Design Space Exploration for Frequency Synchronization of BPSK/QPSK Bursts

T. Brack, U. Wasenmüller, D. Schmidt, and N. Wehn

Microelectronic System Design Research Group, University of Kaiserslautern, Erwin-Schrödinger-Straße,
67663 Kaiserslautern, Germany

Abstract. Frequency synchronisation is a vital part of every inner receiver for wireless communication. In this paper we present different implementation alternatives for non data aided frequency estimation of BPSK/QPSK bursts with respect to implementation complexity and communications performance. Results with regard to different quantization levels, varying burstlengths, frequency offsets and modulation indices for different signal to noise ratios are presented. Implementation results are based on XILINX Virtex II Pro FPGA devices.

1 Introduction

Frequency synchronisation is a vital part of every inner receiver for wireless communication. In the system under consideration BPSK/QPSK bursts are provided in the complex baseband without training sequences. This is typical for many state of the art communication systems, because training symbols decrease bandwidth and power efficiency (Meyr et al., 1997). Hence so called non data aided frequency estimation algorithms become mandatory. First the modulation is removed from all samples of a burst, followed by a fast Fourier transformation (FFT) to estimate the frequency.

In this paper we assume an input sample sequence r with L elements based on BPSK/QPSK symbols s with duration T which has a fixed frequency offset f_o, phase error Φ, and is disturbed with noise sequence n. Hence:

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

We further assume $r(l)$ with one sample per symbol and ideal timing. All examinations of this paper are restricted to frequency synchronization and hence we assume the phase offset $\Phi := 0$. State-of-the art modulation removal is based on a power of M operation on r (Mengali and D'Andrea, 1997),

Correspondence to: T. Brack
(brack@eit.uni-kl.de)

where M is the modulation index i.e. 2 for BPSK and 4 for QPSK respectively.

$$\tilde{r} := r^M = |r|^M \cdot e^{jM \cdot arg(r)} \qquad (2)$$

Frequency estimation is now performed on $\tilde{r}$. An approximation for the maximum likelihood estimation of the frequency is the spectral power calculation of the samples $\tilde{r}$ by use of a fast Fourier transformation (Mengali and D'Andrea, 1997). Since the power operation for modulation removal enhances the noise in the signal superproportional, so called self noise is inferred. This yields large frequency estimation errors especially at medium to low signal to noise ratio ranges. Thus bursts with high bit error rate and therefore high frame error rate after channel decoding result. Thus in this paper we use another approach for removing the modulation (Wang et al., 2003; Viterbi and Viterbi, 1983). This technique for modulation removal is based on the following calculation:

$$\tilde{r} := |r|^k \cdot e^{jM \cdot arg(r)} \qquad k = 0, 1, .., M \qquad (3)$$

Thus k is an optimization parameter ranging from 0 to modulation index M. Observe that for $k = M$ Eq. (3) transforms into Eq. (2). This method reduces the self noise inference and is superior to the power based modulation removal technique. Different possibilities exist to implement Eq. (3). In the following sections we investigate the corresponding design space and trade off communications performance versus implementation complexity with respect to various parameters.

2 Impact on communications performance

In this section we will show the impacts of different quantization levels, varying burstlengths, frequency offsets and modulation indices with respect to the optimization parameter k on communications performance. The architecture of the frequency synchronization is shown in Fig. 1.
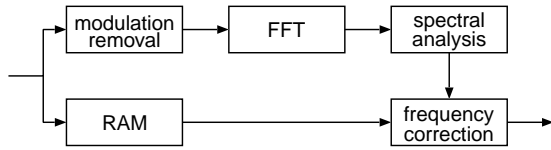
Fig. 1. Base architecture for frequency synchronization.

All performance charts are obtained using bit-true C-models, which are a reference for our hardware realizations. Input data is sampled with 12 bit from an additive white gaussian noise (AWGN) channel and then fed into an automatic gain control (AGC) unit with selectable quantization to fit the chosen quantization used in the frequency synchronization. The reference curves are obtained using perfect frequency synchronization, reflecting the transmission limits of the AWGN channel. These reference curves are denoted as *ref* in all following bit error rate (BER) graphs.

2.1 Quantization level impact

The quantization level affects directly the allocated resources and throughput of the FPGA hardware. To evaluate the communications performance degradation due to quantization we compare the bit-true results with a floating point MATLAB model. The quantization can be selected in all our examinations.

There are extremly subtle differences in behaviour for the varying examined parameters like burstlengths and modulation index. We limit our demonstration to one set of representative parameters ($k=1$, QPSK, frequency offset=1.2% of sample rate, and burstlength=50 symbols). The quantization levels in Fig. 2 range from 5 to 8 bit, because smaller values were not acceptable in all examinations and higher values show no difference to the floating point simulation. This can be exploited to minimize the size of the final design.

The following explorations of the other parameters are performed using a fixed quantization of 9 bit to eliminate any quantization effects.

2.2 Modulation index variation

We examined the influence of the modulation index M on communications performance taking the optimization parameter k into account. Our main goal was to derive the optimal value of k for the two modulation indices discussed here. Again we chose a burstlength of 50 symbols and a frequency offset of 1.2% for this simulation, because the impact of modulation index variations can best be observed with small burstlengths and is relatively constant over a wide range of frequency offsets (see Sect. 2.4). Figure 3 shows the results for BPSK ($M=2$) for all values of k, Fig. 4 for QPSK ($M=4$) respectively. Because BPSK obviously outperforms QPSK modulation we chose a different SNR range in these figures. We observe that $k=1$ significantly improves communications performance for both BPSK and QPSK modulation
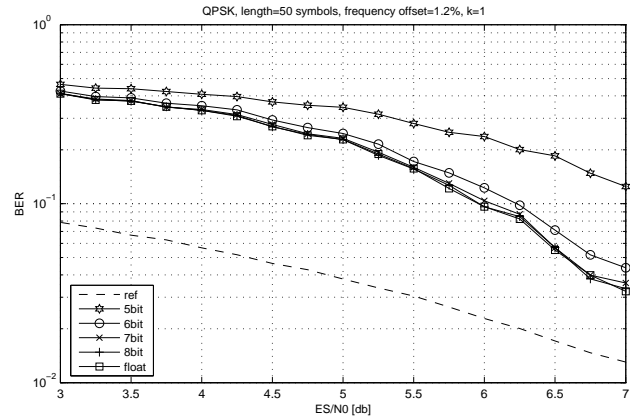

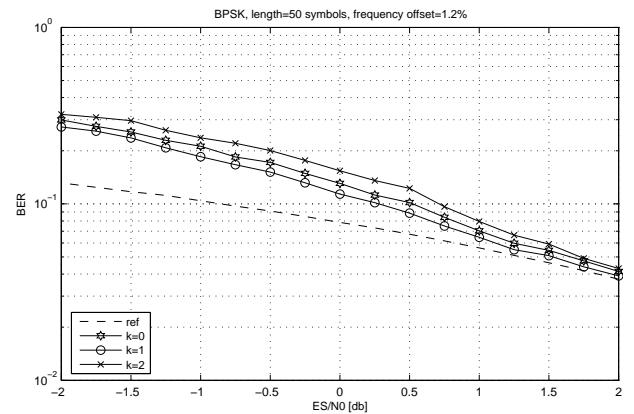**Fig. 2.** Impact of different quantization levels.

Fig. 3. BER of BPSK modulation for different values of k.

in comparison to the standard approach using $k=M$. The actual gain is about 0.8 dB.

2.3 Burstlength variation

The impact of different burstlengths is investigated in the following. We chose 3 different burstlengths of 50, 150 and 300 symbols to reflect a sufficient range, and a fixed frequency offset of 1.2%. Here we only present QPSK results because BPSK shows the same general behaviour (see Sect. 2.2). Again our focus is put on the optimization parameter k. Results for 50 symbols bursts were already presented in Fig. 4. Figures 5 and 6 show results for the burstlengths of 150 symbols and 300 symbols respectively. Larger burstlengths generally improve the quality of frequency estimation (Mengali and D'Andrea, 1997; van Trees, 1968); this is verified by the shown results. Again $k=1$ shows the best performance on all burstlengths. This trend holds for even smaller and even larger burstlengths, and the maximum gain is around 0.8 dB in all examined cases.
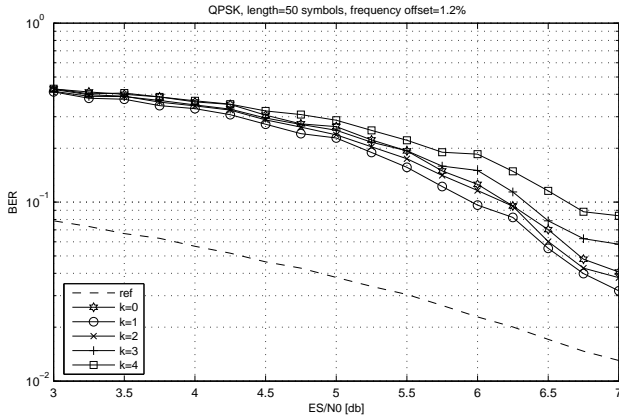
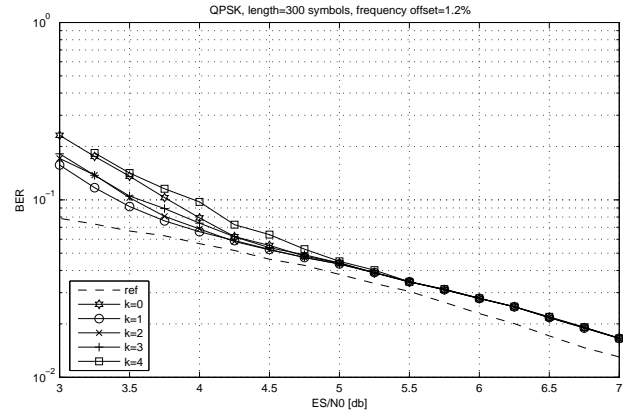**Fig. 4.** BER of QPSK modulation for different values of k.

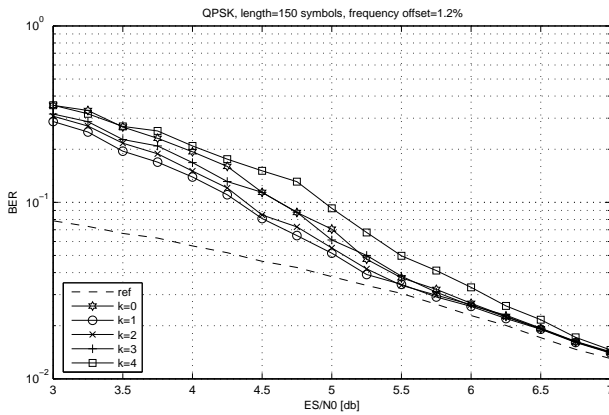**Fig. 6.** BER of 300 symbols QPSK bursts for different values of k.

Fig. 5. BER of 150 symbols QPSK bursts for different values of k.

Fig. 7. Deviation of found frequency offset for perfect match (left) and mismatch (right).

2.4 Frequency offset variation

Due to the use of FFT there exists a general limit for successfully detecting frequency offsets at $\pm(2 \cdot M)^{-1}$ times the sampling rate. In this range there are two effects accountable for slight variations in detection performance. One is the degradation with increasing frequency offset, deteriorating while approaching detection limit. This effect is only measurable in very low SNR. It ranges in a magnitude of 10^{-4} of BER, thus causing no impact on overall performance. The other effect is introduced by the inherently limited resolution of the FFT and discussed in detail in the following subsection. In Sect. 2.4.2 we present a method to improve the communications performance if the range of the occuring offset is known in advance.

2.4.1 FFT quantization effects

Due to the FFT algorithm the frequency offset can only be detected within discrete intervals (bins) depending on the number of points N of the FFT, the sampling rate f_s, and the modulation index M. The number of points of the FFT
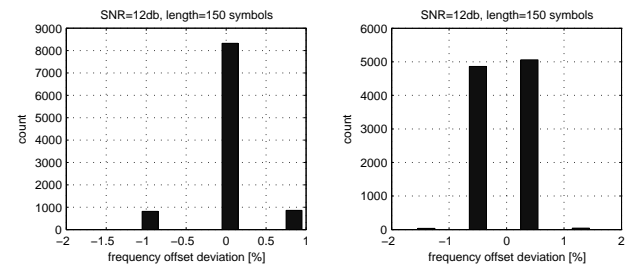
algorithm must be a power of 2, thus $N=2^n$. The resolution Δ of the FFT is $\Delta = f_s/(N \cdot M)$.

The number of points of the FFT has to be larger than the maximum supported blocksize, and smaller than FPGA resources allow. Between these two extremes the finally selected number of points is determined through the needed resolution.

The histograms of 10 000 simulations in Fig. 7 show what happens if the current frequency offset can be perfectly matched with an FFT bin (left) and lies exactly between two bins (right) respectively. In the first case the frequency offset is almost perfectly detected, in the second the frequency offset is detected either too small or too large because no bin can be definitely assigned to the current frequency.

Nonetheless the impact on overall performance in terms of BER is very subtle. We used an FFT with 1024 points in our simulations and implementations without performance degradation for any given frequency offset in comparison to an FFT with a larger number of points.

2.4.2 Improvement through windowing

If the range of the occuring frequency offset is known in advance, one can gain more than 1 dB with a simple technique called windowing. It is based on limiting the spectrum analysis to specific FFT bin ranges and can be easily integrated in the existing unit without noticeable hardware overhead.
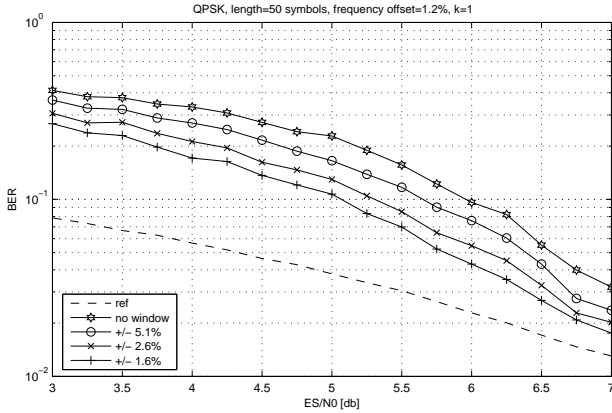
Fig. 8. Influence of different frequency windows on BER.

The effect on bursts with 1.2% frequency offset is shown in Fig. 8 for different frequency windows. Without windowing the performance complies to a frequency window of $\pm12.5\%$ as described in Sect. 2.4.

3   Implementation

We used synthetizable VHDL for modelling the architecture shown in Fig. 1. For rapid development and to minimize debug effort we relied heavily on IP cores (Xilinx Inc.) 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 only important global optimization parameter is k in Eq. (3). Therefore variations of this parameter only affect the modulation removal block, while the other units in Fig. 1 are not affected by the parameter k.

The implemented architecture can handle variable block-sizes of up to 1024 symbols and BPSK as well as QPSK modulation formats and supports all window sizes up to the full possible range.

3.1   Frequency estimation and correction

The 1024-point FFT (see Sect. 2.4.1) needed for frequency estimation is realized using an XILINX IP-core (Xilinx Inc. Fast Fourier Transform). Because the desired throughput was one sample per cycle, the resulting core is fully pipelined and thus rather large and takes most of the area needed for the whole frequency synchronization implementation.

The spectral analysis block including the windowing feature mainly consists of a limited maximum absolute value search on the determined FFT bins and can be implemented with some multipliers and comparators.

The possibility to rotate the samples in the frequency correction component is accomplished with complex number multiplications and a sine-/cosine-look-up-table (SCL) (Xilinx Inc./Sine/Cosine Look-Up Table).

Table 1. Selected alternatives for modulation removal and their resource requirements.

Alt.	Equation	Cores	MULT
0	$\alpha := e^{j4arg(r)}$	aCORDIC, SCL	0
1	$\lvert r \rvert \cdot \alpha$	CORDIC, SCL	2
2a	$r^4/(\Re(r)^2 + \Im(r)^2)$	DIVIDER	10
2b	$\alpha(\Re(r)^2 + \Im(r)^2)$	aCORDIC, SCL	4
2c	$\lvert r \rvert^2 \cdot \alpha$	CORDIC, SCL	3
3	$\lvert r \rvert^3 \cdot \alpha$	CORDIC, SCL	4
4	r^4		6

3.2   Alternatives for modulation removal

Equation (3) uses polar coordinates. The input samples r however are given in cartesian coordinates. The calculation of $arg(r)$ and $\lvert r \rvert$ can be done with the CORDIC algorithm (Volder, 1959) also available as an IP-core (Xilinx Inc. CORDIC). The results then have to be transformed to cartesian coordinates again, using another SCL.

The use of a CORDIC module is not necessary for $k=M$. In this case Eq. (3) can be calculated by a power of M operation using 3 multipliers for BPSK and 6 multipliers for QPSK modulation.

For $M=4$ and $k=2$ algebraic transformations of Eq. (3) lead to $\tilde{r}=r^4/\lvert r \rvert^2$. In this case instead of using a CORDIC module one can use 10 multipliers and a divider. As a third alternative for $k=2$, we can use a simplified CORDIC module (aCORDIC) that only calculates $arg(r)$ but not $\lvert r \rvert$. Then we have to calculate $\lvert r \rvert^2 = \Re(r)^2 + \Im(r)^2$ using two additional multipliers.

Table 1 lists the IP cores and multipliers needed for the alternative realizations for QPSK modulated bursts. We always chose the best alternative for BPSK that can be implemented without using additional resources, hence $k=0$ for alternative 0, $k=2$ for 2a and 4, and $k=1$ else.

4   Results

For synthesis of the architecture we used the XILINX ISE 6.1 suite. The target platform was XILINX Virtex II Pro FPGA (Xilinx Virtex II Pro Data Sheet).

All synthesis results are obtained after place and route, with settings for high-effort area-optimization. Table 2 shows the allocated hardware resources separated in slices, internal multipliers (MULT) and BRAM blocks for all implemented alternatives (see Table 1).

The differences in needed resources are rather small, the maximum overhead is only around 6% between alternative 1 and the state of the art implementation 4. Taking the communications performance results from Sect. 2 into consideration, the application of this slightly more complicated approach is very profitable.

Table 2. FPGA synthesis results.

Alt.	Slices	BRAM	MULT
0	4263	31	25
1	4321	31	27
2a	4287	30	35
2b	4280	31	29
2c	4315	31	28
3	4319	31	29
4	4069	30	31

5 Conclusion

In this paper we have performed a design space exploration for frequency synchronisation with different alternatives for modulation removal. Compared to the state of the art approach a gain of 0.8 dB in communications performance with an overhead of only 6% in hardware resources was achieved. If the target application allows the use of windowing as described in Sect. 2.4.2, an additional gain of up to 1 dB can be reached without increasing hardware demands.

6 Future Work

In future work we will present simulation and implementation results for a complete demodulator architecture. The one currently examined in our work group embeds timing correction, automatic gain control and joint frequency and phase-correction capability, leading to a very compact high-throughput architecture.

Further examination of the introduced windowing approach, especially using adaptive techniques, could increase the gain in communications performance even more.

References

Mengali, U. and D'Andrea, A.: Synchronization Techniques for Digital Receivers, Plenum Publishing Corporation, New York, 1997.

Meyr, H., Moeneclaey, M., and Fechtel, S.: Digital Commucication Receivers, John Wiley & Sons, Inc., New York, 1997.

van Trees, H. L.: Detection, Estimation and Modulation Theory, John Wiley & Sons, Inc., New York, 1968.

Viterbi, A. J. and Viterbi, A. M.: Nonlinear Estimation of PSK Modulated Carrier Phase with Application to Burst Digital Transmission, IEEE Transactions on Information Theory, 543–551, 1983.

Volder, J.: The CORDIC Trigonometric Computing Technique, IRE Transactions on Electronic Computing, 330–334, 1959.

Wang, Y., Serpedin, E., and Ciblat, P.: Optimal Blind Carrier Recovery for MPSK Burst Transmissions, IEEE Transactions on Communications, 51, 1571–1581, 2003.

Xilinx Inc., www.xilinx.com/ipcenter.

Xilinx Inc. Fast Fourier Transform v2.0, 2003.

Xilinx Inc. Sine/Cosine Look-Up Table v4.2, 2002.

Xilinx Inc. CORDIC v2.0, 2003.

Xilinx Virtex II Pro Data Sheet.