A Configurable IP Core for Combined Blind Frequency and Phase Synchronization of MPSK Bursts

Torben Brack, Uwe Wasenmüller, Norbert Wehn
Microelectronic System Design Research Group
University of Kaiserslautern
Erwin-Schrödinger-Straße
67663 Kaiserslautern, Germany
{brack, wasenmueller, wehn}@eit.uni-kl.de

***Abstract*— Frequency and phase correction are vital parts of every inner receiver. Without any training sequence, estimation of the actual frequency and phase offset becomes a very critical task with high impact on communications performance.**

In this paper we present an efficient, configurable, high throughput IP core implementation for combined frequency and phase synchronization on XILINX Virtex II Pro FPGA. The core can be configured with different bit widths, maximum burst lengths, and modulation types. It also provides high flexibility by processing various different burst types during runtime.

We show detailed simulation and implementation results to demonstrate communications performance and hardware complexity of our core.

I. Introduction

Every inner receiver must estimate the general unknown parameters of timing, frequency, and phase offset, and eliminate all possible negative influences introduced by them.

The estimation of these parameters can be done by so called data aided methods, where known symbols are included in the transmitted bursts. Such a training sequence decreases bandwidth and power efficency [1]. The so called non data aided, or blind, methods estimate the parameters with the only help of unknown data symbols. In this case especially estimation of frequency offset becomes a very critical task with high impact on communications performance.

This paper concentrates on the frequency and phase estimation and correction (synchronization) of bursts with Phase Shift Keying modulation for different modulation indices M (MPSK), e.g. $M = 2$ for BPSK and $M = 4$ for QPSK. Frequency and phase offset are estimated without training sequence. The received sample sequence r is given in the complex baseband according to Equation 1:

$$r(l) := s(l) \cdot e^{j(2\pi f_o l + \Phi)} + n(l) \qquad l = 0,1,..,L-1 \quad (1)$$

The sample sequence r with L elements is based on MPSK symbols s with one sample per symbol and symbol duration T, and is disturbed by a noise sequence n. In Equation 1 the frequency offset f_o is annotated as a fraction of the symbol rate $1/T$. The frequency offset f_o and phase offset Φ have to be estimated and corrected. They are considered fixed during one burst transmission.

For non data aided frequency estimation the modulation must be removed from all samples of a burst, followed by a fast Fourier transformation (FFT) to estimate the frequency offset f_o. For blind phase estimation normally the modulation has to be removed again from all samples of the frequency corrected burst, followed by a phase offset estimation algorithm known as the V&V algorithm [2].

In this paper we present an IP Core for combined blind frequency and phase synchronization. First we introduce the used algorithms, followed by architectural considerations, communications performance simulations, and finally implementation aspects and results.

II. Algorithms

This section contains all fundamental algorithms needed for the development of our combined blind frequency and phase synchronization IP core. Further details and resources can be found in [1], [2], [3], [4].

A. Modulation Removal

In general, modulation removal is carried out with a power of M operation on the received sample sequence as shown in Equation 2:

$$\tilde{r}(l) := r(l)^M \quad (2)$$

We proposed an improved modulation removal scheme for BPSK and QPSK in our previous work [5] based on the investigations in [6] to obtain much better communications performance and implementation complexity than the standard approach. These results can be transfered to MPSK modulation without any further modifications. Equation 3 describes this method for removing the modulation:

$$\tilde{r}(l) := |r(l)| \cdot e^{j \cdot M \cdot \arg(r(l))} \qquad l = 0,1,..,L-1 \quad (3)$$

M being the modulation index, and $\tilde{r}$ describing the determined received sequence without modulation. For more details the interested reader is referred to [5]. Here and in most other

operations in this paper the well known polar representation of the received sequence r with $r = |r| \cdot e^{j \cdot \arg(r)}$ is used.

B. Frequency Estimation

The frequency estimation is performed by an FFT algorithm on the sequence $\tilde{r}$, which implements the Discrete Fourier Transformation formula with N points according to Equation 4:

$$X(k) := \sum_{n=0}^{N-1} x(n) \cdot e^{-j\left(\frac{2\pi \cdot k \cdot n}{N}\right)} \qquad k = 0,1,..,N-1 \qquad (4)$$

The sequence x with N elements corresponds to the sequence $\tilde{r}$ with $L \leq N$ elements of Equation 3, which is padded with zeros to achieve the N elements of Equation 4. The estimated frequency offset $\tilde{f}_o$ is given by spectral analysis of the FFT output. In general the FFT bin corresponding to the estimated frequency offset is given by:

$$k_f, \; s.t.: \quad |X(k_f)| = \max_k |X(k)| \qquad k = 0,1,..,N-1 \qquad (5)$$

Note that hereby each bin $X(k)$ represents a range of $(M \cdot N \cdot T)^{-1}$ Hz, not only one distinct frequency. This results in some minor quantization loss introduced by the FFT. The general limit for successfully detecting the frequency offset is $\pm(2 \cdot M)^{-1}$ times the sampling rate. After spectral analysis of the FFT bins, the sample sequence r must be corrected with the calculated frequency offset $\tilde{f}_o$. The corrected sample sequence u is then given by:

$$u(l) := r(l) \cdot e^{-j\left(\frac{2\pi \cdot k_f \cdot l}{M \cdot N}\right)} \qquad l = 0,1,..,L-1 \qquad (6)$$

C. Windowing

If the range of the occurring frequency offset is known in advance, we can significantly improve the communications performance by limiting the spectral analysis. This windowing technique is given by Equation 7:

$$k_{fw} := \max_k |X(k)| \qquad k = 0,..,w_u,w_l,..,N-1 \qquad (7)$$

The parameters w_u and w_l define the frequency range where the frequency offset has to be found. In the following the notation k_{fw} for the FFT bin representing the calculated frequency offset will be used. Setting of w_u to $N/2-1$ and w_l to $N/2$ describes the FFT bin k_f given in Equation 5.

D. Phase Estimation

Phase estimation can be done with the V&V algorithm presented in [2]. In this algorithm the modulation has to be removed from the sample sequence u analogous to Equation 3:

$$\tilde{u}(l) := |u(l)| \cdot e^{j \cdot M \cdot \arg(u(l))} \qquad (8)$$

Then the phase can be estimated using the sequence $\tilde{u}$ according to

$$\tilde{\Phi} := M^{-1} \cdot \arg \sum_{l=0}^{L-1} \tilde{u}(l) \qquad (9)$$

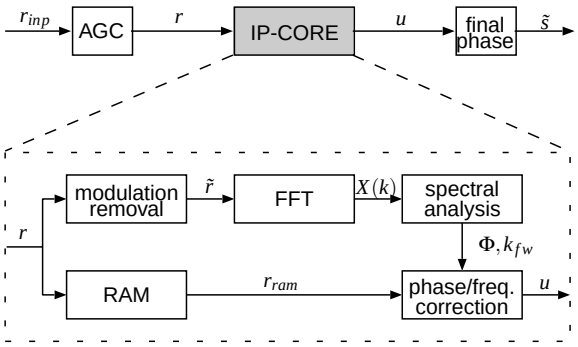

Fig. 1. Base architecture for combined frequency and phase synchronization

Inserting Equation 6 into Equation 8 and further inserting the result into Equation 9 gives an equivalent equation for calculating the estimated phase offset $\tilde{\Phi}$:

$$\tilde{\Phi} := M^{-1} \cdot \arg(X(k_{fw})) \qquad (10)$$

The equation Equation 10 uses the FFT result of equation Equation 4 for the bin k with $k = k_{fw}$. Thus frequency and phase estimation can be efficiently combined.

E. Combined Frequency / Phase Synchronization

As explained in the previous section, the combined blind frequency and phase synchronization uses the FFT result for frequency estimation as well as phase estimation. The frequency estimate is given by Equation 7 and the phase estimate is given by Equation 10. Thus the complex algorithmic steps described by Equation 6, Equation 8, and Equation 9 are combined. The input burst has to be corrected with the estimated values of frequency and phase offset, formally described as:

$$u(l) := r(l) \cdot e^{-j\left(\frac{2\pi \cdot k_{fw} \cdot n}{M \cdot N} + \tilde{\Phi}\right)} \qquad l = 0,1,..,L-1 \qquad (11)$$

III. Architecture

In this section the developed architecture for evaluating the algorithm will be introduced. Figure 1 shows the architecture of the IP Core and the surrounding system.

A. Composition of the IP Core

The components of the IP core architecture are derived straight from the algorithm. The names of the components and connections shown in Figure 1 correspond to the algorithmic parts introduced in Section II.

The modulation removal component, which realizes Equation 3, has to be generated for all modulation indices that have to be independently processable during runtime. The FFT component must be adapted to the maximum burst length allowed in the application, although the actual burst length can be changed during runtime. The spectral analysis component estimates the frequency and phase offset by calculation of Equation 7 and Equation 10. The RAM buffer is used to guarantee constant throughput by hiding the FFT latency. Thus a new sample sequence r can be processed while the already analyzed sequence r_{ram} is being corrected. The phase and frequency correction component realizes Equation 11. All

components are subject to the defined input bit width of the sample sequence r.

We use XILINX IP cores for the implementation of the modulation removal, FFT, and phase and frequency correction components to obtain compact and high-speed processing units. For further implementation details refer to Section V.

B. IP Core Environment

The IP core has to interface with the environment. An automatic gain control unit (AGC) is required as a front-end to adapt the input samples bit width to the IP core input bit width without limiting the dynamic range. Furthermore, timing estimation and matched filtering have to be performed delivering one sample per symbol as required in Equation 1. The back-end must provide some functionality to resolve the M-times phase uncertainty introduced in Equation 10. This final phase correction requires at least one symbol to be known in the transmitted data for a successful correlation with the received data.

IV. Communications Performance

In this section we present communications performance simulations of the chosen architecture implementation based on a bit true C model. We show bit error rate (BER) graphs for different IP core configurations (bit width, maximum burst length) and input bursts (L, M, f_o, Φ, w_u, w_l).

All graphs are obtained using an additive white Gaussian noise (AWGN) channel. A reference graph *ref* is provided to allow comparison of the simulation results to a perfect frequency and phase synchronization reflecting the performance limits of the AWGN channel.

A. Input Bit Width Configuration

Figure 2 shows the impact of different input bit width settings on communications performance. The maximum burst length (MBL) was set to 512 symbols. For comparison purposes, no windowing was used in simulations besides for these in Section IV-D.

The BER graphs show input bit widths n ranging from 4 to 7 bit. There is almost no difference in performance for $n = 7$ and $n = 6$ bit. Using only 5 bit input values ($n = 5$) decreases the performance by around 0.25 dB at high SNR. Reducing the input bit width to $n = 4$ results in a significant degradation on communications performance even on low signal-to-noise ratios (SNR), ranging from 0.5 dB to 1 dB.

B. Maximum Burst Length Configuration

A higher maximum burst length setting influences the needed FFT component, demanding higher frequency offset resolution and therefor an FFT capable of supporting more sample points. The number of FFT points has to be around two times the maximum burst length to prevent major degradation in communications performance [3]. By choosing a higher maximum burst length as really needed in a specific application, we can slightly improve the overall communications performance because of the resolution gain.

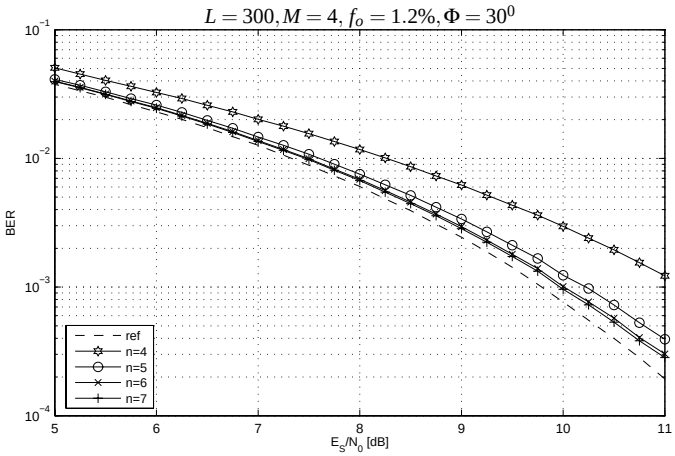

Fig. 2. Impact of different input bitwidth settings

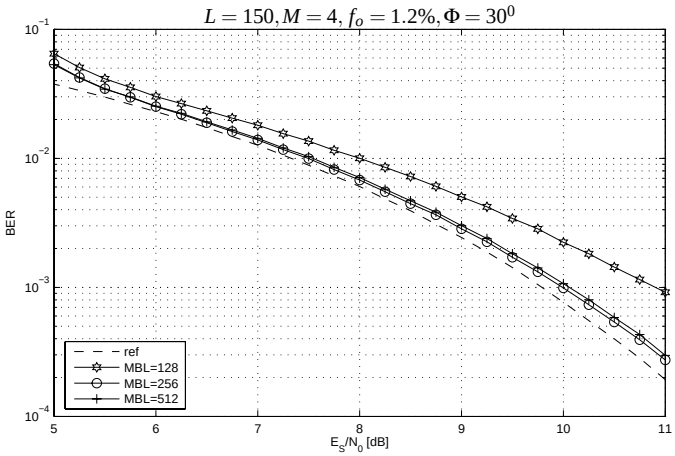

Fig. 3. Impact of different maximum burst length settings

The simulations presented in Figure 3 were performed with 6 bit input data width and MBL settings of 128, 256, and 512 symbols. The improvement for a 150 symbol burst using an oversized MBL of 512 symbols is only 0.1 dB at max for high SNR. Besides, the number of FFT points needed has a significant impact on the size of the final IP core (see Section VI). A too small MBL setting of only 128 symbols shows the predicted major loss in communications performance from 0.5 dB to over 1 dB for high SNR.

C. Modulation Type Configuration

Although the actual modulation index M has a huge impact on communications performance, the IP core configuration of usable modulation types has no influence on communications performance at all. Every selected modulation index prior to core generation can be processed with the best results possible, independently from any other selected indices.

D. Windowing

In Equation 7 we presented the windowing technique for spectral analysis. Figure 4 shows simulation results for an MBL of 256 symbols and variable window sizes. The graphs represent windows of 1.5%, 3%, and 6% of the sample

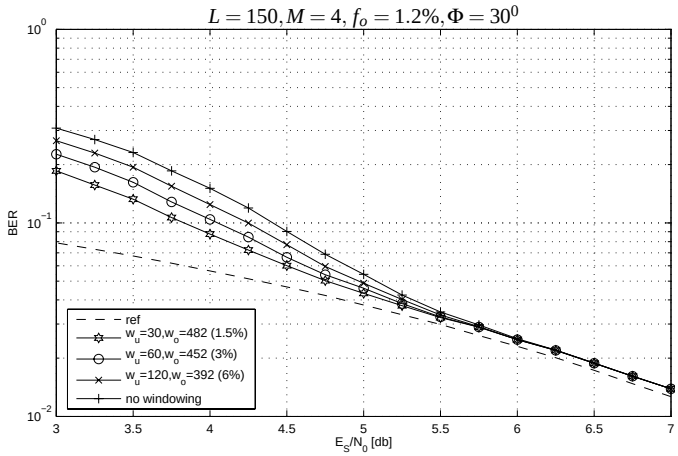

Fig. 4. Impact of different window sizes

rate ($1/T$), and results without windowing for comparison. The actual frequency offset f_o is fixed on 1.2%. There is a significant performance gain for low to medium SNR by around 0.25 dB for the largest window (6%), and respectively an additional 0.25 dB gain for the window half and quarter this size.

V. Implementation

We used synthesizable VHDL for modeling and implementing the architecture shown in Figure 1. For rapid development and to minimize debug effort we heavily relied on IP cores [7] included in the XILINX Core Generator 6.1. We also used specific XILINX resources like the internal multipliers (MULT) and block RAM (BRAM) available on the Virtex II Pro FPGA.

The modulation removal in Equation 3 uses polar coordinates. The input samples r however are given in Cartesian coordinates. The calculation of $\arg(r)$ and $|r|$ can be done with the CORDIC algorithm [8] which is available as an IP core [9]. We utilized the fully pipelined version of this CORDIC IP core to achieve a throughput of one sample per cycle. Afterwards the resulting sample sequence $\tilde{r}$ without modulation has to be transformed back to Cartesian coordinates for the FFT calculation (see Equation 4). This is accomplished with a sine-/cosine-look-up-table (SCL) [10] also available as an IP core and internal multipliers to realize the e^{jx} operation.

The FFT needed for frequency and phase estimation is realized using another XILINX IP core [11] configured to sustain the selected maximum burst length. Because again the desired throughput was one sample per cycle, the resulting core is fully pipelined and covers most of the area.

The windowing feature proposed in Equation 7 for evaluating k_{fw} is realized in the spectral analysis block, using a limited maximum absolute value search on the determined FFT bins. This can be implemented in a very efficient way by using only counters and comparators.

After determining the frequency bin k_{fw}, the spectral analysis block estimates the phase offset according to Equation 10. Again, the argument function needed for this equation is realized with a CORDIC IP core. Because only one argument calculation per phase estimation has to be performed, we can use a rather small serial CORDIC core without the ability to calculate $|r|$.

Rotation of the samples as described in Equation 11 is accomplished with complex number multiplications and a second SCL.

Component	Cores	MULT	BRAM
modulation rem.	SCL, CORDIC	2	1
FFT (MBL=8)	2^4 point FFT	18	3
FFT (MBL=512)	2^{10} point FFT	18	27
FFT (MBL=8192)	2^{14} point FFT	18	54
RAM (MBL$<$ 512)	RAM	0	1
RAM (MBL=8192)	RAM	0	16
spectral analysis	CORDIC	2	0
phase/freq. correction	SCL	5	1

TABLE I

Implementation resources per component

BW	MBL	Slices	BRAM	MULT	MHz
7	64	2926	20	18	119.9
7	512	3636	30	27	119.4
7	4096	5018	78	36	104.0
6	64	2863	20	18	125.3
6	512	3548	30	27	123.5
6	4096	4964	78	36	116.1
5	64	2816	20	18	128.4
5	512	3505	26	27	126.3
5	4096	4931	78	36	116.0

TABLE II

Synthesis results

Table I summarizes all FPGA resources needed for the implementation of every single architectural component. Note that the characteristics of the FFT and RAM components vary with the maximum burst length (MBL) selected, thus we can only present some representative results.

VI. Results

We used XILINX ISE 6.1 for synthesis and place and route with the XILINX Virtex II Pro FPGA XC2VP50 as target platform. Table II shows the results for different IP core configurations.

As expected, the implementation complexity grows strongly with increasing MBL setting due to the huge portion of the FFT component. Decreasing input bit width BW yields only minor savings instead, because mainly the relative small modulation removal block is affected. The maximum clock frequency is also somewhat correlated with the MBL and BW settings, increasing with smaller bit widths and decreasing with a higher MBL. The resulting maximum sample throughput is about half the clock speed because the FFT latency is at least twice the maximum burst length (see Section IV-B). Thus the corresponding bit rate is determined by the modulation:

$$throughput[{}^{\text{bits}}/_{\text{sec}}] \approx (2 \cdot cycle)^{-1} \cdot log_2(M) \qquad (12)$$

For example, using QPSK modulation Equation 12 evaluates to a bitrate beyond 100Mbps for every configuration.

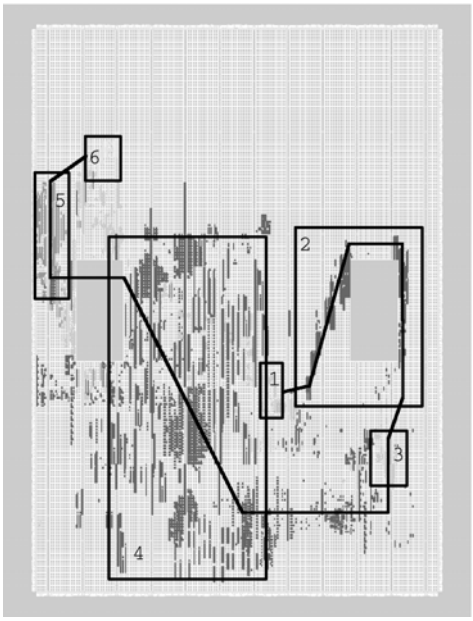

Fig. 5. Floorplan of the synthesized architecture

Such high throughput in FPGA can only be achieved using high effort speed optimized place and route, resulting in a floorplan where the dataflow is strongly reflected through the placement of the embedded cores as shown in Figure 5: All samples are first stored in the small Dual-Port RAM component (1), then fed into the first longly stretched CORDIC for polar transformation (2), and then multiplied with the selected SCL entries for modulation removal (3). The following, very big and scattered FFT component (4) calculates the spectrum, and the second smaller CORDIC takes on the spectral analysis (5). Finally the samples are corrected with the second SCL (6).

VII. Conclusion

In this paper we presented, to the best of our knowledge, the first published configurable IP core for combined blind frequency and phase synchronization of MPSK bursts, utilizing advanced techniques like improved modulation removal and windowing. These features yield high throughput, compact size, and best communications performance achievable for transmissions without training sequence.

References

[1] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Commucication Receivers*. New York: John Wiley & Sons, Inc., 1997.
[2] A. J. Viterbi and A. M. Viterbi, "Nonlinear Estimation of PSK Modulated Carrier Phase with Application to Burst Digital Transmission," *IEEE Transactions on Information Theory*, vol. 32, pp. 543–551, July 1983.
[3] U. Mengali and A. D'Andrea, *Synchronization Techniques for Digital Receivers*. New York: Plenum Publishing Corporation, 1997.
[4] H. L. van Trees, *Detection, Estimation and Modulation Theory*. New York: John Wiley & Sons, Inc., 1968.
[5] T. Brack, U. Wasenmüller, D. Schmidt, and N. Wehn, "Design Space Exploration for Frequency Synchronisation of BPSK/QPSK Bursts," in *Kleinheubacher Berichte*, vol. 48, Sept. 2004.
[6] Y. Wang, E. Serpedin, and P. Ciblat, "Optimal Blind Carrier Recovery for MPSK Burst Transmissions," *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1571–1581, Sept. 2003.
[7] "Xilinx Inc." `www.xilinx.com/ipcenter`.
[8] J. Volder, "The CORDIC Trigonometric Computing Technique," *IRE Transactions on Electronic Computing*, vol. 8, pp. 330–334, Sept. 1959.
[9] "Xilinx Inc. CORDIC v2.0," Mar. 2003.
[10] "Xilinx Inc. Sine/Cosine Look-Up Table v4.2," Nov. 2002.
[11] "Xilinx Inc. Fast Fourier Transform v2.0," Mar. 2003.