International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 3, No. 1, March 2014, pp. 18~24
ISSN: 2089-4864

Realization of Programmable BPSK Demodulator-Bit Synchronizer using Multirate Processing

Anshuman Sharma, Abdul Hafeez Syed, Midhun M, M R Raghavendra
Spacecraft Checkout Group, ISRO Satellite Centre, Bangalore, 560017, India

Article Info

Article history:

Received Jan 2, 2014
Revised Feb 24, 2014
Accepted Mar 1, 2014

Keyword:

BPSK
Costas loop
FPGA
Gardner Timing Detector
VHDL

ABSTRACT

This paper presents the design and implementation of programmable BPSK demodulator and bit synchronizer. The demodulator is based on the Costas loop design whereas the bit synchronizer is based on Gardner timing error detector. The advantage of this design is that it offers programmability using multi-rate processing and does not rely on computation of filter coefficients, NCO angle input for each specific data rate and thus avoids computational complexities. The algorithm and its application were verified on Matlab-Simulink and were implemented on ALTERA platform. A 32 kHz BPSK demodulator–bit synchronizer pair catering for a data rate from 1kbps to 8kbps was implemented.

Corresponding Author:

Anshuman Sharma,
Spacecraft Checkout Group,
ISRO Satellite Centre,
Bangalore, 560017, India
Email: anshuman@isac.gov.in

1. INTRODUCTION

Bipolar phase shift keying (BPSK) modulation is widely used modulation scheme in telemetry chain of satellites due to its power efiiciency. The modulation scheme employed for telemetry transmission is PCM/PSK/PM. Whereas the PM modulation is done at S-band/C-band frequencies, the PSK subcarriers are basically at 32kHz and 128kHz. With the ever increasing complex interplanetary mission being explored by ISRO the telemetry data rates have varied from 100bps to 8kbps on the subcarriers. This work discusses about an implementation scheme of programmable BPSK demodulator-bit synchronizer pair in digital domain which can be dynamically configured for variable data rates.

The BPSK demodulator is Costas loop based design. The block diagram of Costas loop is given in Figure 1. There are existing solutions for implementation of BPSK modem [1]-[4]. The Costas loop extracts the demodulated signal at the "in-phase" branch of the loop. The Numerically controlled Oscillator (NCO) which can be implemented using the **Co**-ordinate **R**otation **Di**gital **C**omputer (CORDIC) algorithm)[5] locks on to incoming signal and simultaneously generates the demodulated output. The output of the demodulator is given to bit synchronizer for clock recovery, data extraction. The bit synchronizer can be implemented using the Gardner timing-error detector [6]. To introduce programmability in this design the arm filters, loop filters and NCOs have to be tuned as per the data rates. The tunable filters have been discussed in [7]. All this retuning calls for a computer interface where new filter coefficients, NCO angle input can be recomputed and passed on to the FPGA design.

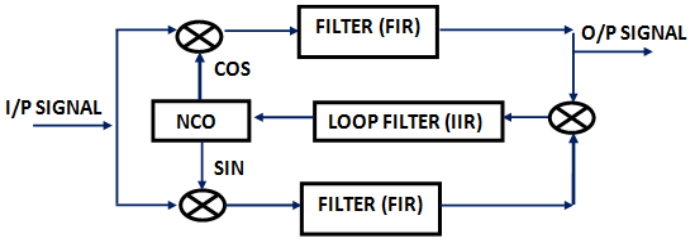

Figure 1. Basic Costas Loop

In this work, we use multirate sampling [8] to configure the demodulator-bit synchronizer pair for any data rate from 1kbps to 8kbps. This approach is based on the concept of achieving reconfigurability by varying the sampling frequency rather than recomputing coefficients for reconfiguring.

In the next section we discuss the basic concept and design of the demodulator-bit synchronizer pair. Subsequently we present the implementation of the design on hardware.

2. RESEARCH METHOD

2.1. The Concept

As explained earlier the focus of this work is to achieve programmability without repeatitive computational burdens while designing a programmable BPSK demodulator-bit synchronizer. The whole concept is based on the fact that in digital signal processing (DSP) all the computations are based on sampling frequency. A filter working at sampling frequency f1 with passband frequency of f_{pass} and stop band frequency of f_{stop} can be made to work as filter with a different passband and stopband frequency just by changing the sampling frequency to f2, since both f_{pass} and f_{stop} are normalized wrt sampling frequency.

As an example refer to the Table 1, a 16 tap FIR filter is designed using Kaiser window for 6-dB pass band frequency of 10kHz at a sampling rate of 100kHz. The ratio of pass band to sampling frequency is 0.1. Next the same coefficients are sampled at a rate of 80kHz and they provide 6-dB pass band frequency of 8kHz again the same ratio of 0.1 is maintained. This concept forms the basis of this work and it has been further discussed.

Table 1. Filter Characteristics WRT Sampling Frequency

Filter Coefficients (Kaiser Window , 16 tap filter)	Sampling Frequency	6-db pass band frequency	Ratio of pass band/sampling frequency
-0.0023, -0.0060, -0.0052, 0.0100, 0.0467, 0.1018, 0.1591, 0.1959, 0.1959, 0.1591, 0.1018,, 0.0467, 0.0100, -0.0052, -0.0060, -0.0023	100 kHz	10kHz	0.1
-0.0023, -0.0060, -0.0052, 0.0100, 0.0467, 0.1018, 0.1591, 0.1959, 0.1959, 0.1591, 0.1018,, 0.0467, 0.0100, -0.0052, -0.0060, -0.0023	80kHz	8kHz	0.1

Similarly the NCO can be modified to give output at different frequency by simple variation of sampling frequency. Thus, changing the sampling frequency avoids re-computation of filter coefficients and NCO angle input for varying data rates.

Figure 2 depicts general phase lock loop (PLL) architecture in digital domain. In the PLL if the loop filter characteristics are to be changed it can be done by two ways- either by recomputation of filter coefficients or as stated above, by changing the sampling frequency. But changing the sampling frequency will change the free running frequency of the NCO and changing coefficients require re-computation. So a new architecture for PLL was thought as depicted in Figure 3. The first NCO-multiplier pair downconverts the input signal to close to zero frequency and the PLL thereafter corrects for the phase and frequency errors. The loop filter characteristics can be easily modified by changing the sampling frequency. This change in

sampling frequency does not change the output frequency of the second NCO as it is configured to work at zero frequency. This work demonstrates this concept on FPGA hardware which is discussed in subsequent section.

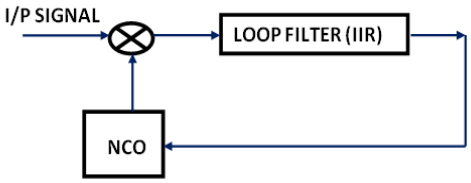

Figure 2. Phase lock loop

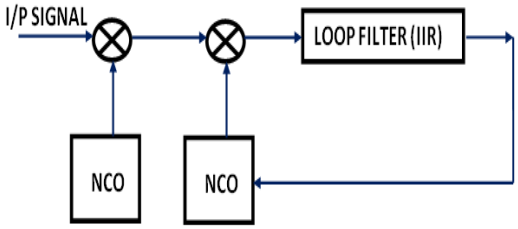

Figure 3. Modified Phase lock loop

2.2. Implementation

2.2.1. Clock Generation

To formulate the concept we did the functional simulation on MATLAB-Simulink and implemented the programmable BPSK demodulator-bit synchronizer pair on an Altera FPGA. The basic block diagram is presented in Figures 5, 8 and 9. The design is based on PLL.

Refer to Table 2; the input sampling frequency is 224kHz. For the data rates from 8kbps to 1kbps the sampling frequency is scaled down from 112kHz to 14kHz. The corresponding interpolation and decimation values are indicated. All along, the ratio of sampling frequency to data rate is maintained as 14 to provide sufficient samples for bit synchronizer to lock. Even though bit synchronizer is based on Gardner timing detector which requires only two samples per bit we are taking 14 samples per bit as we are not adjusting our sampling instants but rather selecting the sample pair which gives the best Signal to Noise ratio. So, sufficient number of samples is required for getting the best strobing instant. For 8kbps data rate the sampling rate after interpolation is 1.792MHz. The maximum clock frequency corresponding to this sampling frequency is 32.256MHz. This clock frequency is required for the anti-aliasing filter in between the interpolator and the decimator, refer to Figure 5. The filter is a 64 tap, Kaiser window FIR filter. It has been implemented with certain amount of parallelism so clock frequency of 16 times as compared to the sampling frequency is required. We have kept two clock cycles as buffer, (32.256/1.792=18). This clock is generated by an onboard Crystal oscillator. All the clocks required for programmability are generated by onboard PLL on the FPGA, routed to multiplexer to which selection is provided through input/output (I/O) lines of the FPGA, as indicated in Figure 4.

Table 2. Data Rates and the Clock Requirement

Sampling Frequency-fs (kHz)	Data Rate(kbps)	Required Sampling frequency-fs1 (kHz)	Ratio fs1/ fs	Interpolation value(M) ↑	Required Filter gain	Sampling frequency after Interpolation (MHz)	Decimation value(N) ↓	Required clock frequency for 64 tap anti-aliasing filter (Figure 5) (MHz)
224	8	112	(1/2)=> 8/16	8	8	1.792	16	32.256
	7	98	7/16	7	7	1.568	16	28.224
	6	84	6/16	6	6	1.344	16	24.192
	5	70	5/16	5	5	1.120	16	20.160
	4	56	4/16	4	4	0.896	16	16.128
	3	42	3/16	3	3	0.672	16	12.096
	2	28	2/16	2	2	0.448	16	8.064
	1	14	1/16	1	1	0.224	16	4.032

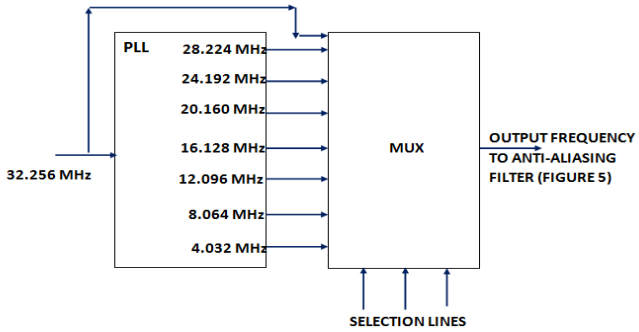

Figure 4. Clock generation on FPGA

2.2.2 Building Blocks

Refer to the Figure 5, the incoming 32kHz BPSK modulated signal in the form of $ICos(x)$ (I is the modulating NRZ-M bipolar data) is passed through an Anti-aliasing filter (AAF) which is a 6th order Butterworth filter providing necessary rejection at (fs) sampling frequency/2. The analog signal is converted to digital signal with sampling frequency (fs) of 224kHz using a 10-bit pipelined (ADC) analog to digital converter. This signal is then brought down to close to "**0**" intermediate frequency (IF) in the form of $ICos(\Delta x)$ and $ISin(\Delta x)$ (Δx is the instantaneous phase and frequency error between the between the incoming signal and the NCO) by the front end IQ detector implemented in digital domain using NCO (running at sampling frequency of 224kHz) and multipliers. The whole idea of segregating the Costas loop has been explained earlier.

The derived I and Q outputs are then passed through a bank of interpolator, filter and decimator which allow sampling rate conversion by non-integer factor. The anti-aliasing FIR filter present in between the interpolator and the decimator is as explained earlier a Kaiser window, 64 tap filter designed to provide sufficient attenuation at stop band frequency of fs/2. Figures 6, 7 show the amplitude response of the filter simulated in MATLAB for sampling frequencies of 112, 98kHz respectively. It uses same filter coefficients but gives the desired performance with changing sampling frequencies.

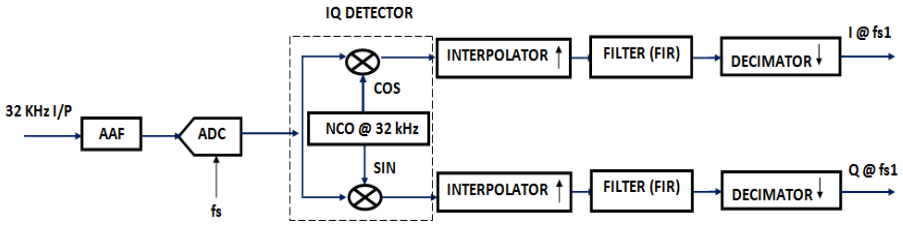

Figure 5. Demodulator front end

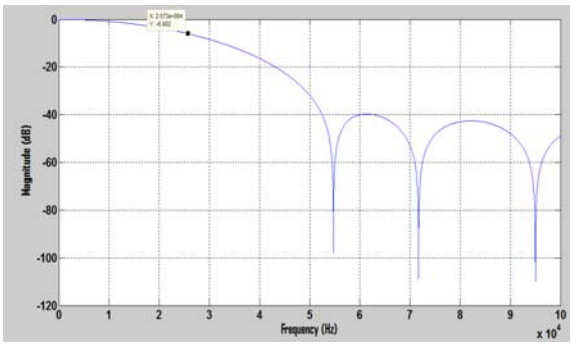

Figure 6. Kaiser filter reponse for sampling frequency of 112kHz

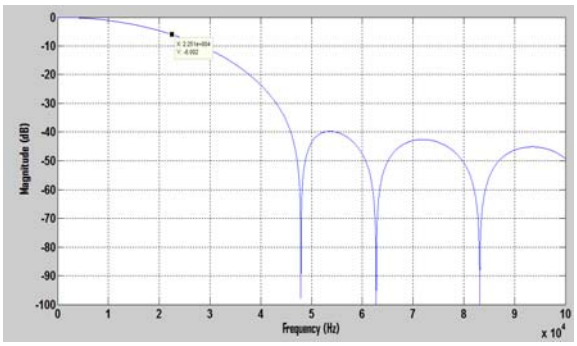

Figure 7. Kaiser filter reponse for sampling frequency of 98kHz

Refer to the Figure 8, the I and Q outputs after the decimator are given to the PLL block which corrects for any residual phase and frequency offset between the incoming signal and the front end NCO in the IQ detector. The NCO in this block is configured to work at “0” frequency. The in-phase arm of the PLL block gives the demodulated output. Tha arm filters are 16 tap, Kaiser window FIR filters which also get reconfigured with the change in sampling frequency.The output is routed to an 8 bit Digital to Analog Converter (DAC) for monitoring purpose and also parallely to the bit synchronizer for clock and data recovery.

Figure 9 shows the block diagram of the bit synchronizer. The bit synchronizer does the clock recovery and also recovers the data. Gardner algorithm is used for Timing Error Detection. This algorithm is suitable for both tracking and acquisition modes of operation. Also, the clock recovery does not depend on carrier phase. In this algorithm, only two samples of the signal are required for each data symbol. And also, one of the two samples is used for symbol strobing (i.e., the sample on which the symbol decision is made). The timing error detector operates upon samples and generates one error sample for each symbol. This error sequence is smoothened by a loop filter and then used to adjust a timing error corrector, which in this case is an NCO. The NCO is configured to run at a frequency which is double the discrete data rates as per the Table 2. With change in the sampling frequency, the NCO output frequency changes according to the sampling rate and the PLL locks for all the data rates between 1kbps to 8kbps as the loop filter is designed to achieve wide acquisition bandwidth [9]. For details on Gardner timing detector, refer to [6].

All the loop filters both in demodulator and bit synchronizer are first order lead-lag filters derived from their analog counter-part using bilinear transformation method.

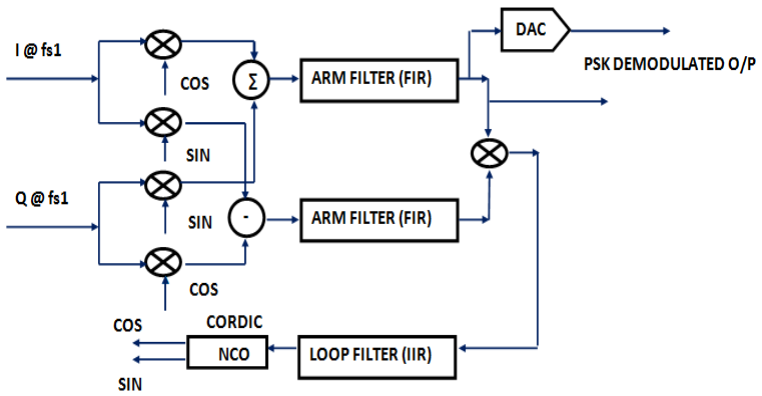

Figure 8. The PLL block

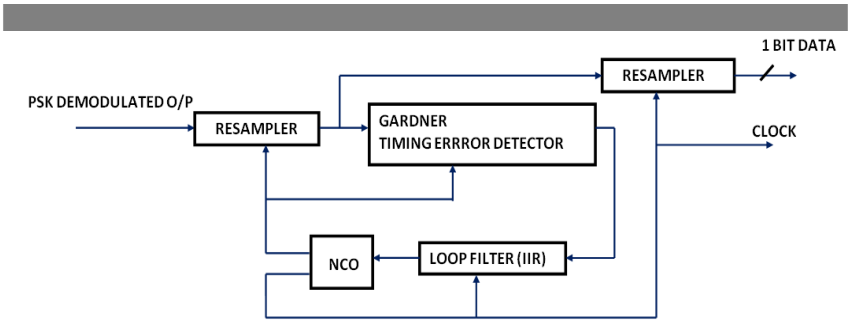

Figure 9. Bit synchronizer block diagram

3. RESULTS AND ANALYSIS

The demodulator-bit synchronizer was implemented on FPGA and the functionality was verified. Figure 10 shows the output of the BPSK demodulator for 8kbps data rate, routed through DAC. The output is seen without an anti-imaging filter, so the sampling steps of 112kHz are seen. To verify the entire functionality a simulated modulated data with known Frame synchronization code was given as an input to

the system and the bit sync data and clock output was given to an external frame synchronizer and continous frame sync lock was verified for all data rates from 1kbps to 8kbps.

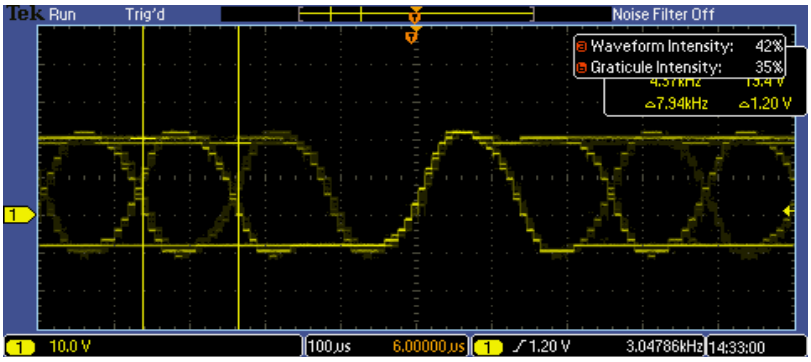

Figure 10. The BPSK demodulator- Output Eye diagram

3.1. Programmabilty for data rates below 1 kbps

Due to resource constraints on the FPGA, programmability for data rates from 8kbps upto 100bps could not be verified on hardware. But individually selection for 100 bps data rate using the same architecture was verified on the FPGA. The clock requirement for 100bps data rate is shown in Table 3. As we see that the ratio for interpolation to decimation is 1/160, additional decimation block of 160 was required for this data rate. Also, a different anti-aliasing filter had to be used with stop band frequency at (fs1)/(2) and pass band frequency at the data rate.

Table 3. Clock Requirement for 100bps Data Rate

Sampling Frequency-fs (kHz)	Data Rate(bps)	Required Sampling frequency-fs1 (kHz)	Ratio fs1/ fs	Interpolation value(M) ↑	Required Filter gain	Sampling frequency after Interpolation (kHz)	Decimation value(N) ↓	Required clock frequency (MHz)
224	100	1.4	1/160	1	1	224	160	4.032

4. CONCLUSION

A detailed design and development of programmable BPSK demodulator- bit synchronizer using multirate processing has been demonstrated as a part of this work. This work will be scaled for other carrier frequencies and other data rates in due course. The whole focus of this work is programmability but the higher clock requirement of 32.256MHz can still be further optimized to a lower clock if the FIR filter in the demodulator front end is implemented with further parallelism and also the bit synchronizer is made to work with exactly two samples per bit.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Mr. K.V.Govinda, Deputy Director ICA, Mr. K.B.A.R.Sarma, Group Director SCG, Mr. Raju Sagi, Sci-Eng "G" and Mrs U.Vasantha Kumari, Head SRCD/SCG, for all the encouragement and technical guidance during the course of design and development. We would like to specifically thank Mrs. Vasuki E. for precision fabrication expertise.

REFERENCES

[1] Pero Krivic, Goran Stimac. FPGA implementation of BPSK Modem for Telemetry Systems Operating in Noisy Environments. *MIPRO.* 2011.

[2] Yuan, *et al. Design and Implementation of Costas loop Based on FPGA.* 3rd IEEE International Conference on Industrial Electronics and Applications (ICIEA). 2008.
[3] Roddewig, *et al. Design of Costas Loop Down Converter.* 52nd IEEE International Midwest Symposium on Circuits and Systems(MWSCAS). 2009.
[4] Marvin K Simon, William C Lindsey. Optimum performance of suppressed carrier receivers with Costas loop tracking. *IEEE Transactions on Communications*. 1977; COM-25(2): 215-227.
[5] Ray Andraka. A survey of CORDIC algorithms for FPGA based computers. Andraka Consulting Group, Inc
[6] Floyd M Gardner. A BPSK/QPSK Timing-Error Detector for Sampled Receivers. *IEEE Transactions on Communications*. 1986; COM-34(5).
[7] Petri Jarske, *et al*. A Simple approach to the design of linear phase FIR digital filters with variable chracteristics. *Elsevier Science Publishers B. V. (North Holland), Signal Processing.* 1988; 14: 313-326.
[8] Jose Franco, *et al. Mutirate Analog-Digital systems for Signal processing and Conversion.* Proceedings of the IEEE. 1997; 85(2).
[9] MR Raghavendra, *et al*. Design and development of High Bit Rate QPSK Demodulator. *IEEE International Conference on Electronics, Computing and Communication Technologies (IEEE CONECCT)*. 2013.

BIOGRAPHIES OF AUTHORS

Anshuman Sharma is a Scientist at ISRO Satellite Centre. He joined ISRO in the year 2005. He did his Masters from Indian Institute of Technology-Madras in Communication Sytems. His research interest includes Digital signal processing, RF communication.

Abdul Hafeez Syed is a Scientist at ISRO Satellite Centre. He joined ISRO in the year 2011. He did his Masters from National Institute of Technology-Calicut in the year 2007 in Digital systems and Communication. His research interest includes Communication system. Digital signal processing, Design and Development of demodulators and bit synchronizers.

Midhun M is a Scientist at ISRO Satellite Centre. He joined ISRO in the year 2008. His research interest includes Communication system, Digital signal processing and baseband data coding-decoding.

M R Raghavendra is a Scientist at ISRO Satellite Centre. He joined ISRO in the year 1992. His research interest includes RF Communication systems, Digital signal processing, Design and development of analog and digital demodulators. Presently, he is the head of Checkout Baseband and RF Division (CBRD) at ISRO Satellite Centre.