 1 Gbps. The processor has innovative streaming architecture that can meet this throughput operating with only 100 MHz clock rate. Baseband processor is completely implemented on a high-speed FPGA platform. We were able to fit the entire transceiver functionality in the set of VIRTEX II pro FPGA chips on the board that we have used. Developed blocks are tested on-board and successfully used in our wireless 60 GHz OFDM demonstrator.

## I. INTRODUCTION

Recent advances in the area of the wireless communications pushed many new applications. It became quite common to have a PC that integrates the features as WLAN, UMTS, Bluetooth, or GPS. However, the application needs are still pushing towards technology limits. There are several standards that are on the way (like IEEE 802.15.3) and requesting much higher datarates of 1 - 2 Gbps. For such high datarates it is foreseen to use the frequency bands in 60 GHz range. We have already developed BiCMOS RF transceiver compliant to 60 GHz band [1]. In this paper we will present hardware architecture of the OFDM baseband processor compatible to this RF front-end. Currently this baseband supports datarates up to 1 Gbps but it is planned that the same architecture is the basis for much higher datarates.

The rapid prototyping of such baseband processing stage must involve high-complexity FPGA platform. The critical part of the OFDM system is convolutional decoding and FFT that require high clock rates not available in FPGA. For example, our system targets 1 Gbps data throughput. With a conventional implementation this will lead to application of Viterbi decoder operating at 1 GHz. However, the Viterbi decoder running at such frequency is unavailable by any FPGA. This performance is difficult to reach even in ASIC with standard non-custom design approach. Additionally, the requirements are constantly growing and there are several research initiatives aiming for up to 10 Gbps data rates in the near future. Therefore, regardless of implementation type (FPGA or ASIC) such systems can be implemented only with highly parallelized structure.

In the following text we will first describe system parameters and structure. Section 2 will describe the general architecture of the baseband processor capable to deal with datarates up to 1 Gbps. After this we will shortly describe main

system components. Section 4 and 5 will present implementation results and demonstrator evaluation. Finally, some conclusions will be drawn.

## II. SYSTEM ARCHITECTURE

In the framework of WIGWAM project [1] we have developed a communication system that supports transmission rate up to 1 Gbps. There were already some experimental studies for Wireless Gigabit LAN (WiGLAN) in 5 GHz band as presented in [3]. However, this solution is not applicable in reality due to the non-availability of the band resources. Until now the main focus for such high data rate was in the area of Ultra Wide Band (UWB) systems, where already some products are available [4]. Additionally, an OFDM solution allowing the real-time transmission of 1 Gbps is provided by IAF [5]. This solution is based on MIMO approach with the parallel use of the several channels supporting usual OFDM transmission scheme. However, this solution also relays on 5 GHz band that is not very much applicable for such high-throughput data transfer.

One feasible alternative can be using of the available spectrum in 60 GHz range. In order to cope with multipath propagation, we decided to use OFDM modulation scheme. Up to our knowledge this is the first implemented OFDM system capable to deal with such datarates in this band.

In some respects, our OFDM based transmission scheme is similar to the 802.11a standard. The same convolutional codes (171,133) are used with fixed transmission modes ranging from BPSK to 64-QAM. As before, the interleaving is performed over one OFDM symbol. The basic OFDM PHY parameters are adapted to a bandwidth of 330 MHz and summarized in [5]. Starting from the expected channel delay spread in the order of up to 20 ns, a guard time in the order of 150 ns of higher is required to support 64 QAM-modulation. To avoid substantial loss of data rate, the guard time should constitute only a small fraction of the OFDM symbol time (e.g. 20% in 802.11a). To lower the impact of the guard time on efficiency, one may rise the symbol duration using a large FFT. On the other hand, this approach is limited, because the resulting smaller subcarrier spacing comes at the price of larger phase noise sensitivity. The cyclic prefix of 160 ns at a FFT period of 640 ns (256 subcarriers in 400 MHz bandwidth) resulting in a symbol time of 800 ns were chosen as a good compromise between phase noise sensitivity and maximum tolerable channel delay spread.

TABLE I. DESIGNATED OFDM SCHEME

FFT bandwidth	400 MHz	Pilot subcarriers	16
FFT size	256	zero subcarriers	5
Subcarrier spacing	1.5625 MHz	Symbol duration	800 ns
Data subcarriers	192	Cyclic prefix	160 ns

The transmission scheme differs from 802.11a in some aspects. First of all, we use a modified and extended preamble structure to allow robust synchronization with low complexity. The detailed synchronization algorithm is described in [6]. The preamble is extended to eleven OFDM symbols to attain good initial synchronization.

To overcome a large overhead, the system must allow the transmission of long frames, hence must track the time varying channel. This is done using decision directed channel estimation in a special way. We employ dual-mode operation to facilitate simple re-estimation of the channel coefficients. After every  $N$  normal data symbols transmitted in some high constellation mode, we insert four data symbols transmitted in a lower constellation mode. These symbols are hereafter referred to as reference symbols. The normal data symbols can be sent with any mode ranging from BPSK to 64-QAM depending on the link budget. On the other hand, the reference OFDM symbols are restricted to BPSK or QPSK in order to always have fixed energy in the subcarriers and to avoid complex division.

To limit the required throughput rate of the decoder, the original data stream is splitted into several streams, which are encoded and interleaved separately. The streams are transmitted interleaved one after each other according to Fig. 1. A time division scheme is used where each OFDM data symbol is assigned to only one stream. In order to enable decision directed channel estimation each stream should be terminated with a zero byte not only at the end of the frame but also in the reference symbols (Fig. 1).

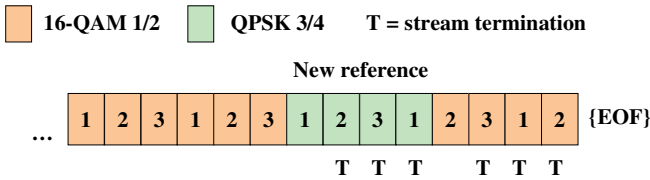


Figure 1. Stream ordering and reference symbols scheme

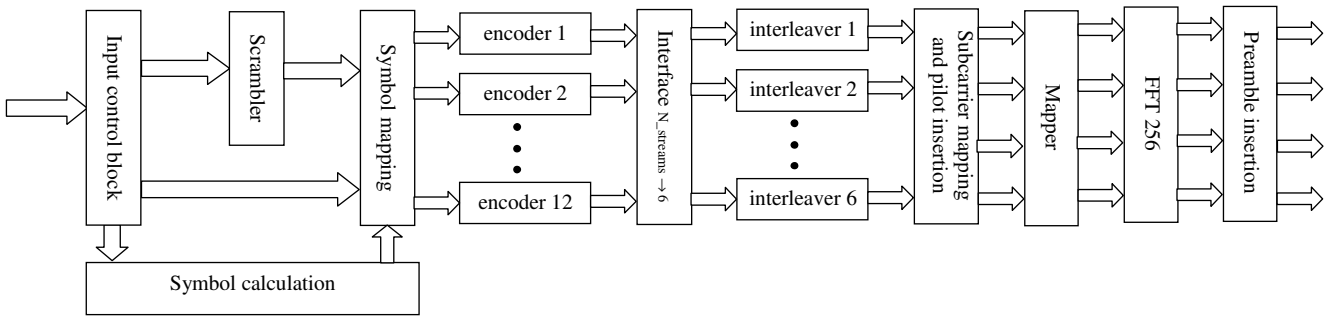


Figure 2. Block diagram of baseband transmitter

### III. BASEBAND PROCESSOR ARCHITECTURE

#### A. Baseband transmitter architecture

The complete baseband transmitter, compliant to the given specifications, is implemented in hardware on the evaluation FPGA board. The block diagram of the implemented transmitter is shown in Fig. 2.

The baseband transmitter is a relatively complex digital circuit. Data from PC is acquired over the input control block. This block also calculates the cyclic redundancy check (CRC) for the signal field and initiates calculation of the number of symbols. Data is delivered to the 32-bit scrambler and further to the symbol mapping block. This block distributes incoming data bytes to the respective symbol. The complete mechanism is fairly complex since for every symbol we have to define several parameters: respective stream of the related symbol, is this a data or reference symbol (and which modulation scheme is used depending on this), and finally is there a termination in this symbol or not. If there is a termination we will insert dummy zero byte at the end of the symbol.

From this point, dataflow is symbol oriented. Depending on the number of streams data from this block is directed to 1 to 12 encoders. According to our calculations six interleavers running at 100 MHz are sufficient to transfer the data with a throughput of 1 GSPS. Therefore, we had to create an interface that redistributes data coming from 1-12 encoders into the fixed structure of six interleavers. Interleaved data is then mapped to the appropriate subcarriers and pilot subcarriers are inserted. Before performing the FFT, data is finally mapped. In order to achieve 400 MSPS effective throughput, we have parallelized operation of the FFT. Finally, at the transmitter backend, data from FFT is multiplexed with the stored preamble symbols.

#### B. Baseband datapath receiver architecture

The implemented receiver is compatible to the presented transmitting scheme. Therefore we have implemented a streaming procedure in the receiver. The main reason for streaming was inability to implement high-speed Viterbi decoder (1 Gbps) in FPGA. Consequently, the complete processing was parallelized into several (in our case 12) moderate speed Viterbi decoders (100 Mbps). In this way we can theoretically support data throughput up to 1.2 Gbps (without any empty processing cycle). In order to simplify the data flow we are not using the full capacity of the decoding section and the maximum throughput used is around 1 Gbps. However, the consequence of this parallelization was necessity to stream the data.

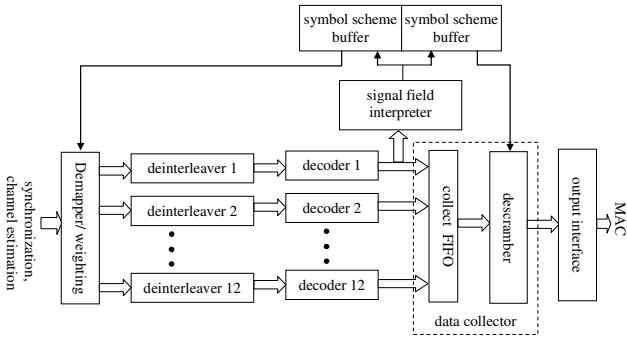


Figure 3. Block diagram of datapath baseband receiver

Fig. 3 shows the datapath baseband processing stage. Our OFDM receiver consists of soft-demapper, deinterleaver stage (with 12 deinterleavers), and 12 Viterbi decoders. After decoding signal field is analyzed in Signal field interpreter to calculate streaming algorithm for current frame and to find positions of reference and termination symbols. The calculated data is then stored in symbol scheme buffers needed for control of the receiver operation. All data transferred from decoders are stored in data collector block. The purpose of this block is to un-stream data and to perform descrambling. Finally, the processed data is delivered to MAC over output interface.

#### IV. MAIN TRANSCIEVER COMPONENTS

##### A. FFT

Our OFDM scheme proposes application of 256-FFT. However, it is required to effectively run this FFT with the clock frequency of 400 MHz. This is not possible by using our FPGA platform, or even using more advanced FPGAs. Therefore we implemented this FFT structure as a combination of the four 64-FFT and 4-FFT. In this way we were able to parallelize the structure with processing 4 samples in on clock cycle with 4 different FFT-64. After that, the output four streams coming out from those 64-FFT are processed in a final 4-FFT stage. With this solution we can process samples arriving with 400 MSPS with a 100 MHz FFT processor. However, we had to pay the price with approximately 4-times increase in area.

##### B. Demapper/Weighting block

The implemented soft-demapper (Fig. 4) calculates bit metrics from received BPSK, QPSK, 16-QAM or 64-QAM symbols. These bit metrics are simplified log-likelihood ratios based on the derivations presented in [7], assuming Gaussian noise. This processing block contains a symbol buffer to overcome possible processing delays caused by the deinterleaver and Viterbi decoder. Four parallel demappers were needed to tackle four incoming parallel data streams coming from the channel estimator. The demapping process produces bit metrics, which are then weighted with the estimated power levels of the corresponding subcarriers. This weighting is part of the bit metric calculation to account for different noise power contributions in the subcarriers after equalization and therefore reflects the varying reliability of different subcarriers received with higher and lower signal strength. After power weighting it is necessary to scale the met-

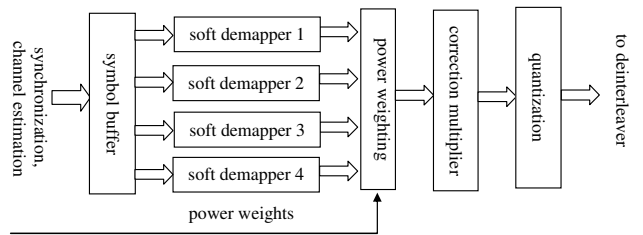


Figure 4. Block diagram of demapper/weighting block

rics with a constant factor for each modulation scheme. With this correction we are able to have quantization degradation lower than 0.5 dB. Finally, data is quantized using rounding. As a result, 5-bit soft values are generated.

##### C. Deinterleaver

The interleavers work with address tables where the patterns for all transmission modes are stored. In this way the interleaving pattern can be changed quickly. Data appears burstwise with soft bits from four subcarriers in parallel. Since 64-QAM is the highest modulation scheme, up to 24 bits are stored per cycle in memory. The deinterleaver performs not only the primary task but also depunctures the code stream by inserting zeros. Therefore a much more flexible puncturing scheme can be implemented, which is not limited to simple repetitive patterns.

##### D. Viterbi decoder

Each Viterbi decoder can process data at a maximum output source rate of 100 MBit/s or one output bit per cycle. The main area consumption results from the input processing block, which gets the two metrics of one source bit per cycle as input data, performs the path metric calculations and decides, which 32 paths out of 64 are survivor paths. The path decisions are stored simultaneously in two dual port block rams, which are denoted as trellis memory. On the opposite side, two logical blocks perform the tracebacks. In the normal case, a traceback is accomplished over 96 steps to deliver 48 output bits. The design doesn't require any wait cycles. Hence the throughput is half a source bit per cycle per traceback unit and the total throughput is one bit per cycle.

A significant reduction in complexity results from the fact that no maximum path metric has to be calculated and tracebacks always can be performed starting from the zero state. This can be achieved when sufficiently long traceback length is used. In this case the path from any initial state will converge with very high probability to the correct path. On the other hand, termination bits at the end of the stream ensure that the zero state is the final state, and with this a priori knowledge the last traceback can correctly start from this state. The output bits during one traceback are assembled in reversed order. Therefore they are stored in a shift register and copied to another shift register when all bits are read. Afterwards they are read out by the output arbiter with a speed of 8 bits per cycle. The output arbiter reads the output of the buffers one after each other. It operates burst-wise in that all bits belonging to a particular traceback have to be read out in conjunction before moving on to the next unit.

In general, termination of the code stream for the purpose of decision feedback channel re-estimation can appear anywhere in the frame and the data stream may be continued afterwards. Therefore the traceback length will vary depending on the termination. The master block generates commands with complete information needed for a traceback. These commands are stored in a FIFO memory and read out by a traceback arbiter who serves the traceback units one after each other. The combined logic of the master and the traceback and output arbiters ensures that termination within the stream is properly handled. One traceback unit can never deliver output data before the other unit is finished with older data. The output bus must be at least two bits wide to ensure the correct operation.

## V. IMPLEMENTATION

As a demonstrator platform we have used the high-speed FPGA development board developed by IAF [2]. The transceiver is fully verified for various channels and SNR ratios in simulation. Afterwards, the complete transceiver is synthesized and mapped to the FPGA. The synthesis summary is given in Table 2. The complexity of the receiver datapath processing unit is around the double of the transmitter due to the high complexity of the Viterbi decoders. However, we are still able to fit both designs in the respective FPGAs.

The work on the project started few years ago when Virtex II pro FPGAs were really state-of-the-art. Today one can find more advanced FPGAs on the market. However, all limitations and challenges that led to our streaming architecture are with such FPGAs just relaxed but not resolved by any means. Therefore, this architecture is applicable independent from the used FPGA hardware platform. For example our estimations showed that change from Virtex II pro to Virtex IV leads to approx. 70% faster circuits. It is expected that Virtex V is approx. twice faster than Virtex II pro designs. It is easy to conclude that this improvement is not sufficient to implement 1 Gbps processing without streaming architecture. In addition to that, we are under development of the second generation of 60 GHz OFDM baseband processor, targeting the datarates up to 10 Gbps. For such cases even ASIC implementation cannot be achieved without streaming architecture similar to the one shown there. Therefore, the methods and architectures described there have general applicability for computationally intensive OFDM baseband processors.

TABLE II. SYNTHESIS RESULTS OF BASEBAND PROCESSOR

block	gates (Mil.)	slices	FF	LUT	BRAM	MULT
Tx	7.8	12430	18069	16375	111	41
Rx	15.5	29142	27450	46632	225	56

## VI. DEMONSTRATOR AND RESULTS

### A. Demonstrator setup

The demonstrator setup is provided in Fig. 5. The complete OFDM baseband transceiver is implemented on board including datapath transmitter and receiver, including inner receiver (synchronization and channel estimation) which is not in the focus of this paper. The demonstrator establishes a transparent TCP/IP link between two PC. The streaming software is able to change the bitrates of the stream. We can also measure the link quality and show the channel parameters like SNR and BER/FER.

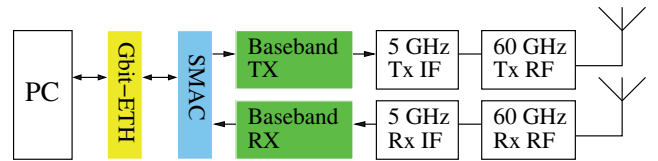


Figure 5. Architecture of 60 GHz OFDM demonstrator

### B. Measurements results

Our 60 GHz OFDM communication system is assembled and tested. For these measurements we have used the complete transceiver implemented on FPGA. The baseband processing is verified first by connecting the analogue baseband transmitter signal to the input of the baseband receiver. We have achieved 36 dB SNR on the receiving side.

To evaluate the system including RF front-end, we have performed the tests based on the simplified baseband hardware platform including the full transmitter implemented in hardware and receiver implemented in MATLAB software. However, the software in this case performs the same algorithms that are implemented later in the hardware version of receiver. Using the full RF 60 GHz link we have achieved error free transmission at a rate of 1080 MBit/s over a short distance with 64-QAM transmission mode.

## VII. CONCLUSIONS

We have described the system architecture of an OFDM baseband processor for extremely high datarates. Since our main goal was rapid prototyping of the system, this baseband processor was implemented on FPGA. However, the similar architecture is very much applicable for the ASIC implementation targeting datarates up to 10 Gbps.

In order to reach high performance requirements we have involved parallelization at several points in the system. This system was successfully tested in simulation and in hardware. To our knowledge, this is one of the first OFDM baseband processors supporting such high datarates.

## ACKNOWLEDGMENT

We are grateful for the support from all partners of the WIGWAM and GALAXY project.

## REFERENCES

- [1] E. Grass, F. Herzel, M. Piz, K. Schmalz, Y. Sun, S. Glisic, M. Krstić, K. Tittelbach, M. Ehrig, W. Winkler, C. Scheytt, R. Kraemer, 60 GHz SiGe-BiCMOS Radio for OFDM Transmission, In Proc. ISCAS 2007, New Orleans, USA, pp. 1979-1982.
- [2] IAF Basic FFP board, datasheet, <http://www.iaf-bs.de/products/ffp-basic.de.html>
- [3] J.K. Tan, K.M. Nguyen, C.G. Sodini, Realization of Baseband DSP core for the Wireless Gigabit LAN, MTL Annual research report, Sep 2006.
- [4] WiQuest, Product Brief, WQST100 and WQST101 with WiDV™ [http://wiquest.com/docs/WQST100-WQST101\\_Product\\_Brief\\_Ver\\_2.3.pdf](http://wiquest.com/docs/WQST100-WQST101_Product_Brief_Ver_2.3.pdf)
- [5] IAF GmbH, Gigabit MIMO OFDM Testbed, <http://iaf-bs.de/projects/gigabit-mimo-ofdm-testbed.en.html>
- [6] Maxim Piz, Eckahard Grass: A synchronization scheme for OFDM-based 60 GHz WPANs, In Proc. PIMRC 2007., pp. 1-5.
- [7] F. Tosato, P. Bisaglia : Simplified Soft-Output Demapper for Binary Interleaved COFDM with Application to HIPERLAN/2, In Proc. ICC 2002, Volume 2, pp. 664 – 668.