Adv. Radio Sci., 4, 313–318, 2006
www.adv-radio-sci.net/4/313/2006/
© Author(s) 2006. This work is licensed
under a Creative Commons License.

Advances in Radio Science

From algorithm to implementation: a case study on blind carrier synchronization

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

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

Abstract. Increasing chip complexities demand a higher design productivity. IP cores, which implement commonly needed operations, can help to dramatically shorten development and verification times for new designs. They often allow for a efficient mapping of algorithmic tasks to a hardware architecture. In this paper we present a novel configurable building block for blind carrier synchronization that features combined frequency and phase offset estimation and an alternative modulation removal that improves communication performance compared to state-of-the-art designs. The used design flow exploits the benefits of IP cores for rapid development times while still offering the designer the full range of optimization possibilities for a specific design. It allowed us to do an almost complete design space exploration, assuring a near-optimal solution to the given problem. The implementation platform is a XILINX Virtex II Pro FPGA.

1 Introduction

It is well known that the increasing chip complexities demand a higher design productivity. An important method for decreasing development effort and shortening development time is the use of IP cores. Many common building blocks, such as multipliers, dividers and dual port rams, can be addressed by freely available IP cores. Configurability regarding precision, finite word lengths or throughput make them suitable for a wide range of designs, thus drastically reducing engineering effort for design and verification.

This paper demonstrates the design of a configurable IP core for blind carrier synchronization with novel features (combined frequency and phase offset estimation, an advanced modulation removal technique, and a configurable windowed spectral analysis). This building block is in use in a large scale communication system. The system throughput requirements can not be fulfilled by a DSP and due to the low volume an FPGA solution is far more cost efficient than an ASIC implementation. Thus, XILINX Virtex II Pro FPGAs were selected as implementation platform. We used the freely available IP cores generated with the XILINX core generator tool. The case study shows the high level design flow from the basic algorithms of the application to the implemented building block. This design flow includes the mapping of algorithmic steps to the available IP cores and the evaluation of algorithmic variations, considering both, communications performance and implementation complexity. While designing a communication system the designer always has to trade off between these two. Our building block offers a configurable interface to accept different word lengths at the input.

Every inner receiver for wireless communication must estimate the generally unknown parameters of timing, frequency and phase offset of the received signal. All possible negative influences introduced by them have to be eliminated. As mentioned, this paper concentrates on the frequency and phase offset estimation and correction (carrier 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.

The rest of the paper is structured as follows. In Sect. 2 an overview about the used design flow is given. Section 3 provides an overview of the problem of carrier synchronization and the basic algorithms for the synchronization task. The individual steps carried out in this case study, according to the design flow, are covered in the following sections. Section 4 shows possible optimizations of the basic algorithm as well as a way to combine the task of frequency and phase offset estimation. Section 5 deals with the examination of the implementation alternatives and the mapping of algorithmic tasks to XILINX IP cores. The resulting architecture is presented in Sect. 6, followed by the conclusions in Sect. 7.

Correspondence to: D. Schmidt (schmidt@eit.uni-kl.de)

Published by Copernicus GmbH on behalf of the URSI Landesausschuss in der Bundesrepublik Deutschland e.V.

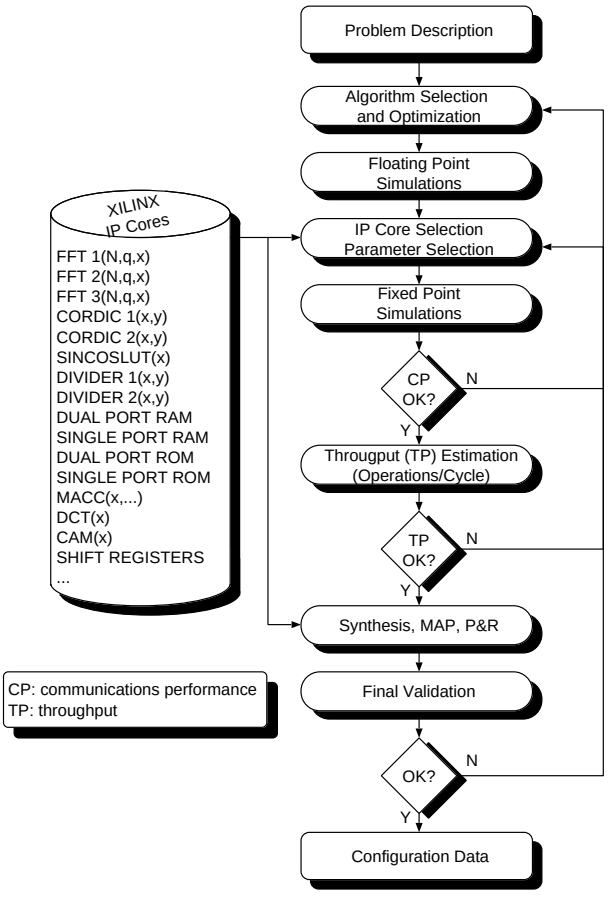

Fig. 1. The used design flow for IP core based design.

2 Design flow

Figure 1 shows the used design flow. Typically any given problem can be solved with various algorithms. Additionally, each of the algorithmic alternatives may depend on several parameters. There may also be different ways to implement an algorithm using different IP cores. Hence, all implementation alternatives form a tree, the so called design space. In the phase of design space exploration the designers task is to select the algorithm, the parameters and the IP cores to solve the problem under all given constraints. In a communication system these constraints usually regard the implementation complexity, the minimum throughput and maximum latency as well as the achievable communications performance. As the characteristics of each algorithmic alternative are not known in advance it is necessary to estimate them for each algorithm under consideration until a feasible solution is found. This is indicated by the iterative loops in the design flow diagram.

To minimize the time spent for design space exploration it is vital to exclude as many implementation alternatives as early in the process as possible. This is best done by evaluating the communications performance of each algorithm prior to the time consuming estimation of the implementation complexity and comparing it to the requirement specifications. A first estimate of the communications performance of an algorithm can be derived using a floating point model. The use of a high level modeling language, like SystemC or MATLAB, allows for faster development and shorter simulation times compared to a VHDL model. These first models are neither cycle accurate nor bit true models of the later implementation. Once an algorithm achieving a sufficient communications performance is found, the different implementation alternatives have to be reviewed. Thus a more accurate fixed point model of the hardware is needed that takes into account the effects of different quantization levels and of the IP cores used. Cores from the XILINX IP core library can be used to reduce development effort. To evaluate the impact of the decisions made on the communications performance, the simulation results of the fixed point model are compared to the results of the floating point simulation. The fixed point models can still be written in a high level language.

The core library provides IP cores for many high latency operations like FFTs or dividers. In cases where the number of clock cycles needed to perform a calculation is dominated by such complex operations, a rough throughput estimation can be done based on the specifications of the used IP cores. The glue logic does not significantly contribute to the computation complexity or the implementation complexity. Hence, to estimate the number of occupied slices, it is sufficient to instantiate and connect the main blocks of the design. Then, only for the design with the optimal trade-off a synthesizable VHDL-model has to be described and validated.

As a last step the resulting design has to be verified using the fixed point model. In the area of communication systems this can either be done with a bit true C-model or by comparison of statistical bit error rates. After synthesis, mapping and place and route have been performed, a final validation of throughput and area ensure the compliance with the given constraints.

3 Problem description

This section contains all fundamental algorithms needed for the development of our blind carrier synchronization building block. Further details and resources can be found in Meyr et al. (1997); Viterbi and Viterbi (1983); Mengali and D'Andrea (1997); van Trees (1968); Barry et al. (2004). The basic synchronization steps are estimation of the frequency offset, frequency offset correction, estimation of the phase offset and correction of the phase offset.

In an MPSK modulation scheme the set of M possible symbols s_m is given by:

$$s_m := e^{j\left(\frac{2\pi \cdot m}{M}\right)} \qquad m = 0, 1, .., M-1 \tag{1}$$

The received sample sequence r is given in the complex baseband according to Eq. (2):

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

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. The frequency offset f_o and phase offset Φ have to be estimated and corrected. They are considered fixed during one burst transmission.

The estimation of the unknown parameters f_o and Φ can be done by data aided or non data aided methods. Data aided methods need known symbols in the transmitted bursts. The introduced redundancy is exploited to estimate the frequency and phase offset, at the cost of decreasing bandwidth and power efficiency (Meyr et al., 1997). Hence, non data aided or blind methods become mandatory. They estimate the parameters without the help of such a training sequence. In this case especially estimation of frequency offset becomes a very critical task with high impact on communications performance. Deviation of the frequency estimation results in bursts with high bit error rates, resulting in high frame error rates after channel decoding.

For blind frequency or phase offset estimation the modulation introduced by the unknown transmitted symbol sequence s has to be removed from the received sample sequence r. State-of-the-art modulation removal is based on a power of M operation on r, where M is the modulation index.

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

Note that by such a power of M operation all the MPSK symbols defined in Eq. (1) are transformed to the symbol s_0. After the modulation has been removed, an approximation of the maximum likelihood estimation of the frequency offset can be found by calculating the spectral power of the sequence $\tilde{r}$. This is done by performing a Fast Fourier Transformation (FFT) (Mengali and D'Andrea, 1997) with N points according to Eq. (4):

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

The sequence x with N elements corresponds to the sequence $\tilde{r}$ with $L \leq N$ elements of Eq. (4), which is padded with zeros to achieve the N elements of Eq. (4). Note that each bin $X(k)$ represents the spectral power of x in a range of $(M \cdot N \cdot T)^{-1}$ Hz, not only in one distinct frequency. This results in a loss of accuracy, depending on the number N of points. The estimated frequency offset $\tilde{f_o}$ is given by spectral analysis of the FFT output, i.e. the FFT bin corresponding to the estimated frequency offset is given by:

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

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:

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

Phase offset estimation can be done using the V and V algorithm presented in Viterbi and Viterbi (1983). This algorithm requires the modulation to be removed from the sample sequence u, analogous to Eq. (4):

$$\tilde{u}(l) := u(l)^M \quad l = 0, 1, .., L-1 \quad (7)$$

The phase offset of the sample sequence u is then estimated to be the average phase of the symbols in the sequence $\tilde{u}$, as given in Eq. (8):

$$\tilde{\Phi} := M^{-1} \cdot \arg \sum_{l=0}^{L-1} \tilde{u}(l) \quad (8)$$

The sequence $u(l)$ must be corrected with the estimated phase offset $\tilde{\Phi}$, which results in

$$v(l) := u(l) \cdot e^{-j\tilde{\Phi}} \quad l = 0, 1, .., L-1 \quad (9)$$

Equation (8) still leaves an M-time phase ambiguity. Apart from this phase ambiguity the sequence $v(l)$ gives an estimation of the transmitted symbols $s(l)$. The resolution of the phase ambiguity can be achieved using several methods, e.g. differential encoding can be used. But this strongly depends on the actual communication system. Thus this step of the synchronization process is not considered in our case study.

4 Algorithm optimization

During the design space exploration we reviewed several modifications of the above mentioned algorithms. As mentioned, we were particularly interested in their impact on implementation complexity and communications performance.

Since the power operation for modulation removal in Eqs. (4) and (7) enhances the noise in the signal super proportional, so called self noise is inferred. This results in large frequency offset 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.

To elude this problem another approach for removing the modulation, proposed in Wang et al. (2003); Viterbi and Viterbi (1983), is used. This technique for modulation removal is based on the following calculation:

$$\tilde{r}(l) := |r(l)|^k \cdot e^{jM \cdot arg(r)}$$
$$k = 0, 1, .., M; \; l = 0, 1, .., L-1 \quad (10)$$

The parameter k is an optimization parameter. Observe that in case of $k = M$ Eq. (10) transforms into Eq. (4).

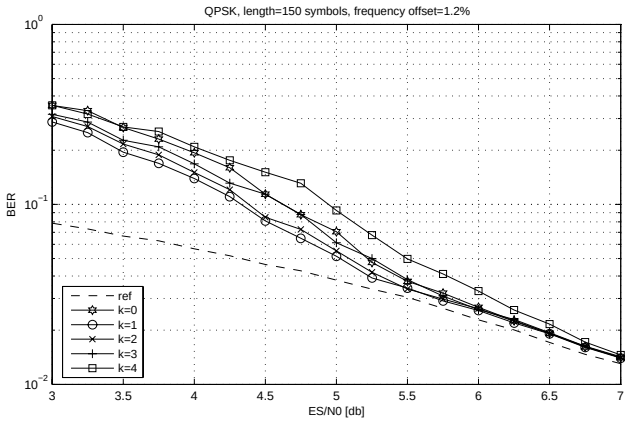

Fig. 2. BER of 150 symbols QPSK bursts for different values of k.

The calculation of the phase offset $\tilde{\Phi}$ can be simplified with further algebraic transformations. Combining Eqs. (6–8) gives an equivalent equation for calculating the estimated phase offset $\tilde{\Phi}$:

$$\tilde{\Phi} := M^{-1} \cdot \arg\left(X(k_f)\right) \tag{11}$$

Equation (11) uses the FFT result of Eq. (4) for the bin k with $k = k_f$. Thus frequency and phase offset estimation can be combined efficiently. Note that in this case phase offset estimation and frequency offset estimation implicitly use the same technique for modulation removal. The input burst has to be corrected with the estimated values of frequency and phase offset, formally described as:

$$v(l) := r(l) \cdot e^{-j\left(\frac{2\pi k_f \cdot n}{M \cdot N}\right)+\tilde{\Phi}} \qquad l = 0, 1, .., L-1 \tag{12}$$

If the range of the occurring frequency offset is known in advance, the communications performance can be improved significantly by simply limiting the spectral analysis. This windowing technique is given by substituting k_f in Eq. (12) with k_{fw} from Eq. (13):

$$k_{fw},\ s.t.:\ |X(k_{fw})| = \max_k |X(k)|$$
$$k = 0, .., w_u, w_l, .., N-1 \tag{13}$$

The parameters w_u and w_l define the FFT bin range, where the frequency offset has to be found. Setting w_u to $N/2-1$ and w_l to $N/2$ describes the FFT bin k_f given in Eq. (5).

Evaluating the impact of the parameter k on communications performance is done by simulation. Figure 2 shows the bit error rates (BER) of QPSK bursts with a length of 150 symbols for the different modulation removal techniques. We simulated transmission of the bursts over an additive white Gaussian noise (AWGN) channel using a floating point model written in C. 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. Clearly, $k = 1$ yields the best performance. Simulations with different burst lengths and BPSK bursts showed similar results. A detailed discussion of the presented algorithms regarding communications performance can be found in Brack et al. (2004).

The frequency and phase estimation can be performed in a combined manner defined by Eq. (11) or in a separate manner, defined by Eqs. (6–8). Obviously, the combined frequency and phase offset estimation reduces the overall computational complexity. Simulation results also showed that using different modulation removal techniques for frequency and phase offset estimation does not yield a better communications performance.

5 IP core selection

To obtain a first estimate of the number of slices occupied by each implementation alternative, a VHDL-model that merely instantiates and connects the needed IP cores and all major building blocks, is used. The time consuming task of describing all the needed control logic and validating the design is not yet necessary at this stage, because for computational intensive algorithms, like the ones under consideration, the control logic does not significantly contribute to the implementation complexity.

The components of the synchronization circuit are derived from the optimized algorithm and mapped to XILINX IP cores. We also used specific XILINX resources like the internal multipliers (MULT) and block RAM (BRAM) available on the Virtex II Pro FPGA.

For the FFT calculation defined in Eq. (4) three XILINX IP cores with different architectures are available (Xilinx, 2003a). The architectures differ in throughput, latency and needed resources. Due to the throughput requirements we had to choose the fastest FFT core with a throughput of one FFT bin calculation per clock cycle independent of the parameter k. The number of FFT points must be chosen to be at least twice the maximum burst length in the application Mengali and D'Andrea (1997). It also has to be considered that more points result in a higher resolution of the FFT bins and hence in a higher estimation accuracy.

For the modulation removal defined in Eq. (10) the mapping to IP cores depends on the parameter k. A mapping from the equation to the implementation requires polar coordinates. Since, however, the input samples r are given in Cartesian coordinates a transformation is necessary. The calculation of $\arg(r)$ and $|r|$ can be done with the CORDIC algorithm (Volder, 1959), which works iteratively. XILINX provides a lot of different variations of a CORDIC IP core (Xilinx, 2003b). A fully pipelined version with a throughput of one sample per clock cycle and a serial version with a throughput inverse proportional to the number of iterations are available. To fulfill high throughput requirements the pipelined IP core was chosen.

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

Alt.	Equation	Cores	MULT	SLICES
0	$\alpha := e^{j4arg(r)}$	aCORDIC, SCL	0	273
1	$\lvert r\rvert \cdot \alpha$	CORDIC, SCL	2	331
2a	$r^4/(\Re(r)^2+\Im(r)^2)$	DIVIDER	10	297
2b	$\alpha(\Re(r)^2+\Im(r)^2)$	aCORDIC, SCL	4	290
2c	$\lvert r\rvert^2 \cdot \alpha$	CORDIC, SCL	3	325
3	$\lvert r\rvert^3 \cdot \alpha$	CORDIC, SCL	4	329
4	r^4		6	79

In some cases of k and M the use of a CORDIC module obsolete. For $k = M$ Eq. (10) reduces to a power of M operation, which can be implemented using 3 multipliers in the case of BPSK and 6 multipliers for QPSK modulation.

For $M = 4$ and $k = 2$ Eq. (10) equals $\tilde{r} = r^4/|r|^2$. To calculate $|r|^2 = \Re(r)^2 + \Im(r)^2$ two multipliers are needed, so the CORDIC module can be replaced by 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 $|r|$. Then, again, we need two additional multipliers to calculate $|r|^2$. The simplified CORDIC module is also sufficient to perform the calculation of Eq. (10) in the case of $k = 0$.

After modulation removal the resulting sample sequence $\tilde{r}$ has to be transformed back to Cartesian coordinates for the FFT calculation (see Eq. (4)). This is accomplished with a sine-/cosine-look-up-table (SCL) (Xilinx, 2002), also available as an IP core, and internal multipliers to realize the e^{jx} operation.

The realization of Eq. (5) as well as the improved version, defined in Eq. (13), is not based on IP cores. The windowing feature proposed in Eq. (13) to evaluate k_{fw} is realized using a limited maximum absolute value search on the determined FFT bins. This can be implemented in a very efficient way using only counters and comparators. Since this part does not differ between two algorithmic alternatives, it does not have to be taken into account when comparing their implementation complexities.

After determining the frequency bin k_{fw}, the phase offset is estimated according to Eq. (11). 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 aCORDIC core without the ability to calculate $|r|$.

Rotation of the samples, as described, in Eq. (12) is accomplished with complex number multiplications and a second SCL.

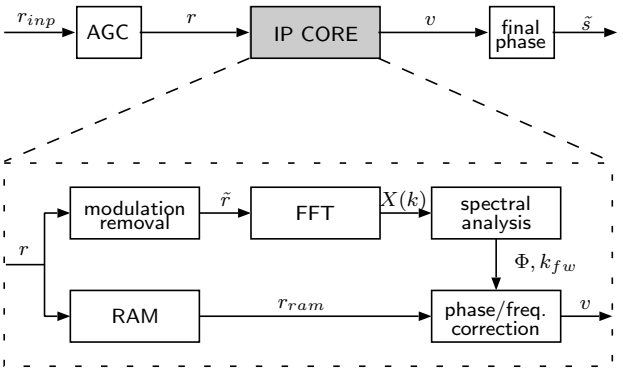

Fig. 3. Base architecture for blind carrier synchronization.

The considered implementation alternatives for modulation removal and the IP cores needed by each of them are listed in Table 1. It also shows the number of slices and other hardware resources occupied. The resources needed by the FFT and for the other building blocks (phase offset estimation, spectral analysis with windowing and combined correction) are not taken into account, because they are the same for all values of k.

6 Results

We used XILINX ISE 6.1 for synthesis and place and route with XILINX Virtex II Pro FPGA XC2VP50 as target platform. Figure 3 shows the main building blocks of the implemented architecture. We assume that the input signal is filtered using a matched filter and is sampled at one sample per symbol. As a front-end to the IP core an automatic gain control unit (AGC) is required to adapt the input samples bit width to the IP core input bit width without limiting the dynamic range. The back-end must deal with the M-times phase uncertainty introduced by Eq. (8).

The building blocks modulation removal, FFT, spectral analysis, and phase/freq. correction correspond directly to the algorithms from Sect. 3 and Sect. 4. The RAM buffer is used to guarantee constant throughput by hiding the FFT latency.

The final working and validated IP core is configurable regarding different bit widths (BW) of the input signal r and maximum burst lengths (MBL). As can be seen in Table 2 the size of the IP core grows strongly with increasing MBL due to the FFT component. The results also show that the overhead introduced by the alternative modulation removal with $k = 1$ (about 250 slices, Table 1) is very small compared to the overall resource needs of the IP. Considering the 0.5 dB gain in communications performance over the state-of-the-art technique with $k = 4$ (Fig. 2), we considered implementation of the alternative modulation removal to be worthwhile.

Table 2. Synthesis results.

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

The resulting maximum sample throughput is about half the clock speed because the FFT point size is at least twice the maximum burst length (Sect. 5). Thus the corresponding bit rate is determined by the modulation:

$$\text{throughput}[{}^{\text{bits}}/_{\text{sec}}] \approx (2 \cdot \text{cycle})^{-1} \cdot \log_2(M) \quad (14)$$

For example, using QPSK modulation Eq. (14) evaluates to a bit rate beyond 100 Mbps for every configuration (Table 2).

We have also shown that an IP core based design flow allows not only for a rapid implementation but also for a fast and exhaustive design space exploration. As in many designs resource needs and throughput are dominated by operations that can be covered with pre-existing IP cores, it is possible to quickly and accurately estimate the performance of each implementation alternative based on high level simulations (in SystemC/MATLAB) and the specifications of the used IP cores. The time demanding step of describing and verifying a fully functional and synthesizable VHDL-model has only to be done for the final architecture. Thus, the IP core based design flow proved to combine near optimal results with very short development times.

7 Conclusions

In this paper we presented a configurable building block that utilizes advanced techniques like combined frequency and phase synchronization, improved modulation removal and a windowed spectral analysis. These features yield high throughput, compact size and best communications performance achievable for transmissions without training sequence. We used a design flow suitable for a wide range of designs. In our case study we showed how to employ it to efficiently explore the design space for a blind frequency and phase synchronization module for MPSK bursts.

References

Barry, J. R., Lee, E. A., and Messerschmitt, D. G.: Digital Communication, Kluwer Acad. Publishers, Boston/Dordrecht/London, 2004.

Brack, T., Wasenmüller, U., Schmidt, D., and Wehn, N.: Design Space Exploration for Frequency Synchronisation of BPSK/QPSK Bursts, in Kleinheubacher Berichte, 48, 337–341, 2004.

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

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

van Trees, H. L.: Detection, Estimation and Modulation Theory, Wiley and 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 Trans. on Inf. Theory, 32, 543–551, 1983.

Volder, J.: The CORDIC Trigonometric Computing Technique, IRE Transactions on Electronic Comp., 8, 330–334, 1959.

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

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

Xilinx Inc. CORDIC v2.0, Mar. 2003b.

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