BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIO ELECTRONICS

ÚSTAV RADIOELEKTRONIKY

X-BAND EARTH OBSERVATION SATELLITE SOFTWARE DEFINED RECEIVER

SOFTWAROVÝ PŘIJÍMAČ PRO DÁLKOVÝ PRŮZKUM ZEMĚ V PÁSMU X

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR
AUTOR PRÁCE

Bc. Radim Zedka

SUPERVISOR
VEDOUCÍ PRÁCE

Ing. Filip Záplata, Ph.D.

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**
Ústav radioelektroniky

Student: Bc. Radim Zedka ***ID:*** 164441
Ročník: 2 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Softwarový přijímač pro dálkový průzkum Země v pásmu X

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte dostupné družice pro dálkový průzkum Země a jejich signály pracující v pásmu X. Specifikujte požadavky na příjem volně poskytovaných dat přímo z vybraných družic, na jejichž základě vytvořte koncept přijímače. Pořiďte záznamy signálů z jednotlivých družic pomocí dostupného vybavení a sestavte a odlaďte algoritmy přijímače v MATLABu.

Odladěné algoritmy implementujte do dostupného HW s FPGA tak, aby vznikl přijímač schopný kontinuálního příjmu v době přeletu družice. Data ukládejte na paměťové médium dostupné pro pozdější zpracování a zobrazení jiným softwarem. Přijímač vhodně testujte v reálném provozu. Funkci přijímače důkladně zdokumentujte a zahrňte i analýzu náročnosti jednotlivých částí.

DOPORUČENÁ LITERATURA:

[1] MARAL, Gérard a Michel BOUSQUET. Satellite communications systems: systems, techniques and technology. 4th ed. Chichester: Wiley, c2002. ISBN 0-471-49654-5.

[2] SKLAR, Bernard. Digital communications: fundamentals and applications. 2nd ed. Upper Saddle River: Prentice Hall PTR, 2001, xxiv, 1079 pages. ISBN 0-13-084788-7.

[3] MENGALI, Umberto. a Aldo N. D'ANDREA. Synchronization techniques for digital receivers. New York: Plenum Press, c1997. ISBN 03-064-5725-3.

Termín zadání: 4.2.2019 ***Termín odevzdání:*** 16.5.2019

Vedoucí práce: Ing. Filip Záplata, Ph.D.
Konzultant:

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

ABSTRAKT

Práce se zabývá procesem návrhu digitálního přijímače pro signály družic dálkového průzkumu Země v pásmu X. V první části je uveden seznam družic které umožnují příjem vědeckých dat pomocí amatérských neautorizovaných stanic. Jsou zde vypsány základní signálové parametry některých družic, z nichž EOS-PM-1 je zvolena jako hlavní objekt pro návrh přijímače. Ve druhé části je použit software MATLAB pro simulaci družicového O-QPSK signálu v základním pásmu, mechanizmu kompenzace kmitočtového ofsetu, synchronizaci nosné, symbolové synchronizaci, rámcové synchronizaci a odstranění fázové dvojznačnosti vlivem modulace. Třetí část práce pojednává detailněji o implementaci jednotlivých bloků přijímače do FPGA při použití aritmetky s pevnou řádovou čárkou. Je zde popsána metoda pro verifikaci celého designu v reálném čase a závěrem je zde uvedeno porovnání výsledků měření touto metodou a výsledků simulace.

KLÍČOVÁ SLOVA

Softwarové radio, družice, pásmo X, Dopplerův posuv, synchronizace fáze nosné.

ABSTRACT

This thesis describes the process of designing the digital baseband receiver for the O-QPSK modulated signals from the X-band Earth observing satellites. The satellites provide free direct data broadcast for the civilian ground stations. The first part lists the main downlink parameters of several EOS satellites, choosing EOS-PM-1 as the main target. The second part uses MATLAB software to simulate the O-QPSK satellite signal in baseband, offset frequency compensation, carrier phase synchronization, symbol timing synchronization, frame synchronization and the carrier phase ambiguity removal. The third part elaborates previously simulated design blocks with respect to the FPGA implementation using fixed-point arithmetic. A sophisticated method of verifying the whole design in real time is introduced and performance comparison between simulations and measurements is done.

KEYWORDS

Software defined radio, satellite, X-band, Doppler shift, carrier phase synchronization.

ZEDKA, R. *X-band earth observation satellite software defined receiver*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2019. 54 s., 3 s. příloh. Vedoucí semestrální práce: Ing. Filip Záplata, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Softwarový přijímač pro dálkový průzkum Země v pásmu X jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Filipu Záplatovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne ....

(podpis autora)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Remote sensing information is essential for many countries around the world and some of them require everyday real-time local weather data to handle critical situations. Such places cannot afford to wait for the regular data updates provided by the international meteorological institutions and are forced to purchase direct broadcast receiver stations accessing the local weather satellite broadcast. However, commercially available stations can be quite expensive getting up to tens of thousands of euros. It is a goal of this thesis to design a digital baseband receiver capable of real time satellite data reception.

The whole project consists of:

- X-band RX antenna – X-band signal reception.
- Low noise block (LNB) – down-conversion into the intermediate frequency range.
- Intermediate frequency receiver - provides two-channel signal in baseband.
- Software defined radio (SDR) – AD conversion of the baseband signal.
- Digital baseband receiver – frequency offset removal, carrier phase synchronization, symbol timing synchronization, PRBS and Reed Solomon decoding.

The thesis focuses on the digital baseband receiver design, simulations in MATLAB software and FPGA implementation.

1 X-BAND EARTH OBSERVATION SATELLITES

This chapter describes some of the X-band satellites which provide the broadcast data accessible to the civilian ground stations.

1.1 EUMETSAT Polar System (MetOp)

European Satellite project MetOp (Meteorological Operational) consists of three satellites: MetOp-A (launched in 2006), MetOp-B (2012), MetOp-C (2018). These are located on the LEO polar orbit in the mean altitude 817 km [1] and their mission is to gather atmospheric measurement data and send it back to both main ground stations and local civilian ground stations.

MetOp uses L-band for transmitting AHRPT (Advanced High Rate Picture Transmission), VHF-band for LRPT (Low Rate Picture Transmission) and X-band for Global data dump (multiplexed data from most of the measuring instruments). More details are in table 1-1.

Table 1-1: MetOp data broadcast services.

Service	Frequency band	Data rate
AHRPT	L-band (1701.3 MHz)	3.5 Mbit/s
LRPT	VHF-band (137.1 MHz)	72 kbit/s
Global data dump	X-band (7750-7900 MHz)	70Mbit/s

1.2 EOS-AM-1 (Terra)

Terra is the first satellite of the Earth Observing System. Its main mission is to gather information about Earth's land, oceans, ice, air and observe the overall behavior of the ecosystem. It has been providing direct broadcast data since year 2000, located on sun-synchronous LEO polar orbit with altitude 705 km and total period 98.88 minutes [3]. It crosses equator in northwards direction at 10:30 am (hence the name EOS-AM-1) and together with EOS-PM-1 (Aqua) they provide morning/afternoon data for the same area.

It carries 5 main instruments: ASTER (land composition), MODIS(vegetation, snow and ice), CERES (reflected energy), MISR (aerosols), and MOPITT (carbon monoxide) [10]. All the measurements are multiplexed into a stream of data which is sent to the EPGS (EOS Polar Ground Stations) once the spacecraft is at their vicinity (in Direct Playback mode - DP)[3].

There are four modes of downlink communication Terra operates in:

- DB (Direct Broadcast mode) – active for most of the time. Provides MODIS data only.
- DB/DP (Mixed DB and Direct Playback mode) – DB/DB data is in the I channel and all-instrument data multiplex in the Q channel. I/Q symbol rates are not balanced.
- DP (Direct Playback mode) – No DB data is provided. All-instrument data is split into I/Q equally.
- DB/DDL (Mixed DB and Direct Downlink mode) – DB data is in the I channel and ASTER instrument data in the Q channel. I/Q symbol rates not balanced.

Together with EOS-PM1 (Aqua) its direct broadcast (DB) service is invaluable in disaster monitoring for providing real-time local weather information (MODIS only) to various places around the globe. Apart from the DB, all the instrument measurements are stored to the on-board solid state recorder and sent regularly to four EOS ground-stations which later distribute the data all over the world (with appropriate processing delay).

More EOS-AM-1downlink parameters are shown in Table 1-2 [2][3]:

Table 1-2: EOS-AM-1 downlink parameters.

Parameter	Value	Note
Data rate (avg)	6.216 Mb/s	Before RS & Conv. coding
Data rate (peak)	11.016 Mb/s	Before RS & Conv. coding
Carrier freq.	8212.5 MHz	
EIRP	14 W	
Bandwidth	26 MHz	
Modulation	SQPSK (UQPSK)	
I/Q power ratio	1/4	In DB mode
RS encoding	RS(255,223)	Interleave depth = 4
Conv. encoding ratio	1:1,	DP mode, DDL mode
CADU length	8192 Bits	
Polarization	RHCP	
Antenna coverage (nadir)	± 64 °	

In the DB mode (active for most of the time) Terra transmits data from the MODIS instrument using CCSDS (Consultative Committee for Space Data Systems) data format. The CADU (Channel Access Data Unit) packets are Reed Solomon coded and split into

I/Q data streams were (in DB mode only) the Q contains 100% of useful data. Both I and Q are then NRZ-M[1] encoded. Next, the convolution coding is applied to the Q part and SQPSK modulation then follows where the I/Q power ratio is ¼ (see Figure 1-1 [2]). The X-band up-converter and power amplifier are at the end of the signal processing chain.

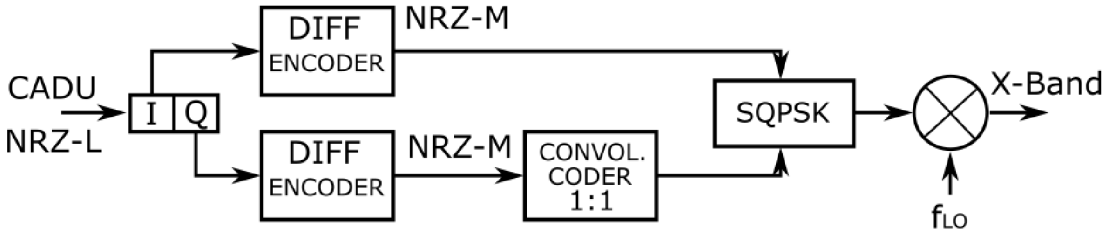


Figure 1-1: Terra DB mode downlink schematic.

1.3 EOS-PM-1 (Aqua)

The second satellite of the EOS program is located on the same orbit as the EOS-AM-1. It flies over the equator in northward direction at 1:30 pm and it is a part of a satellite group called the „Afternoon constelation“ or the „A-Train“ [11].

The DB mode is operational for most of the 99-minute orbit time except for 8 to 22 minute long DP or DB/DDL modes at the vicinity of the EPGS [3].

Aqua carries six earth-observing instruments:

- AIRS (Atmospheric Infrared Sounder)
- AMSR-E (Advanced Microwave Scanning Radiometer for EOS)
- AMSU (Advanced Microwave Sounding Unit)
- CERES (Cloud and the Earth’s Radiant Energy System)
- HSB (Humidity Sounder for Brazil)
- MODIS (Moderate Resolution Imaging Spectroradiometer)

From table 1-3 [3] it can be seen that the downlink signal bandwidth between the first two minimums equals two times the symbol rate. Since the documents [3][4][5] lack any information about the pulse-shaping filter used in Aqua, a simple rectangular pulse of a pulse-width equal to the symbol period was chosen for MATLAB simulation purposes. This also explains the channel bandwidth being equal to 150 MHz (considering 10-times the width of the main spectral lobe).

[1] NRZ-M is a differentially encoded binary signal mapped to ±1 range.

Table 1-3: EOS-PM-1 downlink parameters.

Parameter	Value	Note
Data rate	15 Mb/s ± 1.8 kb/s	After RS coding
Carrier freq.	8160 MHz	
Freq. stability (5 hours avg.)	$\pm 0.1 \times 10^{-6}$ ppm	
EIRP	25 W	
Bandwidth (1^{st} to 1^{st} null)	15 MHz	
Channel bandwidth	150 MHz	Centered at 8160 MHz
Modulation	SQPSK	
I/Q power ratio	1/1	
Link encoding	NRZ-M	In both I and Q channel
PRBS encoding	255-bit sequence	Excluding the start of the CADU frame
RS encoding	RS(255,223)	Interleave depth = 4
CADU length	1024 Bytes	
Polarization	RHCP	
Antenna coverage (nadir)	± 63,8 °	

2 DIGITAL BASEBAND RECEIVER SIMULATIONS

In this chapter, EOS-PM-1 (Aqua) satellite signal receiver is designed and its function is simulated in the MATLAB software. The goal is to simulate the function of the carrier-phase synchronizer, symbol timing synchronizer, ASM detector and the frame formatting circuit without considering most of the implementation issues related to quantization.

2.1 Downlink signal modelling

Before being up-converted and transmitted towards the GS, the data flow from the SC on-board instruments is multiplexed into CADU data packets (see figure 2-1) [5].

Each CADU contains a 4-byte synchronization word (Attached Sync Marker – ASM) which is used to resolve the phase ambiguity on the receiver side (together with the NRZ-M encoding). The ASM equals 0x1ACFFC1D in hexadecimal format [5].

Except for the SYNC word the CADU packet is PRBS coded using a polynomial

$$x^8 + x^7 + x^5 + x^3 + 1$$

where all registers are initialized to "ones" with every ASM detection moment [4]. The data block and both headers are Reed-Solomon coded using the RS(255, 223) coding scheme [4] with interleave factor 4. This can be understood as each RS coding symbol is 8 bits wide and for every 223-byte data-block there is a 32-byte RS parity block generated. So the data block (together with the headers) in the CADU is divided into four 223-byte blocks, each of them having its own 32-byte parity check block (see Figure 2-2).

In the DB mode the CADU packets are generated in a stream with fixed bitrate of 15 Mb/s ±1.8 kb/s.

As it was shown in table 1-3, the downlink signal is modulated by SQPSK (Staggered Quadrature Phase Shift Keying, or equivalently O-QPSK for Offset-QPSK) modulation.

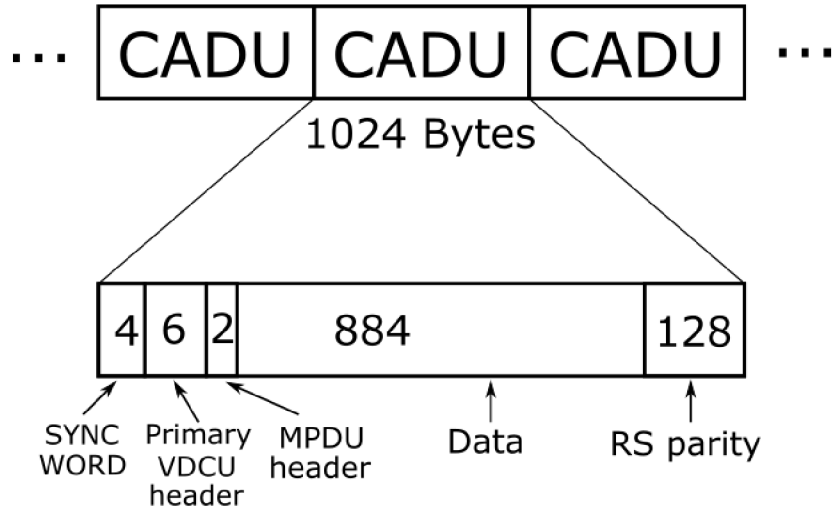


Figure 2-1: CADU frame structure.

This modulation scheme applies a delay of half the symbol period (one bit period) into the Quadrature channel which ensures limitation of phase change to $\pm\pi/2$ radians (see figure 2-3, right). Such a modification allows designers to use nonlinear amplifiers of higher efficiency which in case of QPSK would generate high frequency products [8].

The signal used for the downlink is up-converted to the carrier frequency in X-band (8160 MHz). Such a signal has the amplitude frequency spectrum similar to the one in figure 2-4 (left). Due to the finite frequency stability (see table 1-3) of the local oscillator (LO) in both SC and GS (more significant) and also due to the Doppler effect, the frequency offset after the down-conversion (figure 2-4, right) cannot be overlooked.

Figure 2-5 shows the magnitude spectrum of the downlink signal in baseband with frequency offset 180 kHz sampled at 50 MSPS (MATLAB). Before it is down-sampled to 16.666 MSPS (the final sampling rate chosen for further processing in the FPGA, see figure 2-6) the signal bandwidth must be narrowed to prevent aliasing caused by spectrum folding. A raised cosine filter with cut-off frequency 6 MHz and roll-off factor 0.5 has been used for this purpose.

More similar testing signal sequences (with different values of frequency offset, data length, SNR..) have been generated for the purpose of design and verification of the design blocks discussed later in this chapter.

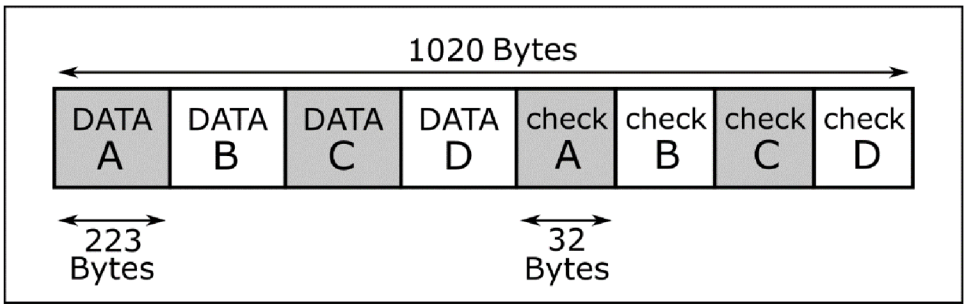


Figure 2-2: Reed-Solomon interleaving scheme.

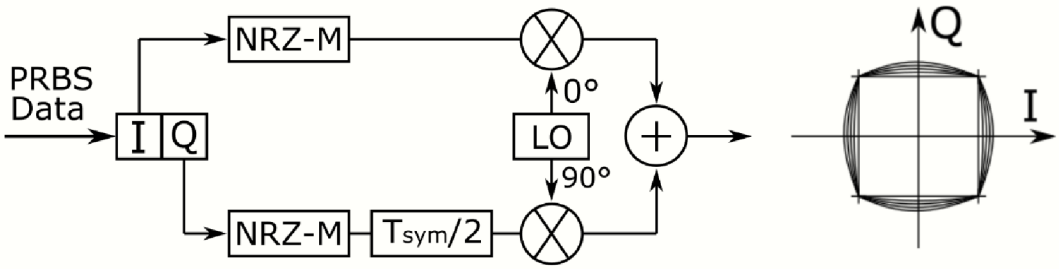


Figure 2-3: SQPSK modulation schematic (left) and I/Q vector diagram (right).

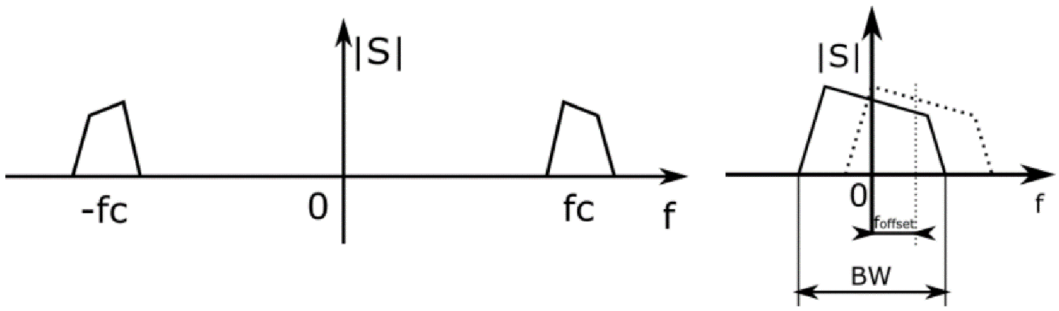


Figure 2-4: Simplified downlink signal spectrum before (left) and after down-conversion (right).

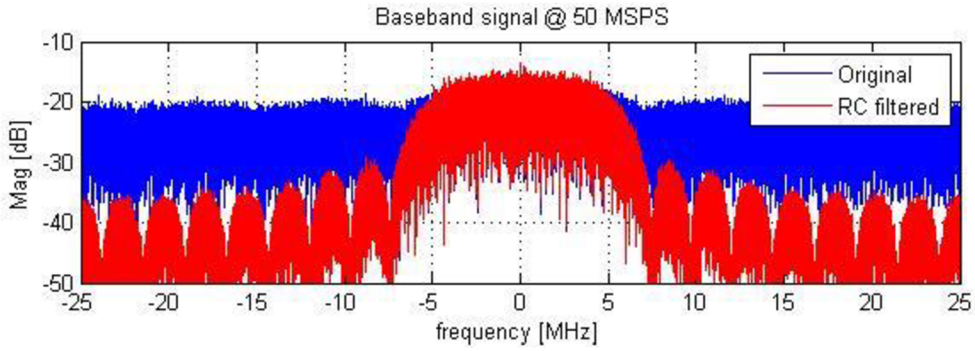


Figure 2-5: Downlink baseband signal spectrum (50 MSPS) before and after filtration.

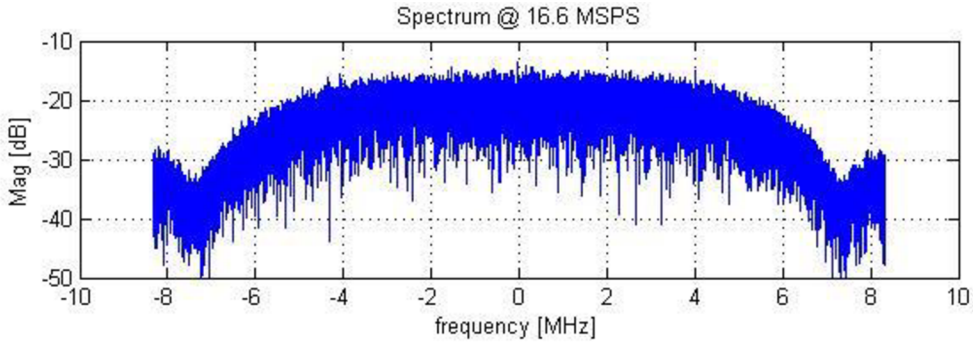


Figure 2-6: Decimated downlink baseband signal spectrum (16.66 MSPS).

2.2 Frequency offset

2.2.1 Doppler frequency shift analysis

One of the first challenges when designing a polar-orbiting satellite signal receiver is the Doppler-effect affecting the carrier frequency – the doppler frequency shift. Thanks to the availability of the satellite orbit position data, the offset frequency calculation is done in the real time.

For the purpose of this thesis the Doppler frequency offset was calculated considering only:

- SC's orbital speed and mean altitude
- GS's altitude

Earth's rotation and other phenomena were neglected. The situation is depicted in figure 2-6. The velocity vector length

$$v_{tan} = \frac{circ}{T_{orbit}} = \frac{2 \cdot \pi \cdot (R_{EARTH} + H_{SC})}{99 \cdot 60} = \frac{2 \cdot \pi \cdot (6378 + 705)}{99 \cdot 60} = 7.492\ km/s$$

is calculated using the Earth's mean radius R_{EARTH}, H_{SC} as the altitude of the SC and T_{orbit} as the orbital period. Figure 2-7 describes the situation where maximal frequency shift occurs – once the SC gets into the line-of-sight with the GS. To calculate the frequency offset it's necessary to get the velocity of SC related to GS. The angle is given by

$$\varphi = \cos^{-1}\left(\frac{R_{EARTH}}{R_{EARTH} + H_{SC}}\right) = 25.78\,°$$

and the line-of-sight vector component of the velocity is

$$v_{SC-GS} = v_{tan} \cdot \cos(\varphi) = 6.746\ km/s.$$

The maximal Doppler frequency shift estimation is given by

$$f_{shift\_max} = f_{carrier} \cdot \frac{v_{SC-GS}}{c - v_{SC-GS}} = 8160 \cdot \frac{6.746}{3 \cdot 10^5 - 6.746} = \pm 183.5\ kHz$$

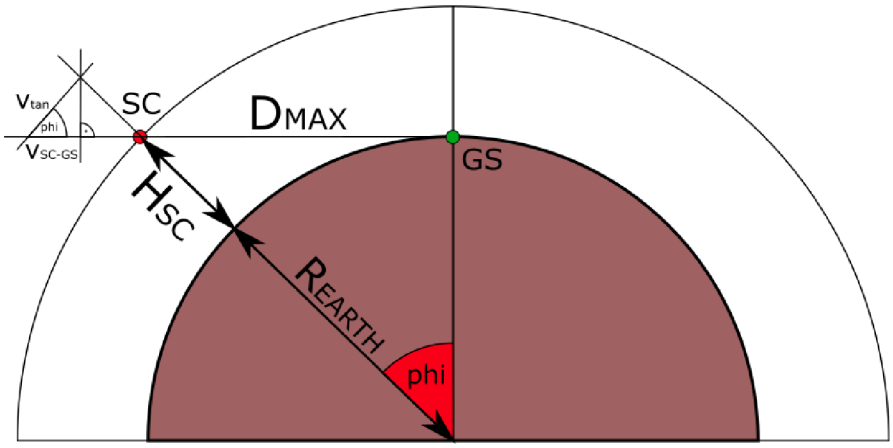


Figure 2-7: Simplified situation of the doppler frequency shift for GS elevation 0°.

where $f_{carrier}$ is the X-band carrier frequency and $c = 3 \cdot 10^5\ km/s$ is the speed of light constant. According to the angle φ, SC is visible to GS for a limited period of time

$$T_{vis} = T_{orbit} \cdot \frac{2 \cdot \varphi}{360} \cong 14\ minutes$$

which is true only if the SC's trajectory leads directly above the GS. Slightly more sophisticated formula was used to calculate the frequency offset as a function of time or elevation (see figure 2-8). As expected, the Doppler frequency shift at 90 degrees of elevation is zero. It is so because the mutual velocity between SC and GS is diminished.

2.2.2 Signal-aided frequency offset removal

In some cases when the received signal is affected by greater, yet unknown value of frequency offset, some more sophisticated mechanisms must be implemented. This is the case especially if the carrier synchronization (Costas loop) ability to track down the frequency offset is not sufficient (the pull-in range is not wide enough).

To prepare the signal for the carrier-phase synchronization a significant amount of frequency offset needs to be removed, otherwise there is a risk that the feedback loop mechanisms will not settle in the locked state. For this purpose a structure called Balanced Quadricorrelator (BQ) was chosen [6][7] (see figure 2-9 for the schematic). The main signal spectra are illustrated by figure 2-10. The input signal is multiplied with a complex exponential signal in the complex multiplication block. This provides the frequency shift compensation. The signal at the output of the multiplication block is given by

$$\hat{c}_x(n) = \hat{c}(n) \cdot e^{i \cdot \tilde{\theta}(n)}$$

where $\hat{c}(n)$ is the input complex signal, $\tilde{\theta}(n)$ is the phase estimation signal and i is the imaginary unit. $\hat{c}_x(n)$ is then filtered by H(z) and $H_D(z)$ where the first filters-out the uninteresting parts of the spectrum and the second gives it the bipolar nature. The error signal (real number) is given by

$$e(n) = re\{\hat{x}_2(n)\} \cdot im\{\hat{x}_1(n)\} - re\{\hat{x}_1(n)\} \cdot im\{\hat{x}_2(n)\}$$

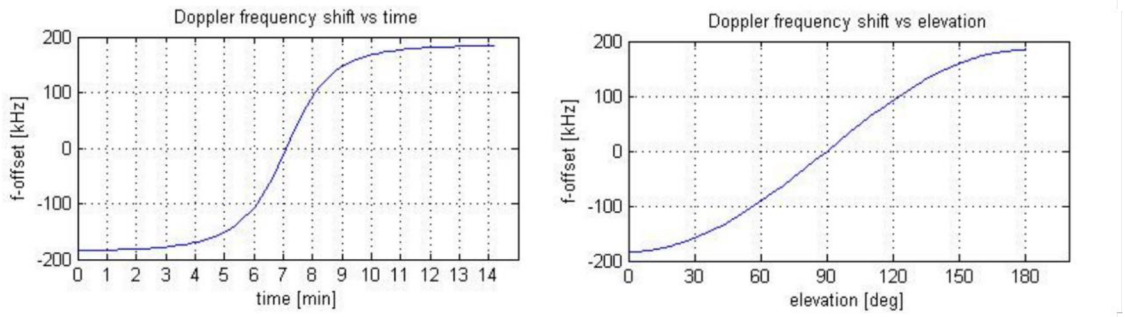


Figure 2-8: Frequency offset as a function of time (left) and elevation (right).

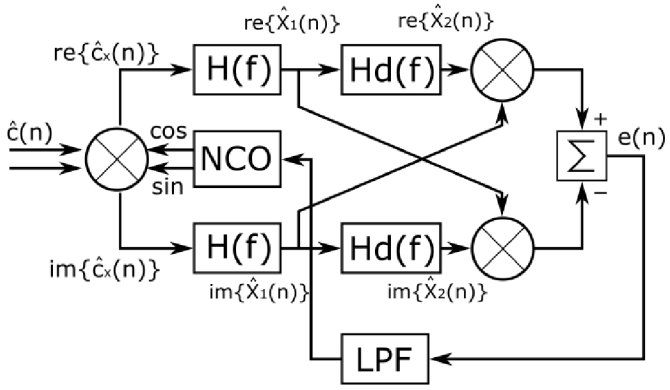


Figure 2-9: Balanced Quadricorrelator circuit schematic (simplified).

where $\hat{x}_1(n)$ is $\hat{c}_x(n)$ after H(z) filtration and $\hat{x}_2(n)$ is $\hat{c}_x(n)$ after H(z) and $H_D(z)$ filtration.

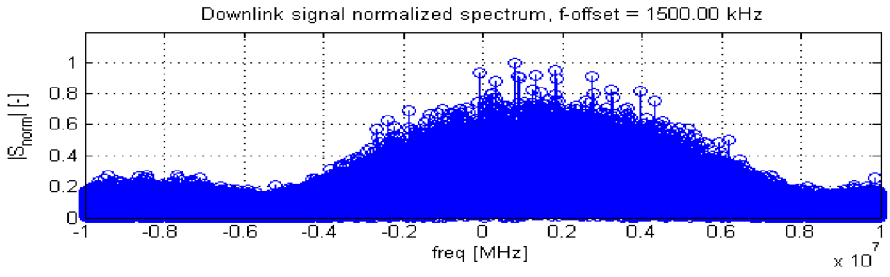
Whenever the input signal has a frequency offset, it is no longer symmetrical and such a disbalance is amplified and manifested as the $e(n)$ signal which needs to be low-pass filtered in the loop filter and converted to the phase estimate $\tilde{\theta}(n)$ signal using the phase accumulator.

It needs to be pointed out that the process of BQ filtering and detection is done in real time (sample-by-sample) otherwise it is not possible to simulate a closed-loop system. Both H and H_D must be implemented as digital filters with finite length of the impulse response.

The ability of BQ to compensate frequency offset is not unlimited. In fact it can be described by empirically found formula

$$\left|f_{BQ-error}\right| > \frac{\left|\delta_{BQ-error}\right|}{100} \cdot BW_{signal} \qquad (1)$$

where $\left|f_{BQ-error}\right|$ is the absolute difference between f_{offset} and f_{NCO} (NCO frequency of the BQ), BW_{signal} is the bandwidth of the input signal and $\left|\delta_{BQ-error}\right|$ is the relative BQ-error found experimentally to be a constant $\left|\delta_{BQ-error}\right| \sim 0.125\%$.



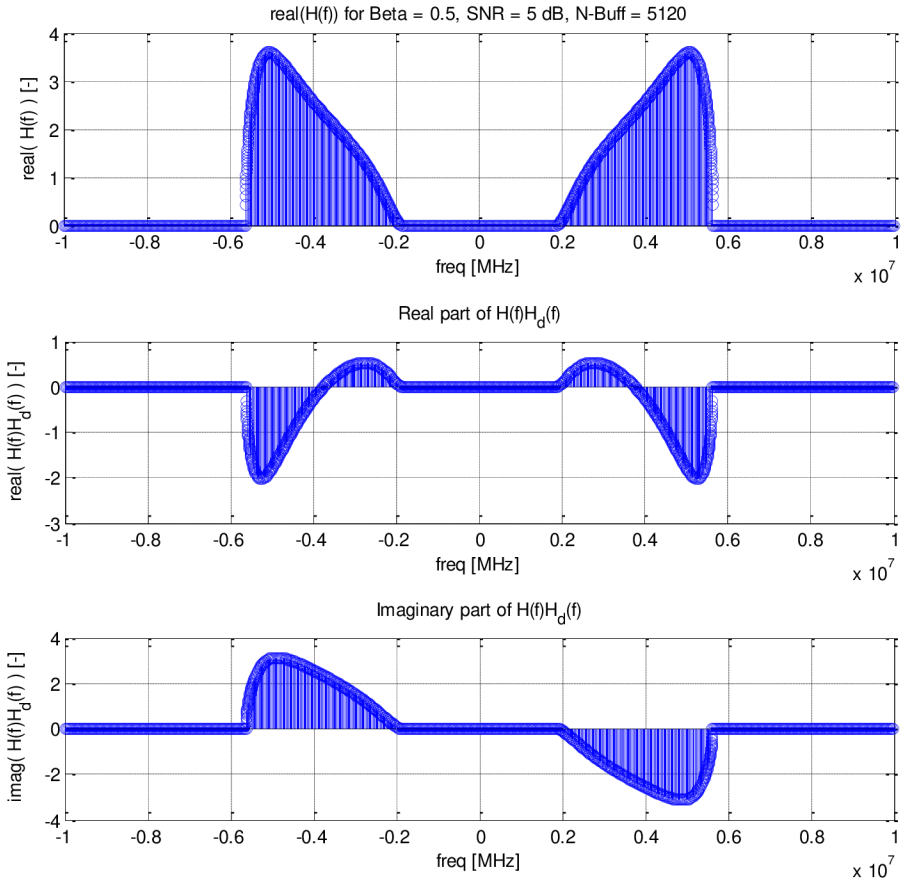


Figure 2-10: BQ input signal spectrum (top), frequency characteristics of the H filter, combined H and HD filters characteristics (bottom).

In other words, if any signal has a frequency offset smaller than 0.125% of its own bandwidth, it cannot be removed using this experimental model of the BQ. This implies that BQ is suitable for compensating bigger offsets (tens of percent) rather than small ones.

As for the loop filter, second order type of PLL filter was used according to [9]. This filter has a latency of one sample, its proportional path gain is K_1 and the integration (accumulation) path gain is K_2 (see figure 2-11 on the left). The positions of both poles

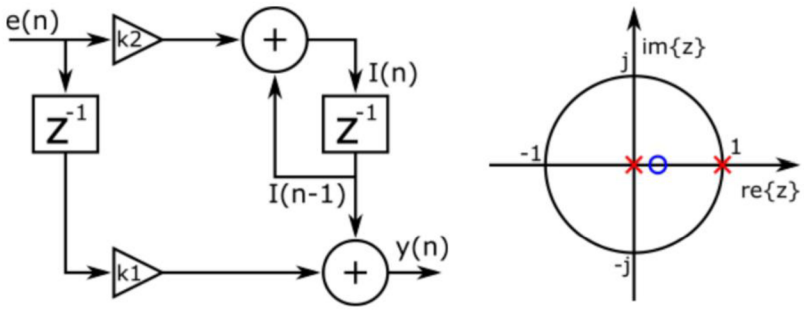


Figure 2-11: BQ Loop filter schematic (left) and Z-plane (right).

are fixed on the real axis (figure 2-11 on the right) but the position of the zero is tunable according to

$$zero = \frac{K_1}{K_1+K_2}.$$

The filter gain then equals $K_1 + K_2$.

As it was said before, the BQ is necessary to remove the major frequency offset. When this is done, the carrier phase synchronization mechanism may take its place. The function of the BQ was examined for various design parameters and an interesting behaviour was found. The whole BQ process can be simplified and described by an exponential function of time

$$f_{NCO}(t) = f_{offset}\left(1 - e^{\frac{-t}{\tau}}\right) \quad (2)$$

where $f_{NCO}(t)$ is a frequency of the NCO, t is the time variable and τ is a time constant. The time constant was found to be an inverse function of the loop filter coefficient K_2 according to

$$\tau \cong \frac{30}{K_2}. \quad (3)$$

The other filter coefficient K_1 affects mainly the noise of the loop. Figure 2-12 shows the BQ process of the offset frequency compensation.

Once the NCO frequency is near enough the offset frequency (which is unknown in the real situation) the BQ attempts to get as close as possible to it, but the nature of the BQ algorithm makes it impossible to remove the offset completely. Figure 2-13 illustrates the BQ process with the offset frequency equal to 1500 kHz. At this point the BQ is stopped and the carrier phase synchronization algorithm is started with the NCO frequency preset value taken from the BQ.

2.2.3 Frequency offset prediction

In case of this project there is a need to know the satellite position in advance to maintain proper aiming of the receiving dish antenna. The information required can be derived from the latest ephemeris updates for each satellite. One of the software

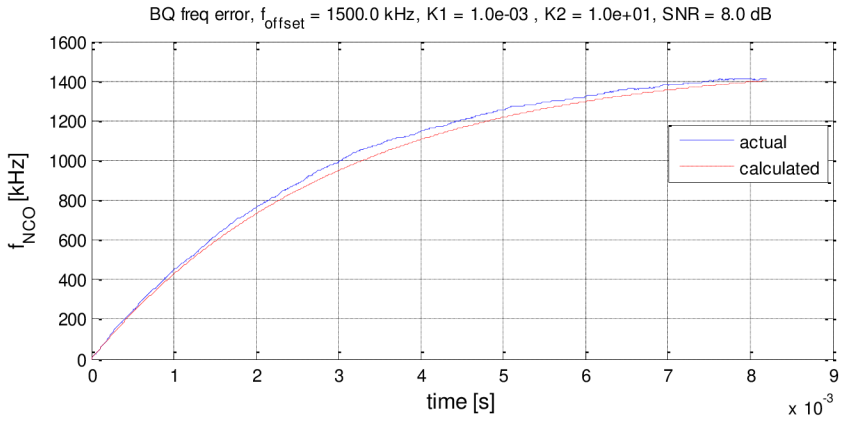


Figure 2-12: The frequency offset removal process.

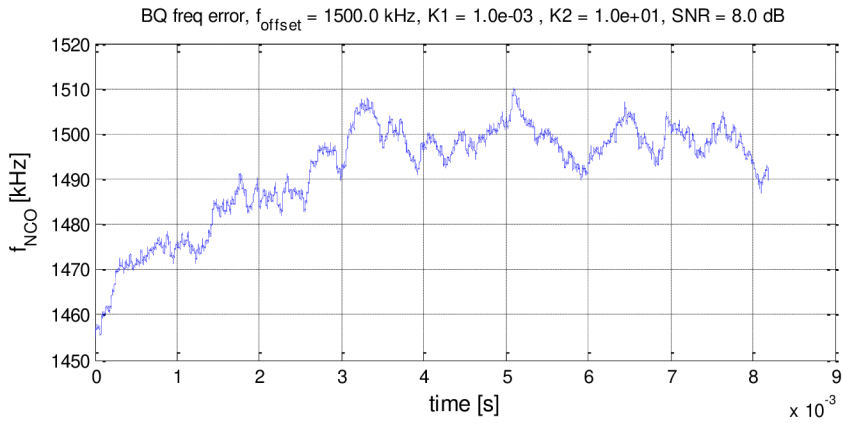


Figure 2-13: Frequency offset removal process (loop noise detail).

applications used for visualizing the ephemerides and satellite tracking is Orbitron. This software also provides frequency offset values for given satellite positions.

Given the starting time, Orbitron provides the appropriate offset frequency value which can be used as the preset NCO frequency value for the carrier synchronization circuit. There will be enough time for it to get into the locked state before the doppler frequency starts changing with greater steepness. From that point on, no other frequency corrections will be necessary because the tracking speed of the carrier synchronization is sufficient.

2.3 Carrier phase synchronization

2.3.1 Costas loop for the O-QPSK

For the purpose of carrier phase synchronization the Costas Loop modified for the offset QPSK was chosen [8]. A simplified schematic is depicted in figure 2-14.

In this case the carrier frequency has been already removed (down conversion) so the input signal is a complex envelope with some residual frequency offset. The input complex signal is multiplied with the complex exponential signal with frequency f_{NCO}. The resulting complex signal $\hat{c}_m(n)$ is filtered by the pulse shaping filter and split into real and imaginary part where the real part is delayed by half of the symbol period (10/9 of a sample). The next design block is called the phase detector (PD) which outputs the error signal $e(n)$ depending on the phase relationship of the I-channel and the Q-channel symbols. The error signal is then filtered in the loop filter and it drives the NCO which closes the feedback loop.

2.3.2 Fractional sample delay

Delaying a signal with non-integer symbol sampling rate (20/9 samples per symbol) by half of the symbol period is not a trivial task. It requires using the interpolation techniques such as the Lagrange polynomial interpolation [13].

Lagrange interpolation of a function is done using polynomial of a given order N. The interpolation polynomial is equivalent to the original function only at $N + 1$ function values (samples of the original function). This knowledge can be used to design the FIR filter which - using a number of input samples - computes a new sample equivalent to the (fictional) input sample delayed by a fraction of the sampling period.

The Lagrange interpolation function of N-th order is given by

$$L_N(x) = l_0(x) \cdot f(x_0) + l_1(x) \cdot f(x_1) + \cdots l_{N+1}(x) \cdot f(x_{N+1})$$

where $f(x_k)$ are the samples of the original function and where the Lagrange basis polynomials are given by

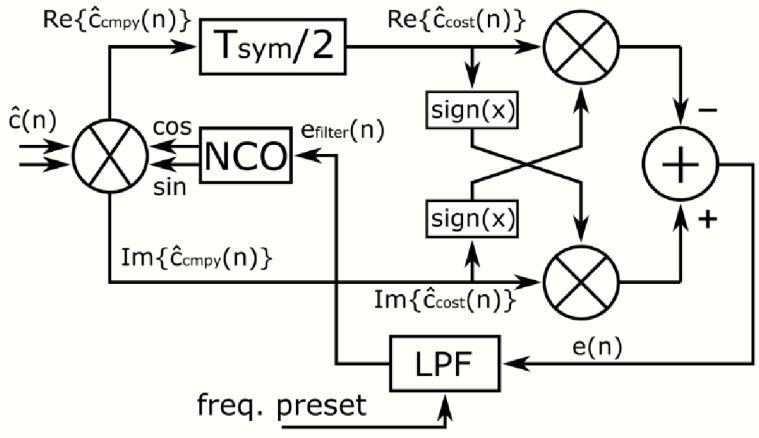


Figure 2-14: Costas Loop schematic (simplified) for the O-QPSK.

$$l_k(x) = \frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_{N+1})}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_{N+1})}$$

where k is the index of the sample and x_k is the position of the sample on the x axis. The basis polynomial $l_k(x)$ is designed this way to ensure that

$$l_k(x_k) = 1$$

and also that

$$l_k(x_{k+n}) = 0.$$

In other words, each Lagrange basis polynomial equals one at the sample position it has been calculated for. In every other sample position it equals zero. As it is a continuous function of x every basis polynomial is non-zero between the sample positions (see figure 2-15).

The quadratic function was used as the interpolation polynomial for the FIR filter. It uses 3 samples so the order of the filter is 3 (see figure 2-16, left). The fractional delay μ is defined as a fraction of the sampling period T measured from the most recent input sample $x(n)$ (see figure 2-16, right).

Half-the-symbol delay block consists of a single sample delay block and the fractional delay block with $\mu = 0.\overline{1}$ (the total delay equals 10/9 of a sample). The complete design process of the fractional delay FIR filter is described in Appendix 1. The resulting FIR filter difference equation (with the additional single-sample delay) is

$$h_{FD}(n) = 0.8395 \cdot x(n-1) + 0.2099 \cdot x(n-2) - 0.0494 \cdot x(n-3).$$

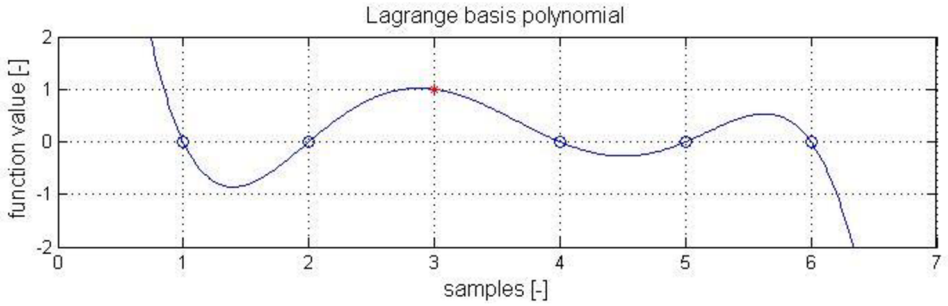


Figure 2-15: Example of the Lagrange basis polynomial of the order 5.

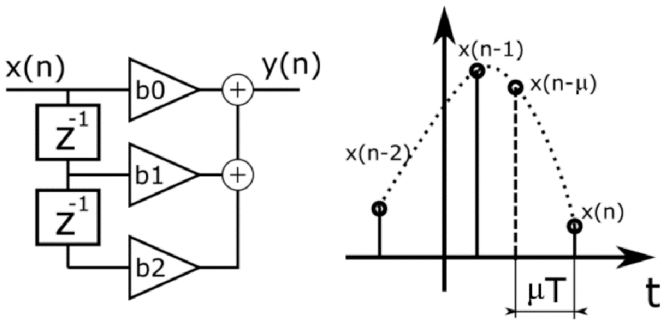


Figure 2-16: Fractional delay FIR filter implementing the quadratic interpolation (left) and its timing diagram (right).

2.3.3 The phase detector

The phase detector (PD) specific for the QPSK (the half-symbol offset has been removed by the preceding delay block) is described by its S-curve [14]. It is a relationship between the mean value of the PD error signal and the actual symbol phase difference $\Delta\theta$ (see figure 2-17, left). The S-curve can be measured in the open-loop mode. It can be clearly seen that the phase ambiguity is ±90 degrees because the slope of the S-curve is identical at -90 °, 0 ° and +90 °. The multiples of 90 ° are the stable points where the phase value always converges during the closed-loop mode of the Costas loop. On the other hand multiples of 90 ° with a 45 ° (negative slope) offset are the unstable points and the phase tends to move to the closest stable point. Unfortunately only one half of the stable points are the correct ones, the rest is to be avoided.

The next thing that must be pointed out is that the Costas loop is a non-linear device. The PD output is given by

$$e(n) = \tanh\big(I(n)\big)Q(n) - \tanh\big(Q(n)\big)I(n)$$

where $I(n)$ and $Q(n)$ are the real and imaginary parts of the complex signal at the multiplier output. For low SNR signals it is essential that the amplitude is greater than or equal to 1 otherwise the signal stays in the linear region of the tanh function (see figure 2-17, right) and the $e(n)$ products subtract and cancel out. The $\tanh(x)$ function can be approximated by $sign(x)$ to make the implementation easier.

2.3.4 Loop behavior analysis

The loop filter used in the Costas loop is similar to the one used in the BQ. The structure of the filter is the same (see figure 2-18), but the tuning is quite different for the Costas loop.

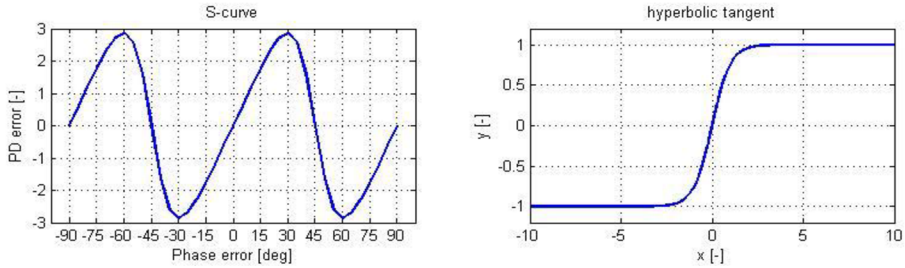


Figure 2-17: The S-curve of the QPSK phase detector (left) and the hyperbolic tangent function (right).

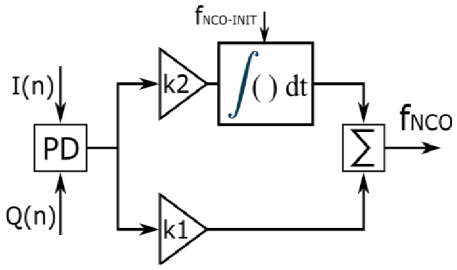


Figure 2-18: Simplified schematic of the Costas loop filter.

The PD output is proportional to the phase error (inside the linear region of the S-curve). When a phase error is detected it is amplified by K_1 and such a signal comes out as a momentary rise or fall of the f_{NCO} which causes the actual phase of the NCO signal to compensate the phase error. This is a mode of operation where $f_{NCO} = f_{offset}$ and it is called the Tracking mode. Another situation is when $f_{NCO} \neq f_{offset}$. The integrator block is loaded with the initial NCO frequency from the BQ and the Costas loop is started. Figure 2-19 describes the situation (MATLAB) where the input signal $SNR = 20\,dB$, $f_{offset} = 15\,kHz$ and $f_{NCO-init} = 7.5\,kHz$. The complex signal at the input of the PD (figure 2-19, top) still contains some residual frequency offset $f_{offset} - f_{NCO}$. The frequency of the beats - the beat frequency [15] - is given by

$$f_{beat} = 4 \cdot |f_{offset} - f_{NCO}|$$

which implies that it is related to the fourth power product of the multiplication.

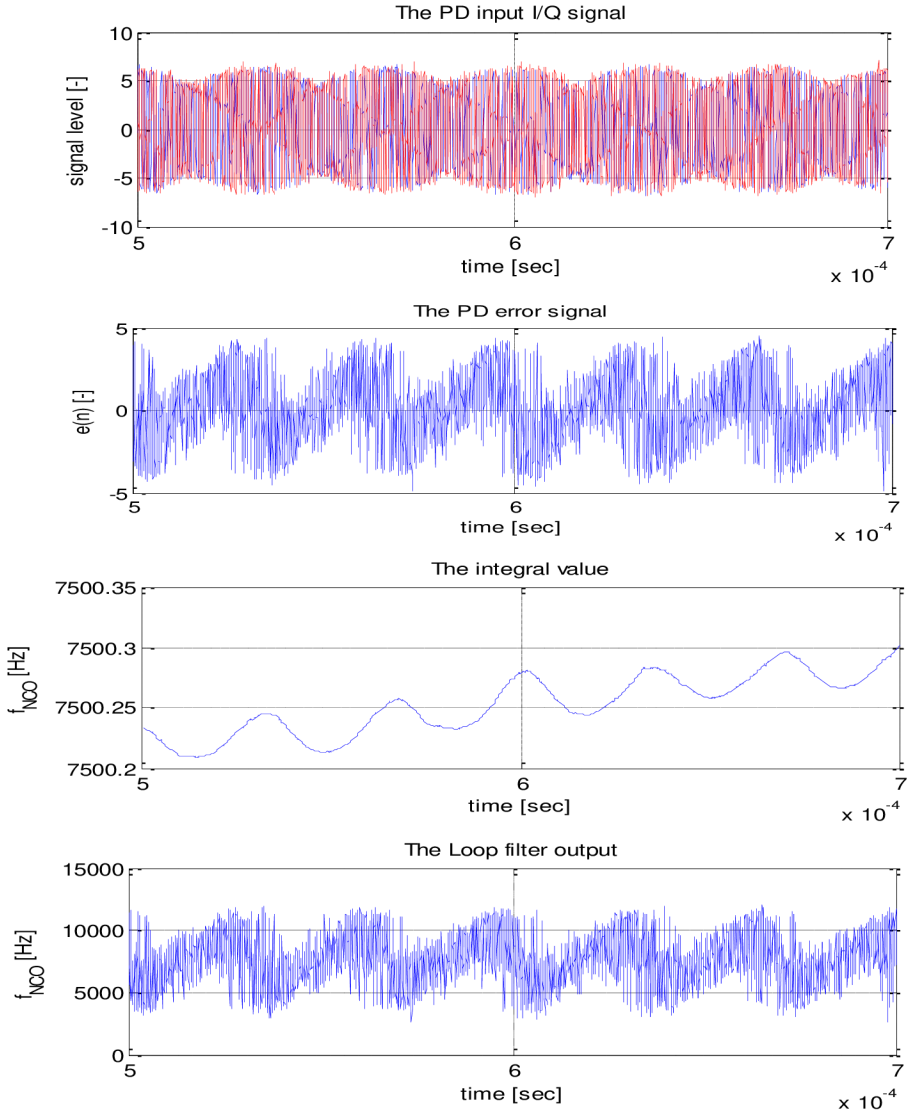


Figure 2-19: The operation of the Costas loop in the acquisition mode. The PD input signal (top), PD output signal, Loop integrator signal and the Loop output signal (bottom).

The beats contain a non-zero mean value which can be integrated (depends on K_2) and eventually achieve the Costas loop phase lock (figure 2-19, second from the bottom). This mechanism is called the pull-in effect [15]. The intensity of the pull-in effect is raised by increasing the K_2 coefficient, but loop oscillations occur when it gets too big. The optimal coefficient values found for this project are $K_1 = 4 \cdot 10^3$ and $K_2 = 10^{-2}$. The coefficients can be defined by analytic formulas [9] but here they were set manually to allow faster simulation debugging.

2.3.5 Testing the loop acquisition speed

Carrier synchronization using the coefficients values given above can handle frequency offsets of ±50 kHz without difficulties, but it remains to be tested whether it is capable of tracking the satellite in the real time and handle the doppler frequency shift.

During the SC flight the Doppler frequency shift is known from figure 2-8. The steepness of the frequency shift (time differentiation) is shown in figure 2-20. The maximal time differentiation of the offset frequency is naturally at the zenith point and it is given by

$$\frac{d(f_{offset})}{dt} = 1.94\ kHz/s = 1.94\ Hz/ms.$$

Assuming that typical loop filter reaction is in units of kHz per millisecond, the influence of the Doppler shift steepness can be safely ignored.

2.3.6 Costas loop output signal

Figure 2-21 depicts the eye diagram at the output of the synchronized Costas loop (I-channel output). The samples-per-symbol ratio is given by

$$\frac{f_{smp}}{R_{sym}} = \frac{16.666\ MSPS}{7.5\ MBaud} = \frac{20}{9}\ \frac{samples}{symbol}$$

where f_{smp} is the sampling frequency and R_{sym} is the symbol rate. For this reason the eye diagram is 20 samples wide and it contains 9 symbols. The eye diagram of the other channel is very similar.

Another way to illustrate the impact of the carrier synchronization on the baseband signal is to use a histogram (see figure 2-22). It is apparent that the carrier synchronization process moves the sample values away from the zero value towards -1 and +1.

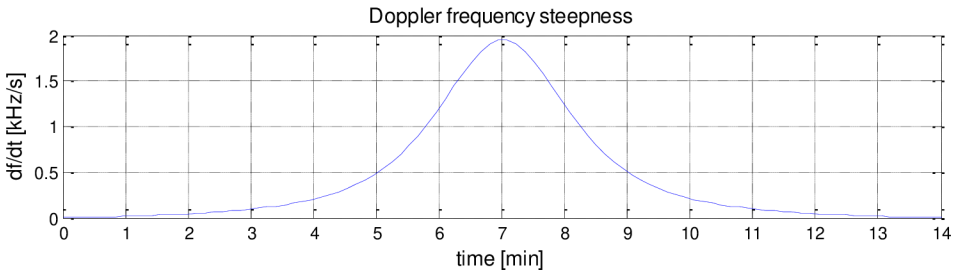


Figure 2-20: Steepness of the Doppler frequency shift during the SC flight.

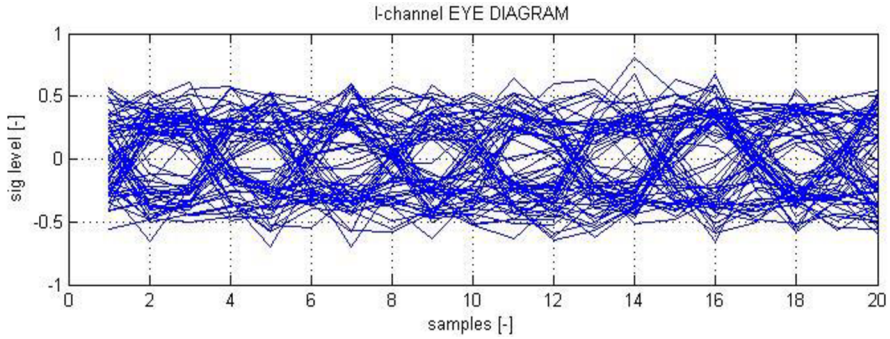


Figure 2-21: Eye diagram of the I channel with a ratio 20/9 sample per symbol.

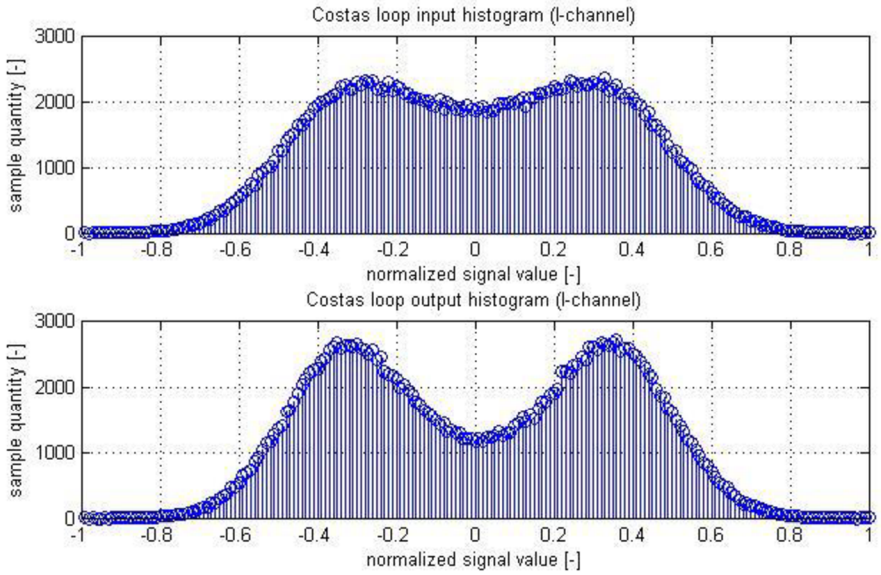


Figure 2-22: Histogram of the Costas loop input (top) and output (bottom) signal.

2.4 Symbol timing synchronization

The next step before obtaining the data in form of the CADU packets is the symbol timing synchronization (STS). This process consists of re-sampling the symbols at the output of the Costas loop using Farrow interpolation structure [12], clock domain crossing FIFO block and Gardner timing error detector [17] all featured in a feedback loop [16].

2.4.1 Farrow structure interpolator

One way of implementing constant fractional sample delay block is to use Lagrange interpolation described in chapter 2.3.2. In order to implement variable delay block of fine resolution it is possible to use the Farrow structure (figure 2-23). The main idea about this structure is reducing the number of multipliers by adjusting the difference equation of the interpolation FIR filter. Given the equation of the quadratic interpolation filter, following adjustments are made:

$$h_{FD}(n) = l_2 \cdot x(n) + l_1 \cdot x(n-1) + l_0 \cdot x(n-2)$$

Making the following substitutions $l_0 = \frac{\mu \cdot (\mu - 1)}{2}$, $l_1 = \mu \cdot (2 - \mu)$ and $l_2 = \frac{(\mu - 2) \cdot (\mu - 1)}{2}$ results in a complex function. Adjusting this function as a polynomial of μ yields

$$h_{FD}(n) = (X_A(n) \cdot \mu + X_B(n)) \cdot \mu + X_C(n).$$

There $X_A(n)$, $X_B(n)$ and $X_C(n)$ are output signals of the 3rd order FIR filters with constant coefficients given by

$h_A = [0.5, -1, 0.5]$, $h_B = [-1.5, 2, -0.5]$ and $h_C = [1, 0, 0]$.

2.4.2 Symbol resampling method

Interpolation techniques are frequently used for realizing fractional-sample delay blocks. In order to change the fractional symbol-per-sample ratio (such as 20/9) to the integer one (such as 2.0), more sophisticated structure must be used. The principle of fractional

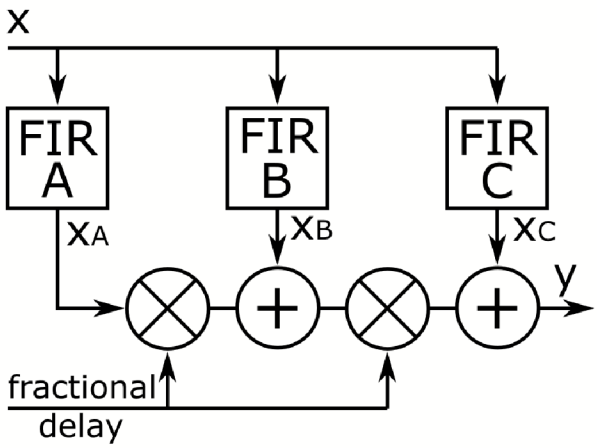


Figure 2-23: Farrow interpolation structure schematic.

resampling (using the above given symbol-per-sample ratios) is illustrated in figure 2-24.

The waveform on the top contains input samples, output samples and the signal envelope (non-discrete). The waveform below represents fractional-sample delay μ between input and output samples. This delay is decreased by $step$ value with every input clock cycle. The decrement value is given by

$$step = 1 - \frac{N_{OUT}}{N_{INP}}$$

$$step = 1 - \frac{2}{2.\overline{2}} = 0.1\ [-]$$

where N_{OUT} is the output sample-per-symbol ratio and N_{INP} is the input one. μ is limited to the range $\langle 0,1 \rangle$ and every time it underflows, the next input sample is omitted (see the red cross in the waveform). Figure 2-25 shows the block design schematic of the Farrow re-sampler. This type of fractional re-sampler may be generalized to any input/output symbol-per-sample ratio respecting the condition $N_{OUT} < N_{INP}$. The Farrow structure schematic is

Setting the correct $step$ value is not enough to provide the optimal symbol timing moments, because neither N_{INP} nor $step$ are infinitely precise. The resampling mechanism is best when working in a feedback loop with the TED (Timing Error Detector).

2.4.3 Gardner Timing Error Detector

One kind of TED frequently used with a symbol sampling ratio 2.0 is the Gardner TED. A schematic of the QPSK Gardner TED is in figure 2-26 (left). The corresponding formula for the error signal is

$$e(n) = I(n-1) \cdot \big(I(n) - I(n-2)\big) + Q(n-1) \cdot \big(Q(n) - Q(n-2)\big)$$

where $e(n)$ is the error signal, $I(n)$ and $Q(n)$ are the input quadrature signals. The desired state of the synchronized signal (TED error is minimal) is when every odd sample is close

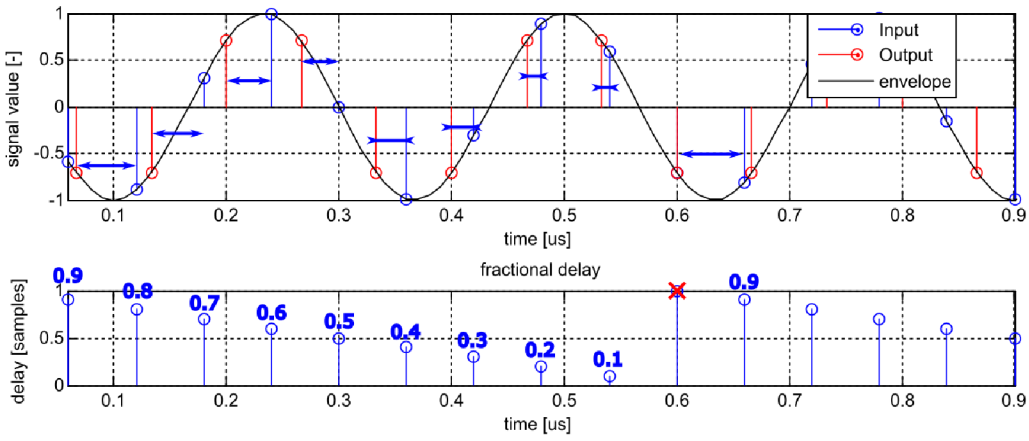


Figure 2-24: Fractional resampling method explanation.

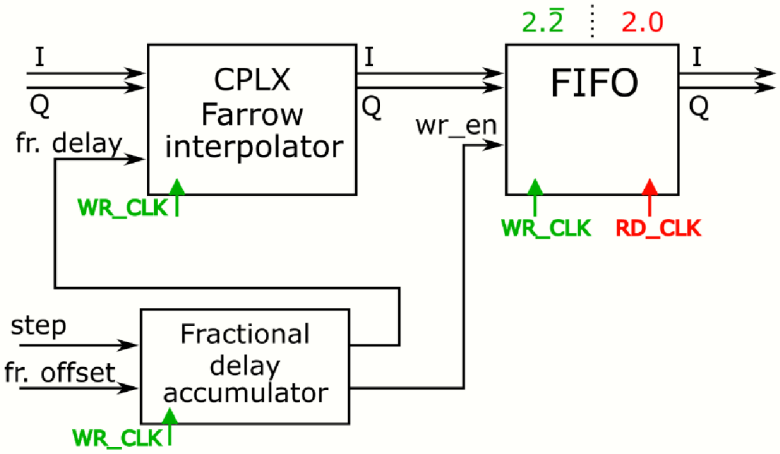


Figure 2-25: Farrow re-sampler block schematic.

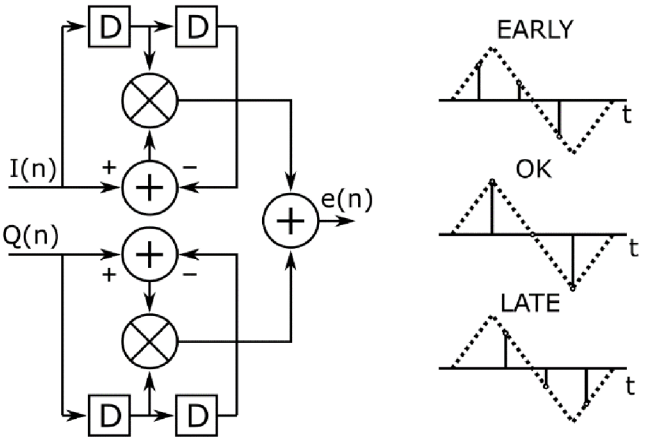


Figure 2-26: Gardner TED schematic (left) and symbol timing situations (right).

to zero and every even sample is very close to the ideal symbol constellations (±1). If the symbol timing gets out of sync (early or late, see figure 2-26 on the right) the TED error signal increases towards either positive or negative value.

It is very important that only every second TED output sample is processed into the feedback loop and every other sample is ignored (the STS feedback loop clock frequency is two times smaller than the frequency of the resampler's output). The S-curve of the TED for this case is shown in figure 2-27 (top waveform - each curve for either odd or even output sample, bottom waveform – one curve for both odd and even samples). If all the output samples are used (the STS feedback loop clock frequency is not divided by two), the S-curve value is negated with every other sample and the feedback loop oscillations occur.

Another important features of the STS feedback loop are the polarity of the STS feedback loop and the choice of either odd or even output sample of the TED. Periodic nature of both S-curves implies that no matter which sample is chosen, the STS loop always ends up in the locked state. The same rule applies to the STS feedback polarity, which can be

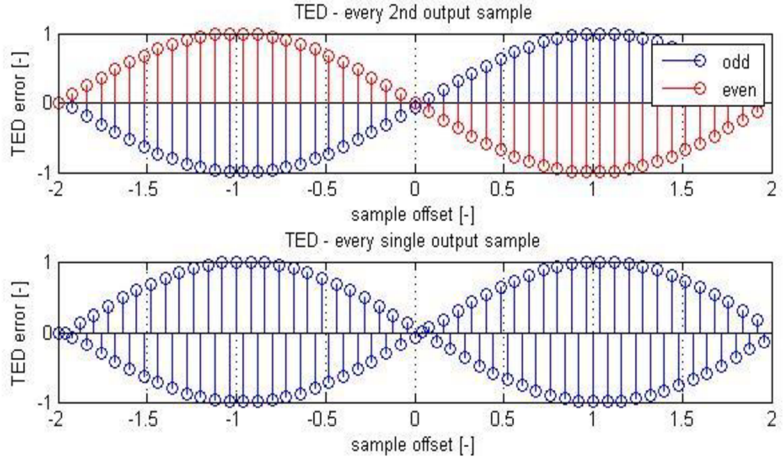


Figure 2-27: Gardner TED S-curves.

imagined by reversing the polarity of the S-curve. Changing the S-curve's polarity is essentially equivalent to choosing the odd sample instead of the even one. Therefore, all four combinations of the parameters described above bring the same results.

2.4.4 Symbol Timing Loop Filter

The STS loop filter is processing the TED output signal in order to provide stability and optimal reaction speed of the STS feedback loop. The schematic is identical to the Costas loop filter with only one difference – it operates on half the clock frequency.

It is a second order filter using proportional and integral blocks. Unlike the Costas loop filter, this filter keeps the integrator mean value equal to zero (Costas loop filter accumulates the offset frequency value). Both proportional and integral gains were found experimentally and their values are:

$$KP_{STS} = 1/8, KI_{STS} = 1/512.$$

The values are chosen as powers of 2 for easier implementation (bit shifting). Figure 2-28 shows the STS loop filter waveforms and Figure 2-29 shows IQ diagrams of the STS output signal. Taken only the odd samples, the IQ diagram (on the left) clearly distinguishes the symbols. However, choosing only the even samples results in completely incomprehensible IQ diagram (on the right). The choice between odd and even samples is made in the following design block. Figure 2-30 illustrates the whole STS feedback loop system.

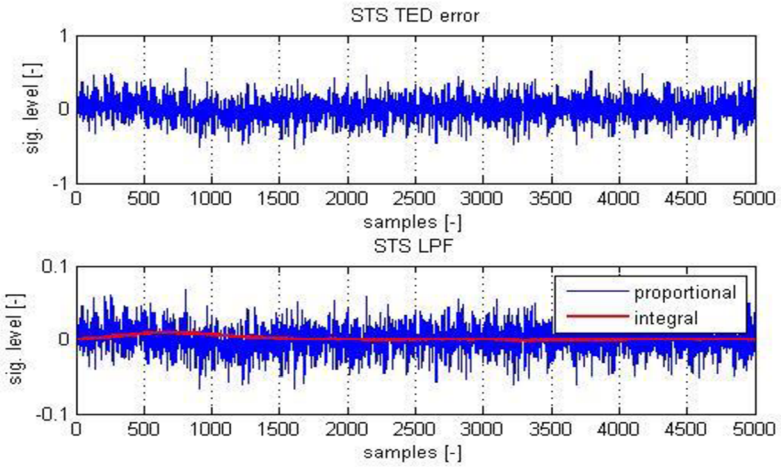


Figure 2-28: STS loop filter timing diagrams.

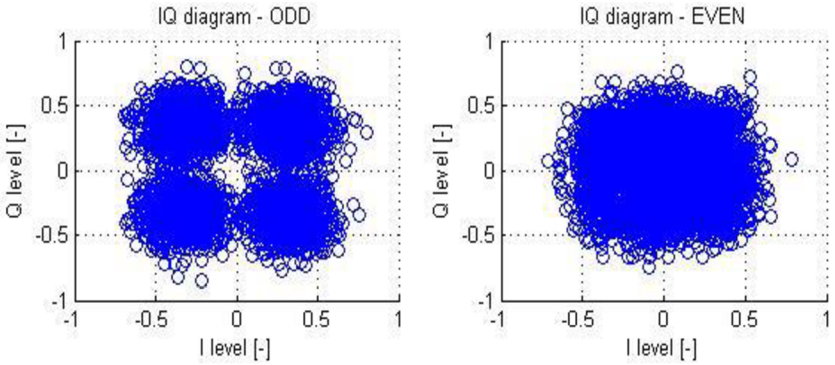


Figure 2-29: IQ diagram for odd (left) and even sample selection.

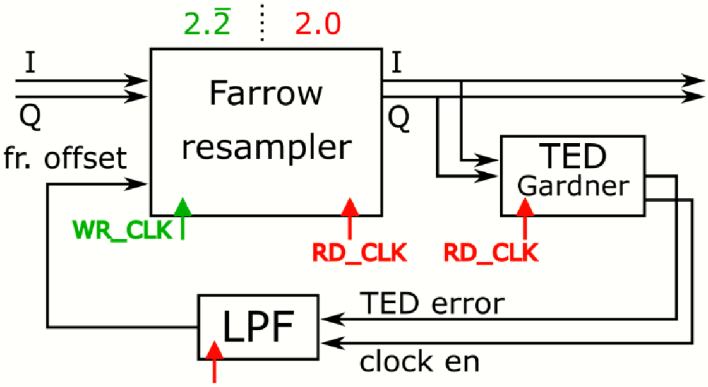


Figure 2-30: STS loop block schematic.

2.5 ASM detection

The use of quadrature modulation (in this case SQPSK) brings also some issues where the most significant one is the carrier phase ambiguity. One way to improve this problem from four possible phase values to just two values is using the NRZ-M (differential) encoding. To eliminate this problem completely, the ASM detection must be done by specific type of complex correlation.

2.5.1 Detection and NRZ-M decoding

The input signal must be first detected (rounded to ±1) and differentially decoded from NRZ-M to NRZ-L (still considering 2.0 samples per symbol). For sample detection a simple *signum* function is used. NRZ-M decoding is done by one-bit multiplication of two following symbols. Figure 2-31 illustrates this situation for a single channel (both I and Q decoders are identical). The multiplication block is used in ±1 logic and it is identical to XOR operation in 0/1 logic.

2.5.2 ASM correlation

The ASM in hexadecimal form equals 0x1ACFFC1D according to [5]. Its autocorrelation properties are significant, but considering the fact the signal has been split into I/Q channels, the autocorrelation properties of the partial ASM signals are not as good as the whole signal. Figure 2-32 depicts the autocorrelation results of the respective ASM parts (top) and the whole ASM (bottom). Both parts of the ASM can be written in hexadecimal form as $ASM_I = 0x47DC$ and $ASM_Q = 0xE7D2$. The ASMs are already bit-reversed from left to right to match the shift-register-based correlation technique.

In order to remove the phase ambiguity, each ASM part must be correlated with both I and Q input channels. This implies the need of total four simultaneous correlation procedures. If the carrier phase is 90/270 degrees, the correlation peaks will appear in 2 out of 4 resulting signals. In case of 0/180 degrees phase, the peaks will appear in the other two resulting signals. The four correlation outputs are named $R_{II}, R_{QI}, R_{IQ}, R_{QQ}$ and the correlation peaks appear either in $R_{II}\&R_{QQ}$ or $R_{QI}\&R_{IQ}$. Moreover, in $R_{QI}\&R_{IQ}$ the peaks are synchronized in time and in $R_{II}\&R_{QQ}$ the timing is off by 2 samples (see Appendix 2 for a detailed explanation). Figure 2-33 shows block schematic of the ASM detection circuit. The two-sample delay block at the Q channel input fixes the $R_{II}\&R_{QQ}$ timing problem. The correlation output values are integer (caused by sample detection) and the maximal value is given by the bit-length of the partial ASM (±16). The maximal values of the combination outputs (R_{IIQQ} and R_{IQQI}) are ±32. The combined correlation waveforms together with the ASM threshold are displayed in figure 2-34.

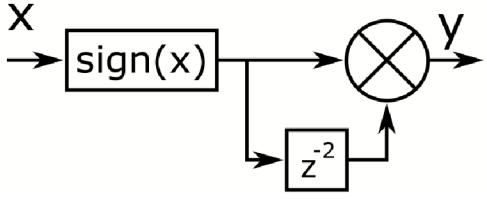


Figure 2-31: Sample detection and NRZ-M decoder (one channel).

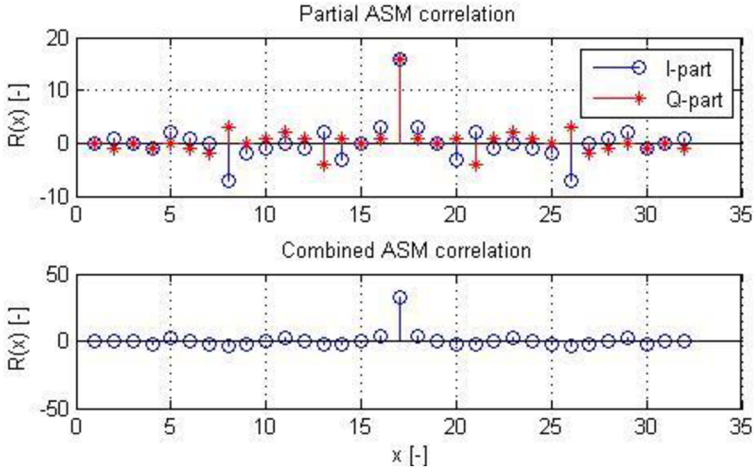


Figure 2-32: Partial and combined ASM autocorrelation.

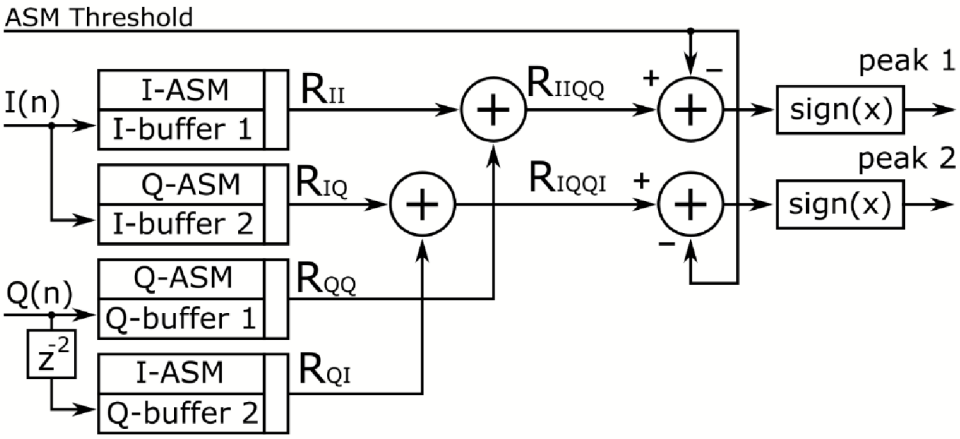


Figure 2-33: ASM correlation and peak detection schematic.

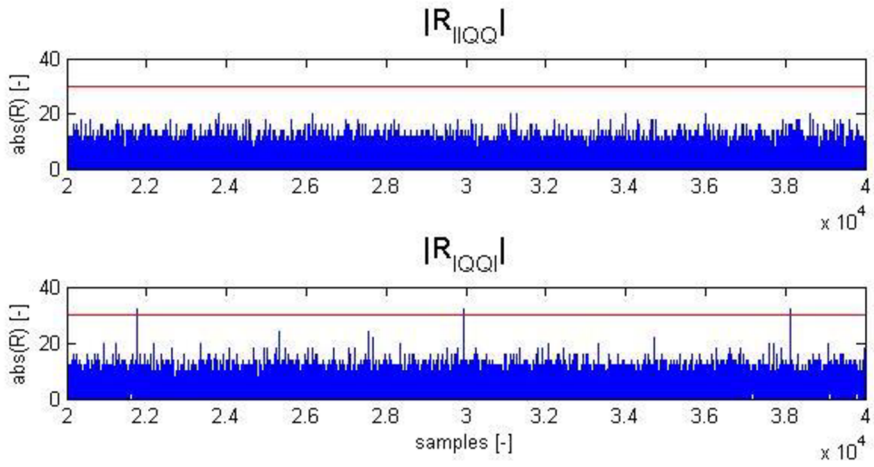


Figure 2-34: ASM combined correlation outputs.

2.5.3 ASM peak detection and data serialization

Most of the signal detection errors caused by low SNR can be fixed by the Reed-Solomon decoder. However, this kind of data protection is useless in case of the ASM detection failure. If a false peak is detected, the whole CADU is corrupted and lost.

In order to preserve the expected rhythm of incoming correlation peaks (once per 8192 samples) a peak detection masking mechanism has been designed. Whenever a false peak appears (out of rhythm), it is ignored thanks to this mechanism.

The key element is the binary counter with the limit value 8160. The counter is triggered by the peak detection event, which at the same time enables the masking flag signal. This signal stays enabled until the counter elapses (32 samples before the next expected peak). If the peak does not come as expected, the masking flag stays disabled until the next peak detection event (schematic in figure 2-35). Another component is the synchronous R-S flip-flop with reset priority. Its truth table is displayed in table 2-1.

Table 2-1: Reset priority R-S flip-flop truth table.

S	R	Q(n)
0	0	Q(n-1)
0	1	0
1	0	1
1	1	0

Whenever the peak occurs (A or B), the appropriate flip-flop is set and controls the data multiplexor (figure 2-36). The **data enable** signal is used in the following design block (figure 2-36) and remains set high as long as the counter is running. The principle of data formatting is displayed in figure 2-37. Using the double input D-flip-flop with clock enable helps to choose only the odd/even samples and the data serialization is done by the output multiplexor.

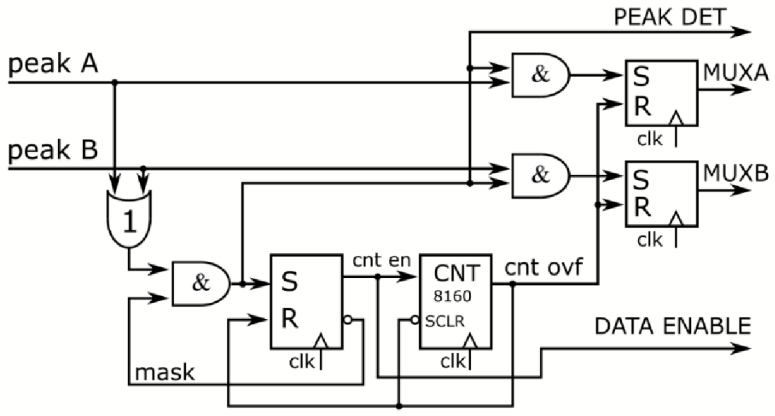


Figure 2-35: ASM Peak detection schematic.

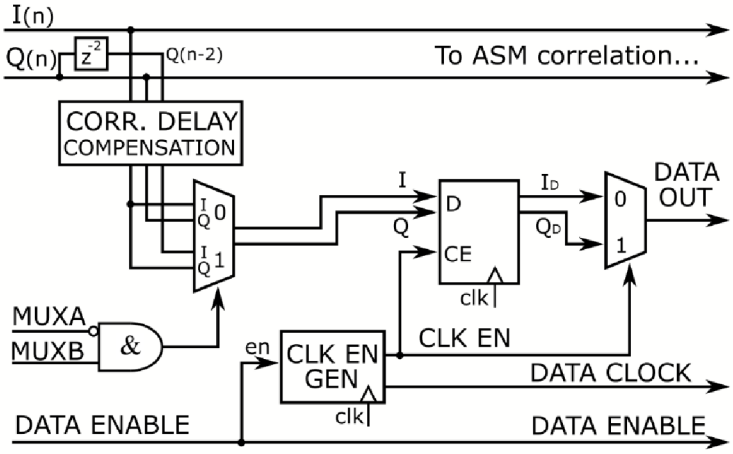


Figure 2-36: ASM detector data formatter schematic.

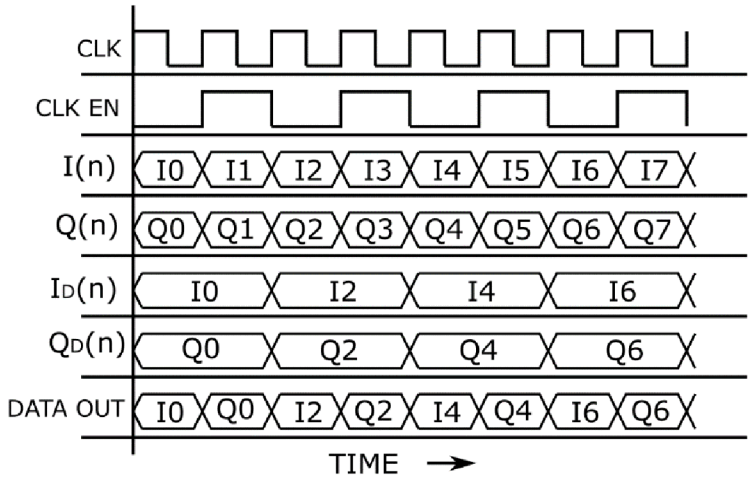


Figure 2-37: Data formatter simplified timing diagram.

2.5.4 PRBS decoding

The data stream of CADU is PRBS encoded (randomized) on the SC (except for the ASM) to improve the spectral purity by suppressing the data periodicity. The decoding process is depicted in figure 2-38. The input serial data stream is PRBS decoded by the logic exclusive OR (XOR) operation with the original PRBS sequence.

The PRBS generator (figure 2-39) consists of 8-bit shift register with four taps which are combined using the XOR operation and the result is fed into the shift register input.

The PRBS sequence output is generated from the last (right) D flip-flop (single bit register). The PRBS schematic corresponds to the polynomial

$$x^8 + x^7 + x^5 + x^3 + 1$$

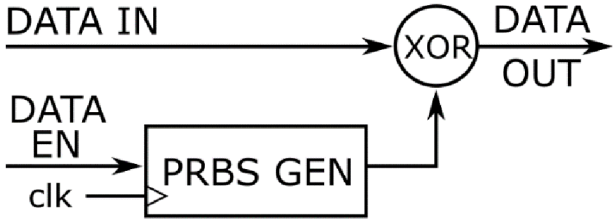


Figure 2-38: PRBS decoding schematic.

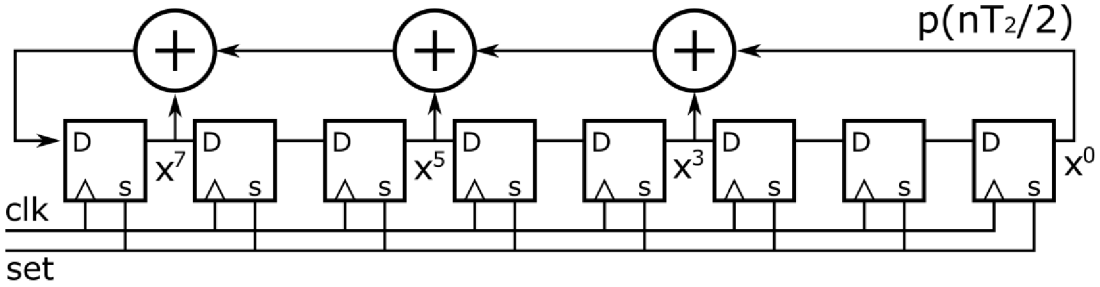


Figure 2-39: PRBS generator schematic.

where the highest product (here it is x^8) is always omitted in the implementation. The first 40 bits of the sequence are

1111 1111 0100 1000 0000 1110 1100 0000 1001 1010.

according to [4]. Each D flip-flop has the clock input sensitive to the rising edge of the clock signal and the set input which re-initializes the output value to logic one. This happens in every ASM detection moment (beginning of the CADU).

3 FPGA IMPLEMENTATION

This chapter describes the design implementation into Xilinx FPGA Zynq-7000 placed on the ZYBO z7 7020 development board. Fixed-point 16-bit arithmetic was used for the main signal flow and all the design blocks (except for the IP cores) were designed by the author of this thesis. The Xilinx Vivado 2017.4 software was used for the FPGA RTL (Register-Transfer Level) design using Verilog as the coding language.

3.1 Costas loop

3.1.1 Arithmetic multiplication and division by a constant

The most resource-demanding DSP design blocks are multiplication and division. In some applications the variability of coefficients used in those operations is reduced to multiples of 2 and both operations can be implemented by bit-shifting.

Given an 8-bit two's complement number $A_{bin} = 00111011$, it can be represented by various numbers in other numeric bases, such as:

- signed-integer number $A_{Int8} = 59$,
- Q7 fractional number $A_{Q7} = 59/256 = 0.23046875$.

Dividing this number by 2 (shift right by 1 bit) results in $A_{bin}/2 = 00011101$:

- $A_{Int8}/2 = 29$,
- $A_{Q7}/2 = 59/256 = 0.11328125$.

Given another 8-bit two's complement number $B_{bin} = 11000101$, it can be represented in other numeric bases as:

- signed-integer number $B_{Int8} = -59$,
- Q7 fractional number $B_{Q7} = -\frac{59}{256} = -0.23046875$.

Dividing this number by 2 results in $B_{bin}/2 = 11100010$:

- $B_{Int8}/2 = -30$,
- $B_{Q7}/2 = -30/256 = -0.1171875$.

By comparing numbers $A_{Q7}/2$ and $B_{Q7}/2$ it is clear that cutting off the division results (8-bit division result instead of 9-bit) accomplishes the mathematical function known as round towards negative infinity or **floor.** The floor operation is unwanted in DSP since it produces unbalanced division results. It is of much bigger benefit to use round towards zero, also known as **truncation.**

The first two design blocks (Round-to-zero and bit-shift) are shown in figure 3-1. The bit-shift block is just a simple wired connection without any clock cycle latency. The Round-to-zero (R2Z) block uses the bit-shift block for Q15 and according to sign of the input signal it chooses between the bit-shifted result and output of the 16-bit adder. The adder latency is an optional design parameter of the adder IP core. Increasing the latency

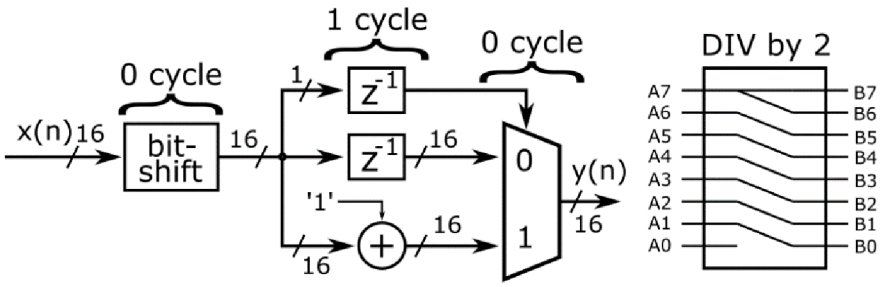


Figure 3-1: Round-to-zero (left) and bit-shift block schematic.

of the adder also increases its maximum operating frequency. The delay blocks are used to compensate this latency. Total latency of the whole block is 1 clock cycle. The R2Z block is used in many other design block in this project.

3.1.2 Complex multiplier block

This block is geneated as a Xilinx IP core and its pinout is described in figure 3-2 on the left. In order to simplify the access to this block, another design block was made above it (figure 3-2 on the right).

The complex multiplier does the multiplication of two complex number inputs according to this formula:

$$y_I = A_I \cdot B_I - A_Q \cdot B_Q,$$

$$y_Q = A_I \cdot B_Q + A_Q \cdot B_I.$$

Each of the inputs is 16-bit wide (32 bits per complex input), so the output is twice the input width. The B input is connected to the DDS block (described in chapter 3.1.3) with outputs only 10 bits wide (to reduce the resources), so the CMPY top block adds zeros into 6 least significant bits. The output width is cut off to match the 16-bit signal width.

3.1.3 DDS block

Another Xilinx IP core is the DDS (Direct Digital Synthesis). It consists of two main

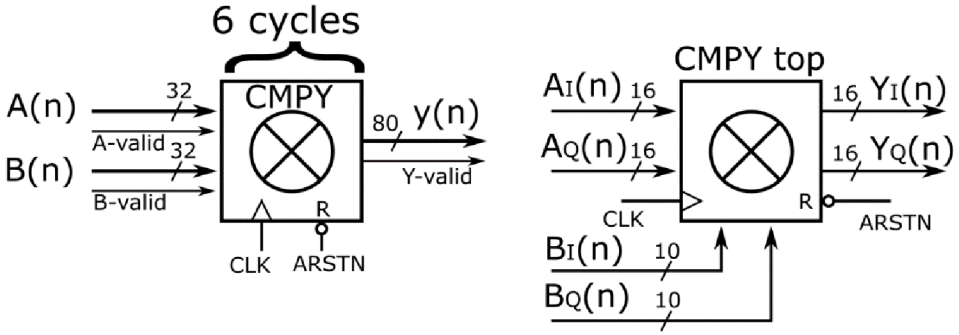


Figure 3-2: Complex multiplier IP core (left) and CMPY top (right).

parts: the phase accumulator and the Look-up table (LUT). The DDS in this project has to generate the complex exponential signal described by:

$$e^{j\theta(n)} = \cos\theta(n) + \sin\theta(n).$$

Phase accumulator (PA) is designed to have variable phase increment $\Delta\theta(n)$ so it is possible to dynamically change the output frequency f_{DDS}. The output frequency is given by

$$f_{DDS} = f_{clk} \cdot 2^{-WPA} \cdot \Delta\theta(n),$$

where WPA is the bit-width of the PA (16 bits), and f_{clk} is the clock frequency. The DS frequency resolution is given by

$$\Delta f_{DDS} = f_{clk} \cdot 2^{-WPA} = 16.666 \cdot 10^6 \cdot 2^{-16} = 254.3\ Hz.$$

Xilinx DDS compiler IP core provides many configuration options which allow to generate either complete DDS with PA and LUT or just one of them. Since the PA IP core does not support negative values of the phase increment (specific design requirement; the frequency offset changes sign due to the satellite position), it had to be designed manually and the IP core was configured as an LUT. Figure 3-3 shows the PA block schematic. The phase increment $\Delta\theta$ and phase offset input range is from -32768 to 32767. The phase output value θ increases/decreases by $\Delta\theta$ with each clock cycle and its range is defined from 0 to 65535. Each full PA accumulation cycle generates one period of cosine and sine signal at the output of the LUT.

3.1.4 Half-symbol delay block

As it was mentioned in chapter 2.3.2, one way to implement delay by fraction of a sample is to use quadratic Lagrange interpolation FIR filter. Using Xilinx FIR compiler design eases the work of designing the filter by just loading the coefficient vector and setting the order of the FIR filter. Here, the order is 4 and the coefficient vector is

$$[0, 0.839508, 0.20987, -0.049377].$$

The first coefficient equals zero to provide additional one-sample delay (total delay is 1.111 samples). The latency of the filter is 10 cycles and it must be compensated in the Q channel using a FIFO memory delay block (see figure 3-4 on the left). The FIFO requires another control block (FIFO CTRL) to synchronize the write and read enable signals correctly so the delay between input and output samples is exactly 10 clock cycles.

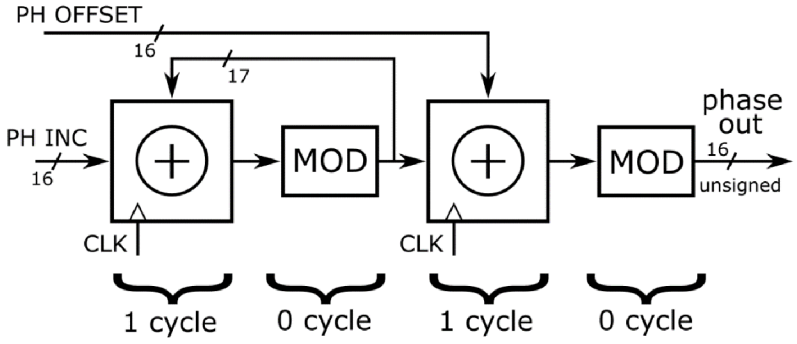


Figure 3-3: DDS phase accumulator block schematic.

Total latency of the Half-symbol delay block is 10 clock cycles.

3.1.5 Phase detector block

The phase detector for the QPSK Costas loop consists of two multipliers and one subtractor. Considering the fact that one of the multiplier's input comes from the signum function, the multiplier can be replaced with much simpler structure with multiplexor and 16-bit negation block (see figure 3-4 on the right). The NEG block in the schematic operates as a simple bitwise negation and after that the LSB is added to the result using the ADD block. This procedure is identical to two's complement signal negation so the multiplexer chooses the positive/negative version of the input channel (I/Q) according to the sign (MSB) of the other input channel.

This phase detector design has been verified for any over/under-flow events. Total latency is 2 clock cycles.

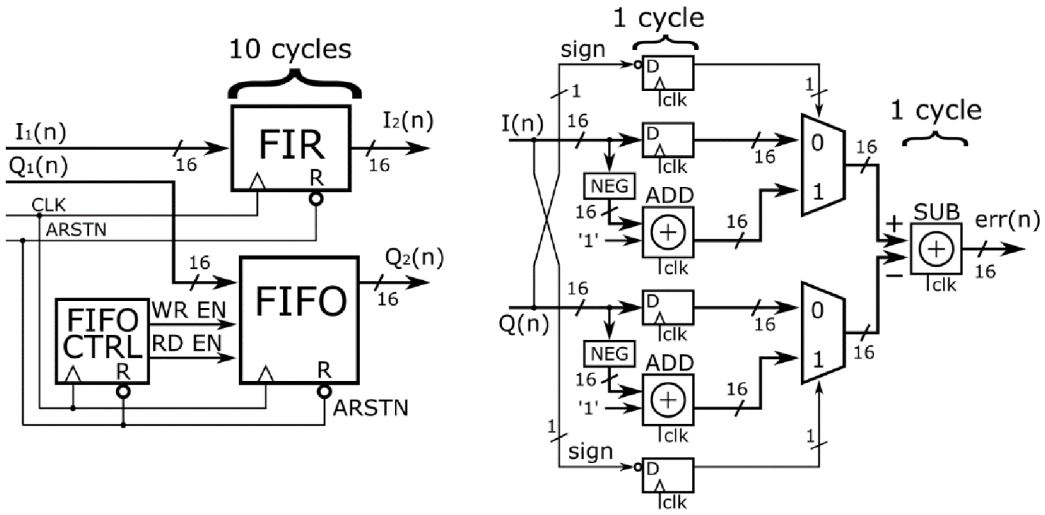


Figure 3-4: Half-symbol delay (left) and Phase detector block schematic.

3.1.6 Costas loop filter block

This is the most difficult block to implement, because it makes the feedback loop complete and it has to convert the input error signal (phase detector) to the output phase increment value (DDS). In order to tune the loop filter identically to the MATLAB simulation, certain conversion formulas were derived for the loop coefficients.

The filter function is similar to PI controller (Proportional-Integral) where the proportional and integral gain is realized by R2Z block (see chapter 3.1.1) with specific division ratios NP and $NI1$. Detailed schematic of the filter is in figure 3-5. The input error signal is split and divided by 2^{NI1}(Integral gain) and 2^{NP} (Proportional gain). The ADD1 adder accumulates the output of the R2Z block and continue to the cascade of multiplexers. Their function is to prevent the overflow and underflow events and also to enable loading the I_{LOAD} preset value into the I_{SAT}value which is proportional to the frequency offset. The following R2Z block provides further division to the signal before it becomes the phase increment of the DDS. The proportional branch compensates the delay of the integral branch so they can both be added in ADD2. The Saturation block is

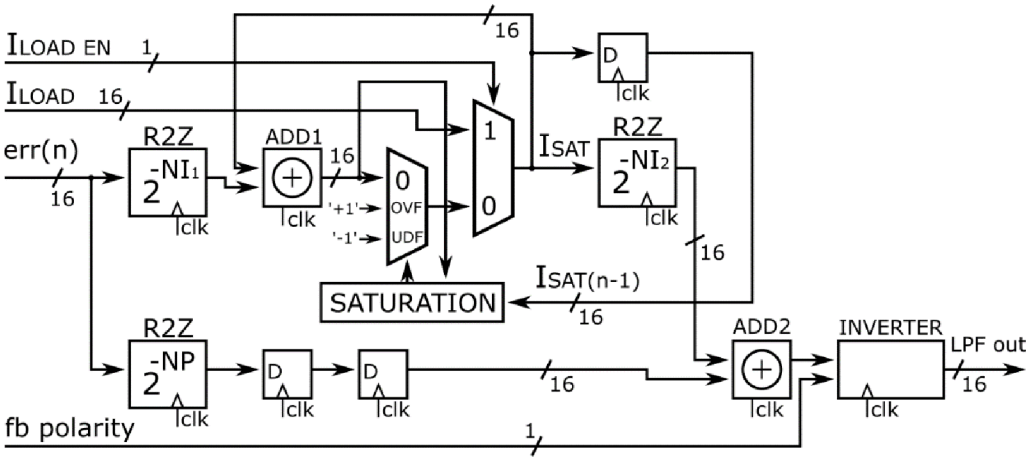


Figure 3-5: Costas loop filter - detailed schematic.

purely combinational (no latency), acting as an IF statement for both overflow and underflow conditions. The overflow condition is passed when $I_{SAT}(n-1) > 0$ and $ADD_1(n) < 0$ and therefore $I_{SAT}(n) = 0x7FFF$ (positive saturation). In case of underflow event $I_{SAT}(n) = 0x8000$ (negative saturation).

There are two different approaches to the situation around DDS and phase increment. The first one deals with the integer version of phase increment $\Delta\theta_{int}$ (having no real physical unit) and the other sees phase increment as $\Delta\theta_{rad}$ (radians). The respective formulas are

$$\Delta\theta_{int} = \frac{f_{DDS}}{\Delta f_{DDS}} = \frac{f_{DDS}}{f_{CLK}} \cdot 2^{16}$$

and

$$\Delta\theta_{rad} = 2\pi \cdot f_{DDS} \cdot \Delta t,$$

where Δt equals sampling (clock) period $T_{clk} = 1/f_{CLK}$. Substituting one equation into another by f_{DDS} yields

$$\Delta\theta_{int} = \Delta\theta_{rad} \cdot \frac{2^{15}}{\pi}. \tag{3.1}$$

Equation (3.1) describes relationship between physical phase increment in radians and the input of the DDS phase accumulator. The coefficients used in MATLAB can be calculated using expressions

$$K_2 = 2^{-NI1-NI2-16} \cdot f_{CLK}, \tag{3.2}$$

$$K_1 = 2^{-NP-16} \cdot f_{CLK}. \tag{3.3}$$

Detailed derivation of the coefficient formulas is in Appendix 3.

Since the I_{SAT} is 16 bits wide, the maximal DDS frequency is limited by the Int16 range $\langle -2^{15}, 2^{15} - 1\rangle$ and the $NI2$ division ratio. The maximal DDS frequency is defined by

$$f_{DDSMAX} = f_{CLK} \cdot 2^{-16} \cdot I_{SATMAX} \cdot 2^{-NI2}, \tag{3.4}$$

where $I_{SATMAX} = 32767$. For example, if $NI2 = 5$, the maximal DDS frequency is 260.4 kHz.

The loop filter has another function of loading a particular value of frequency offset. The I_{LOAD} value (16-bit integer) can be calculated from the desired f_{DDS} value using formula

$$I_{LOAD} = \frac{f_{DDS}}{f_{CLK}} \cdot 2^{NI2+16} \,. \tag{3.5}$$

The loading is done by pulling the $I_{LOAD\ EN}$ signal high for one clock period. This function can be used at the start of the satellite flight when the received signal is weak and the frequency offset is known by the tracking software (Orbitron).

3.2 Symbol timing synchronization loop

3.2.1 Farrow interpolator block

This is the most resource-consuming block of the whole project. (see figure 3-6 for the schematic). The Farrow structure designed as quadratic interpolation filter implements 3 FIR filters described by coefficient vectors $h_A = [0.5, -1, 0.5]$, $h_B = [-1.5, 2, -0.5]$ and $h_C = [1, 0, 0]$. The first simplification concerns h_C which is basically a wired connection. h_A coefficient vector lies within the two's complement range and it can be loaded into the Xilinx FIR compiler. However, h_B exceeds this range and some modifications must be done.

One way is to divide all 3 coefficients by 2. This results in $h_B{}' = [-0.75, 1, -0.25]$, where the middle coefficient still exceeds the range. The most simple solution is to decrease the coefficient by one LSB, which yields

$$h_B{}' = [-0.75, 0.99996946242, -0.25].$$

The output of the 2nd FIR filter must be multiplied by 2 to compensate this modification. Even with the coefficient vectors fixed there are still some overflow/underflow risks. Those can be prevented by adding the R2Z block with division factor 2 to the input of the structure. Another multiplication block must be put to the output of the structure to compensate the total gain.

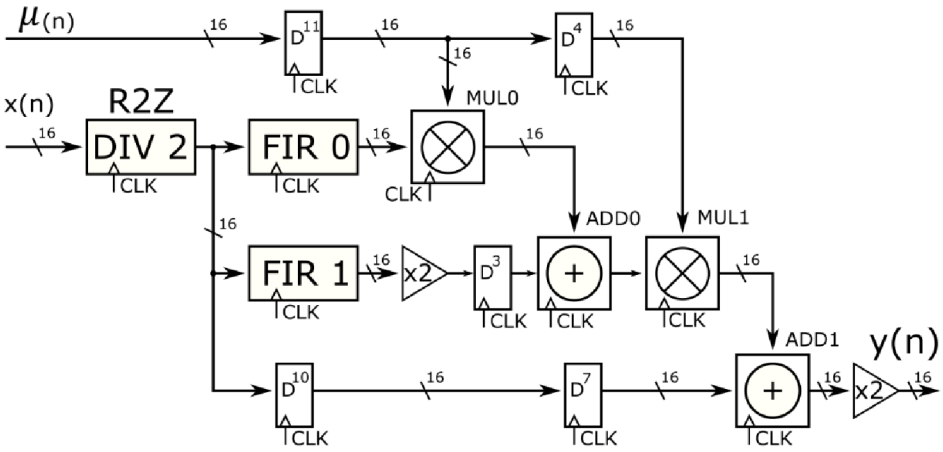


Figure 3-6: Farrow structure - detailed schematic.

The FIR latency (10 cycles), multiplier latency (3 cycles) and adder latency (1 cycle) must be compensated in all parallel signal paths. That is the reason for all the D flip-flops in the structure. The total latency of the implemented Farrow structure is 19 cycles.

3.2.2 Fractional delay accumulator

The function of this block can be described by difference equation

$$\mu_{(n)} = mod_1\left(\mu_{(n-1)} + \mu_{STEP} \cdot (1 + \mu_{OFFSET})\right), \tag{3.6}$$

where $\mu_{(n)}$ is the current value of fractional delay, μ_{STEP} is the fractional delay step and μ_{OFFSET} is the fractional delay offset value. According to chapter 2.4.2, $\mu_{(n)}$ belongs to the range $\langle 0, 1)$ and it is a decreasing sawtooth function of time. For this reason the μ_{STEP} is negative, given by

$$\mu_{STEP} = \frac{N_{OUT}}{N_{IN}} - 1 = -0.1, \tag{3.7}$$

where $N_{OUT} = 2.0\ smp/sym$ and $N_{IN} = \frac{20}{9}\ smp/sym$. Figure 3-7 shows detailed schematic of the fractional delay accumulator. The modulo function is done by ADD1 where the 16-bit feedback signal has the MSB set to zero. The remaining 15 bits (14 to 0) are taken from the ADD1 output (this happens inside the CON block). This modification takes advantage of the underflow event in the adder and the output signal $\mu_{(n)}$ fits into the pre-defined range.

Special attention needs to be paid to μ_{OFFSET} which is driven by STS loop filter operating in another clock domain. A cascade of two 16-bit D flip-flops (running in the new clock domain) works as a simple domain crossing circuit.

The **skip sample** signal (16th bit of the ADD1 output) later serves as the **write enable** for the FIFO used in the Farrow re-sampler block.

3.2.3 Farrow re-sampler block

This block realizes the sample interpolation and clock domain crossing and it consists of Complex Farrow interpolator (one interpolator for each I/Q channel), Fractional delay

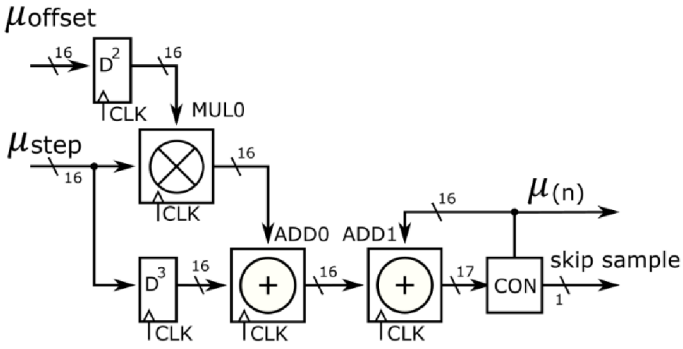


Figure 3-7: Fractional delay accumulator detailed schematic.

accumulator and a FIFO. The block design is depicted in figure 3-8. It is essential that the **skip sample** signal is synchronized with the Farrow interpolator output samples so the correct sample is omitted from writing to FIFO. Maintaining the precise latency is the purpose of the FIFO CTRL block. It consists of two shift registers, one delaying the **skip sample** signal (WR clock domain) and the other delaying the FIFO RD enable signal (RD clock domain). Delaying the FIFO RD enable gives FIFO enough time to get approximately half-full and prevent the FIFO-EMPTY or FIFO-FULL states from occurring.

3.2.4 Gardner TED block

Implementing Gardner TED results in more resource-consuming design than the Costas loop phase detector. It is so due to the multipliers which cannot be replaced with multiplexer-based work-around. Figure 3-9 shows detailed schematic of the Gardner TED. The R2Z block in each channel prevents overflow events in the subtractors SUB I and SUB Q. In this case the output amplitude compensation (for the R2Z blocks) is not necessary since the signal is processed later in the STS loop filter.

As it was announced in chapter 2.4.4, the STS feedback loop operates on half of the clock frequency. For this reason there is the **clk enable** output dividing the clock signal by 2.

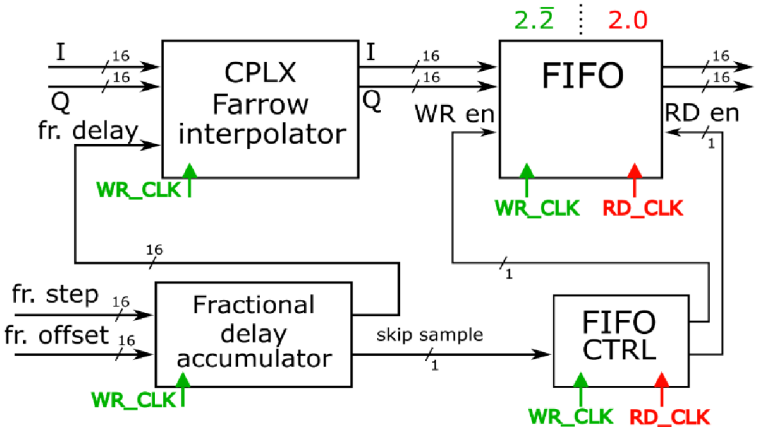


Figure 3-8: Farrow re-sampler block schematic.

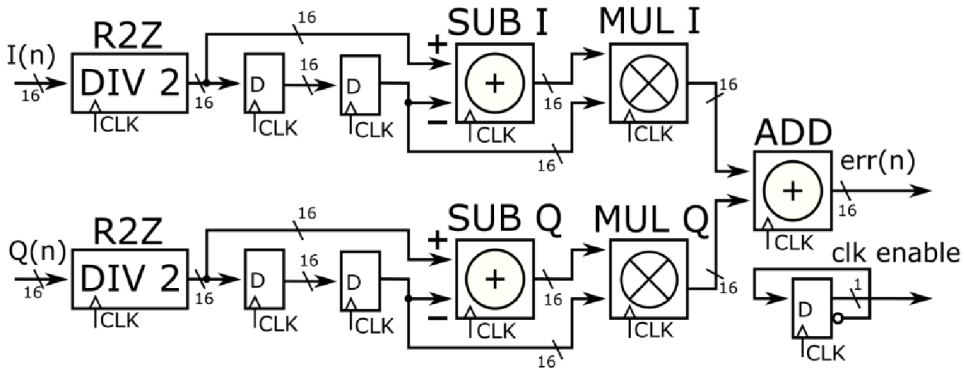


Figure 3-9: Gardner TED detailed schematic.

3.2.5 STS loop filter

The filter consists of integral and proportional paths where each gain is provided by R2Z block (see figure 3-10). Unlike in the Costas loop filter, there is no saturation mechanism for the integral path because its stationary value remains zero (in locked state). The gain bit-shifting ratios equal $NI_{STS} = 9$ and $NP_{STS} = 3$ and they correspond to the values in the MATLAB simulation $KI_{STS} = 2^{-9} = 1/512$ and $KP_{STS} = 2^{-3} = 1/8$. The total latency of the loop filter is 5 cycles (considering the clock enable).

3.3 ASM detection

3.3.1 Symbol detection and NRZ-M decoder

Figure 3-11 (on the left) shows detailed schematic of the symbol detection and NRZ-M decoding, The symbol detection (rounding from 16-bit value to 1-bit) is done by extracting the MSB of the input signal (logic 0 for positive symbols, 1 for negative ones). The exclusive OR (XOR) gate is used for differential decoding.

3.3.2 ASM correlation

Right after the differential decoding both channels enter shift registers shreg I, Q1 and Q2. Shift registers output 32-bit parallel buses which need to be reduced to 16 bits with respect to the ASM correlation word length. That is the purpose of the Buffer interface block described in figure 3-11 (on the right). Figure 3-12 shows detailed schematic of the ASM correlation circuit. The correlation itself is done over 16 bits (samples) of length using the bitwise NXOR block (negated exclusive OR). Each of the ASM words has to be bit-reversed appropriately in order to match the shift register result. The amount of logic ones in the NXOR 16-bit result is proportional to the correlation result.

As trivial as it may seem, counting the logic ones in a binary word regardless their position is not very common logic operation. A special structure of adders was designed for this purpose (see figure 3-13 on the left). This structure sums all its logic inputs and provides an arithmetic result. For a 16-bit input there is a 5-bit output (0 to 16).

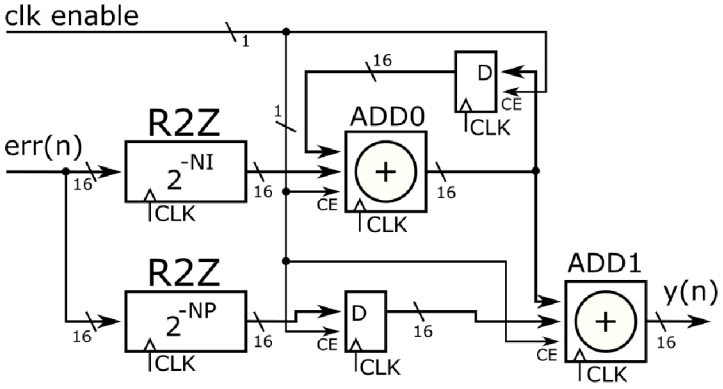


Figure 3-10: STS loop filter - detailed schematic.

The rest of the ASM circuitry (figure 3-13, right) combines two of the 5-bit correlation results (output range is from 0 to 32) and subtracts them from the ASM threshold (set to 29). The sign block extracts the MSB of the result which is later used in the data formatting circuitry (described in chapters 2.5.3 and 2.5.4. Thanks to differential decoding and specific shift register buffering (shreg Q2 is delayed by 2 samples), the ASM correlation peak polarity stays positive for any kind of carrier phase state.

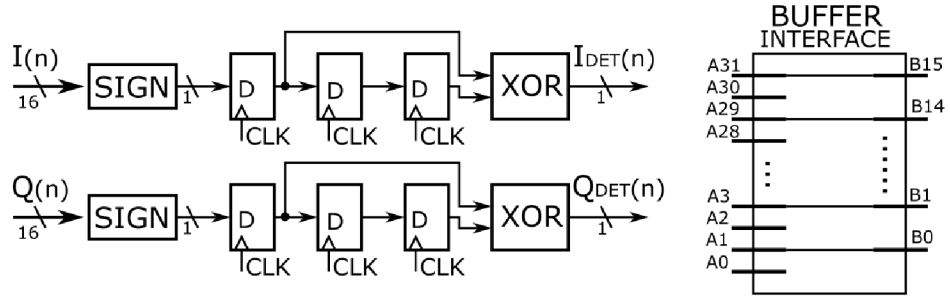


Figure 3-11: Symbol detector and NRZ-M decoder schematic (left) and the buffer interface block description.

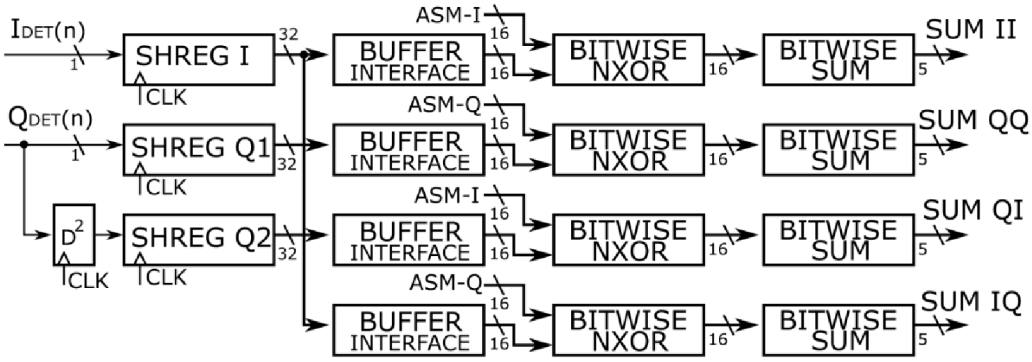


Figure 3-12: ASM correlation circuit schematic.

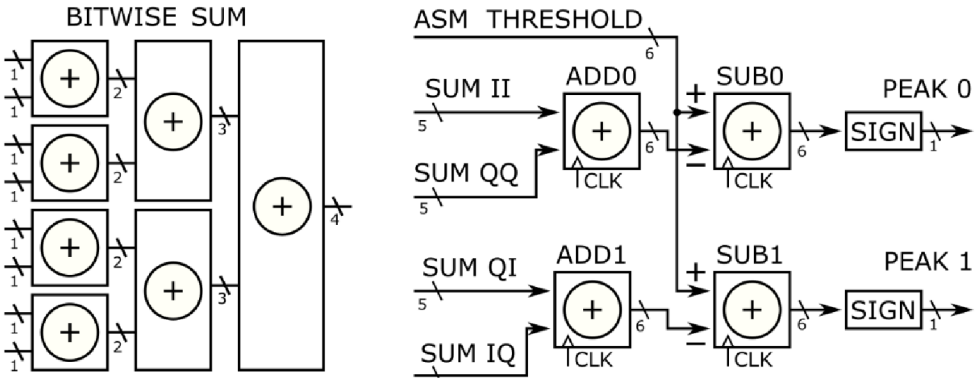


Figure 3-13: The 8-bit version of the bitwise sum block (left) and the ASM correlation threshold conditioning.

3.4 Auxiliary design blocks

Implementing this design into the ZYBO z7 board saved a lot of potential trouble related to designing a custom PCB. Since ZYBO z7 does not directly interface an ADC, another solution had to be found in order to test the receiver in a situation close to real application. The easiest way how to achieve this was to store a raw signal recording in the block RAM (BRAM) and to keep re-playing the recording to the receiver (simulating the real data reception).

Following blocks are necessary for this design to operate in the ZYBO board and help testing the receiver's performance.

3.4.1 Block RAM signal storage

The idea is to store a raw baseband data recording (true satellite signal recording) with the sampling frequency equal to 16.666 MSPS and re-play it to the receiver design blocks periodically. Several aspects have to be considered to accomplish this task.

Since the sampling rate of the raw input signal recording is 50 MSPS (see chapter 2.1), it must first be filtered (CIC or RCF) and decimated to the desired sampling rate. A very important feature of the recording is to be seamless and easily concatenated when moving from its end to the beginning. Any kind of spur or discontinuity would periodically disturb the carrier synchronization loop, devaluating the whole design.

The first criterion for the recording is that it contains integer number of CADU frames so the ASM detection block generates correlation peaks periodically with each CADU. Another criterion is that frequency offset period is an integer fraction of total recording time. This ensures that the offset frequency continuity is maintained. The number of samples in the recording must also be integer.

The CADU frame period is $T_{CADU} = 8192\ bits/15\ Mbps = 546.1\overline{3}\ \mu s$ and the number of samples in it (@16.666 MSPS) is

$$N_{CADU} = 8192 \cdot \frac{16.666}{15} = 8192 \cdot \frac{10}{9} = 9102.\overline{2}\ samples.$$

Using 9 CADU is the shortest recording length which satisfies the condition of integer number of samples. It equals $N_{REC} = 81920$ samples. The total duration of the recording is $T_{REC} = 4.9152\ ms$. The offset frequency criterion is defined as

$$f_{offset} = k/T_{REC}, \text{ where } k = 1, 2, 3 \ldots$$

The offset frequency of the real (measured) satellite signal recording is 48249 Hz. It had to be slightly modified to meet the condition above for the closest value of k. For $k = 237$ the actual offset frequency is $f_{offset} = 48217.77$ Hz

Considering that the input signal consists of two 16-bit complex samples the total memory space required is 2621440 bits (72 BRAMs of size 36 kbit). The BRAM IP core (used as ROM) is addressed by binary counter with pre-defined range from 0 to 81919. The block schematic of the ideal (top) and actual (bottom) FPGA assembly is in figure 3-14.

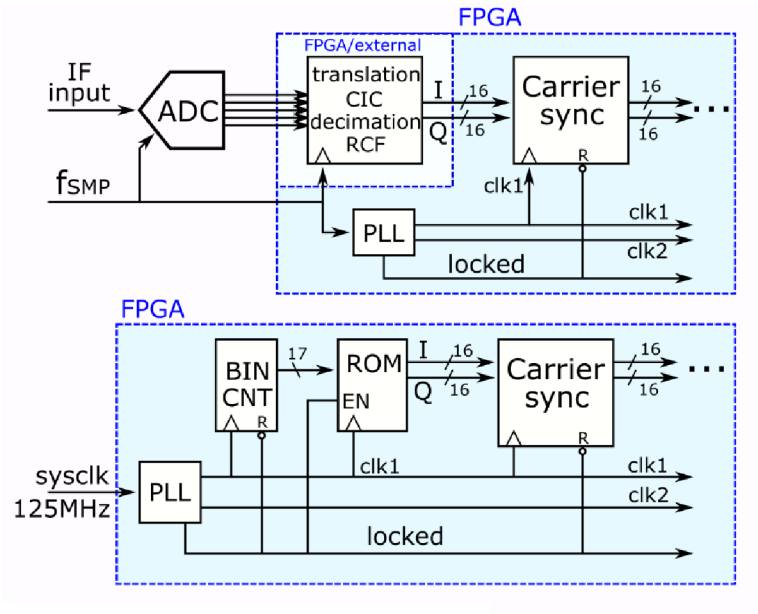


Figure 3-14: BRAM and binary counter schematic.

3.4.2 Clock distribution

The main system clock on the ZYBO z7 board is the 125 MHz clock produced by RTL8111E-VL integrated circuit. The frequency of the reference crystal oscillator is 25 MHz. The XC7Z020 FPGA contains 4 PLLs (Phase-Locked Loop) and 4 MMCM (Mixed-Mode Clock Manager). Each PLL consists of a classic architecture with the phase comparator, loop filter, VCO and frequency dividers (see Figure 3-15).

Two clock domain frequencies desired for this project are 16.666 MHz (write clock, first part of the design) and 15 MHz (read clock, second part). Parameters of the PLL set for this configuration are shown in table 3-1. The PLL locked signal remains low until the PLL outputs are stable enough. This signal is used as the asynchronous reset for majority of blocks in the design.

Table 3-1: PLL parameters.

f_{REF}	125 MHz
f_{FB}	25 MHz
f_{VCO}	900 MHz
DIV-D	5
DIV-M	36
DIV-O1	54
DIV-O2	60

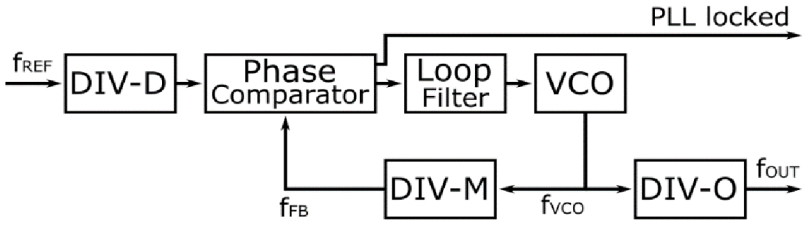


Figure 3-15: PLL architecture schematic.

3.4.3 Serial scope design block

This block was designed purely for verification purposes. It is no straight-forward task to visualize the internal digital signals of the digital receiver, especially if the signals are 16-bit wide and their sampling frequency is 16.666 MHz. The ideal solution is to re-sample the signals with lower frequency (decimate) and put them on the FPGA debug pins (P-mode connectors on the ZYBO board) in the SPI protocol (Serial Peripheral Interface). Such signals can be easily captured, decoded (SPI protocol) and exported to CSV format by logic analyzers.

Figure 3-16 shows the Serial scope block schematic. It consists of 16-bit shift register with parallel loading option and clock-enabled serial output shifting function. The TRIG signal (1 clock cycle impulse) loads the parallel input sample into the shift register and it also sets the R-S flip-flop. Consequently, the clock enable generator (CLK GEN) starts generating clock enable pulses of desired period. Each pulse causes shift register to put a new data sample to the serial output (DATA OUT), starting with the MSB. The pulses are counted by the 4-bit binary counter (CNT) and once it overflows, it resets the R-S flip-flop and terminates the whole cycle. The data clock (DATA CLK) has the same

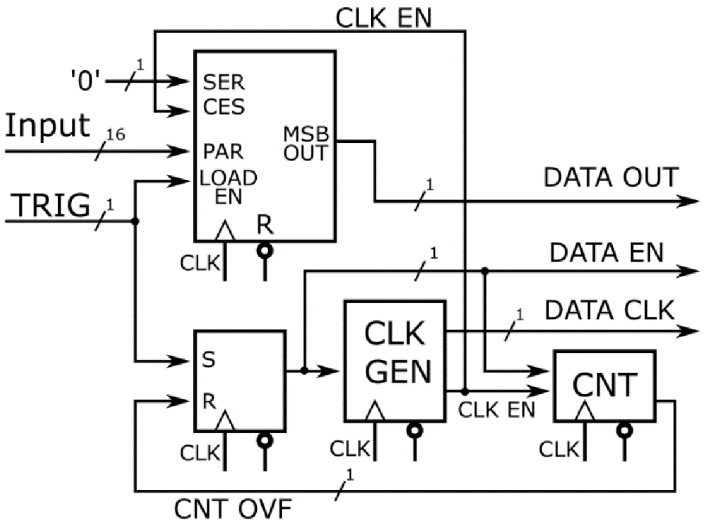


Figure 3-16: Serial scope - schematic.

frequency as the clock enable signal, but the duty cycle is 50%. Example of the signal waveform is in figure 3-17 where the parallel signal value 0xABBA is converted into SPI 16-bit burst. Serial data (MOSI) transitions are synchronized with the falling edge of the data clock (SCLK).

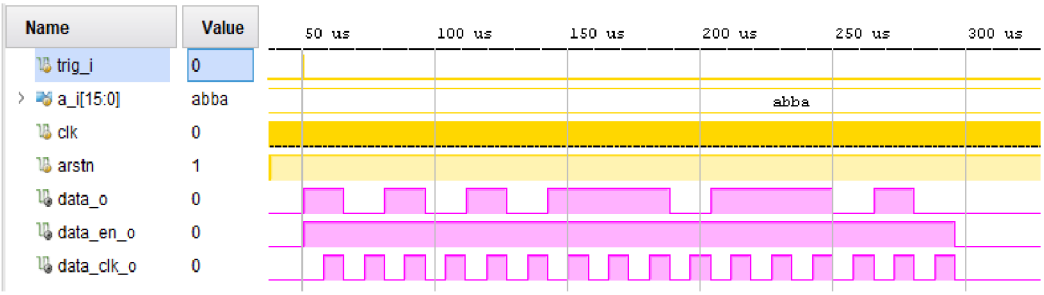


Figure 3-17: Serial scope waveform example (Vivado simulation).

3.4.4 CADU header scope block

This design block is similar to the Serial scope block with one fundamental difference – the input signal is not parallel, it is serial. For quick verification purposes it is handy to check only the first 16 bits of each CADU (the frame header). Those contain the version number (2 bits), SCID (8 bits) and VCID (6 bits). The first two remain the same for the AQUA satellite and the VCID changes according to the current instrument the CADU comes from.

The goal is to extract the 1st 16 bits of data just after the ASM detection peak. For this purpose a structure containing 16-bit shift register was designed (see figure 3-18). The peak detect input comes from the ASM detection block 2 cycles ahead from the actual data on the data input. Both FF1 and FF2 are set and both counters (CNT1 and CNT2)

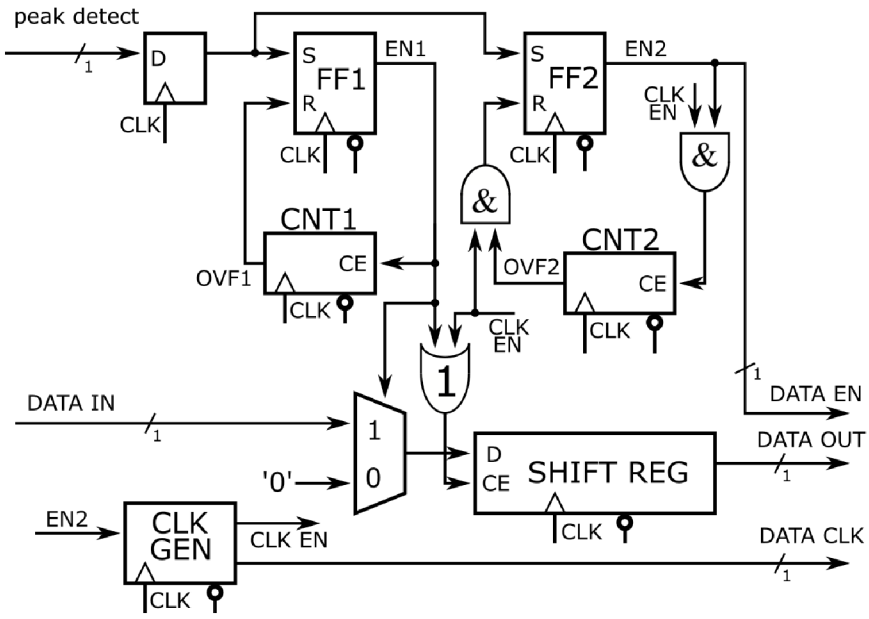


Figure 3-18: CADU header scope block schematic.

start counting. Because CNT1 counts with the 15 MHz clock signal, its overflow event (OVF1) happens after 16 clock cycles. In this moment, the shift register contains the data desired. The OVF1 signal resets FF1 and the shift register starts operating with the clock enable pulses generated in the CLK GEN block. The other counter CNT2 operates at the same speed, counting up to 16. At that time it resets FF2 and the whole cycle is finished, ready for the next ASM peak. The output signal waveforms are identical to the waveforms in figure 3-17 (SPI protocol).

3.5 FPGA Implementation results

Once the design was implemented into the Zynq FPGA, several measurements were made to verify its function. Saleae 8-channel 24 MHz logic analyzer together with the Saleae Logic 1.2.18 software was used for this purpose.

3.5.1 Costas loop performance verification

Measuring the Costas loop filter output signal was done using the Serial scope design block implemented together with a trigger pulse source (all in the 16.666 MHz clock domain). The initial frequency offset (loaded into the Costas loop filter after device reset) was set to 18 kHz while the BRAM sample signal has frequency offset 48217 Hz. Figure 3-19 shows the Costas loop filter output signal comparison between Matlab simulation and the actual FPGA measurement. The loop coefficients in the FPGA were set to $NI1 = 12$, $NI2 = 5$ and $NP = 10$. Coefficients in Matlab were set according to (3.2) and (3.3) as $K_1 = 0.2484$ and $K_2 = 1.94 \cdot 10^{-3}$. The sampling frequency was reduced from 16.666 MHz to 2034 Hz (Serial scope trigger frequency) so the captured signal is affected by aliasing to a certain degree. Even though the Matlab model does not respect many implementation aspects, both performances can be compared by means of the frequency ripple in the locked state (approximately 5 kHz). The time of acquisition, however, cannot be compared with simulation because it heavily depends on initial conditions (non-linear dynamic system).

The FPGA measurement signal (blue curve in figure 3-19) in the locked state appears

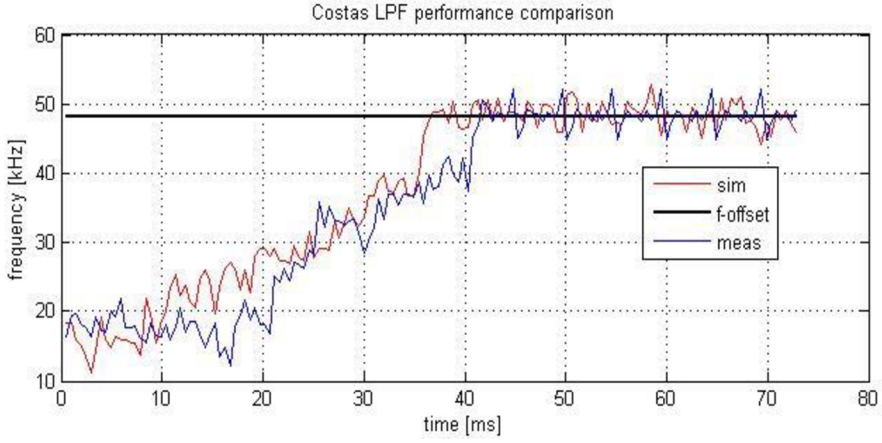


Figure 3-19: Costas loop performance comparison (2034 Hz, original configuration).

to have periodic spikes. It is no coincidence that the period is identical to the BRAM repetition period (4.915 ms). Even modified, the BRAM signal is still not seamless enough and the Costas loop performance is affected. In the real application the loop response will be less noisy. Another performance of the same loop sampled at 8138 Hz is in figure 3-20. The offset frequency ripple remains the same because the coefficients were not changed.

Figure 3-21 illustrates the situation where the loop coefficients are $NI1 = 12$, $NI2 = 5$ and $NP = 9$ (The proportional gain is doubled, sampling stays the same). The loop acquisition time has decreased but the amount of frequency ripple in the locked state has doubled (~12 kHz).

After several re-configurations of the loop coefficients it appears that the original

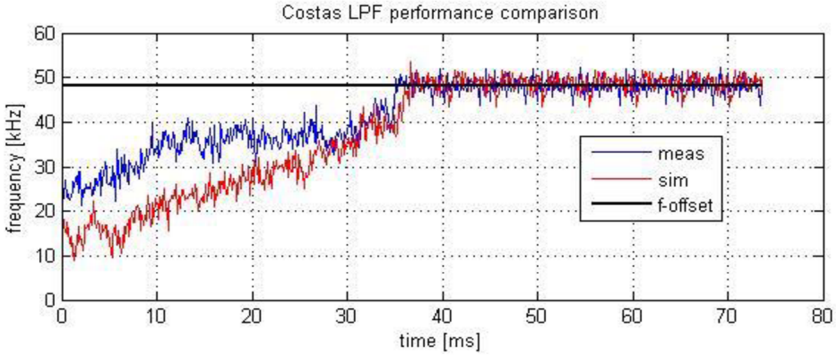


Figure 3-20: Costas loop performance comparison (8138 Hz, original configuration).

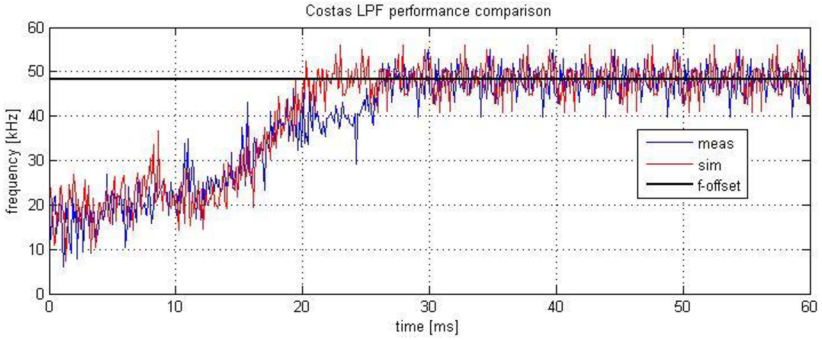


Figure 3-21: Costas loop performance comparison (2034 Hz, NP value doubled).

configuration ($NI1 = 12$, $NI2 = 5$ and $NP = 10$) is optimal.

Just for the illustration, figure 3-22 shows both the Costas loop filter output and the Symbol timing loop filter output for the original configuration. Since those blocks are in cascade, it is no surprise that the STS (bottom waveform) starts operating after the Costas loop gets into locked state (after 60 ms).

3.5.2 Overall design performance

One way to measure the performance of the whole design is to use the CADU header scope block. It outputs data in 3-wire SPI format with clock frequency 58.82 kHz and total burst length 273.2 µs (see figure 3-23). As soon as the Costas loop and the STS are

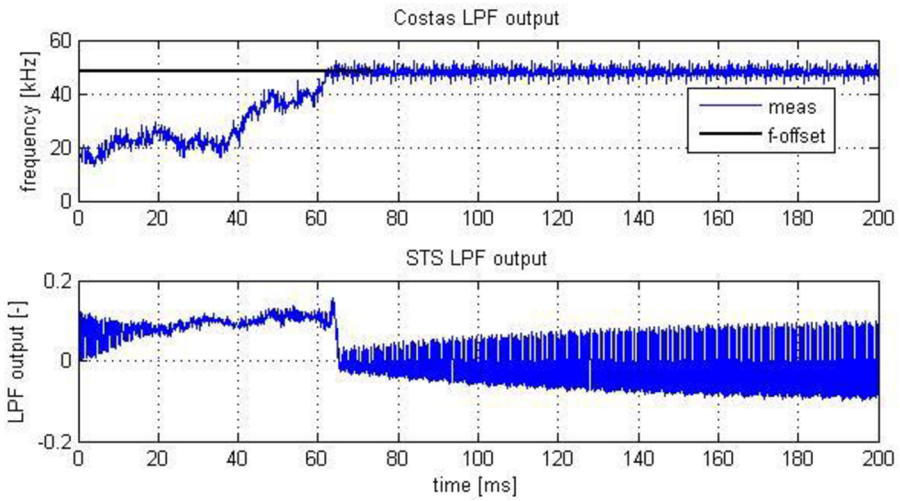


Figure 3-22: Costas LPF and Symbol timing LPF output waveforms.

locked, the CADU scope headers start coming with a fixed period $546.1\overline{3}\,\mu s$. The decimal number 26270 equals 0x669E in hexadecimal base and 16‘b0110011010011110 in 16-bit binary base. The main header components are the version number = 2‘b01, SCID = 8’b10011010 = 0x9A and VCID, which in case of this recording equals 6’b011110 (MODIS instrument) or 6’b100011 (AIRS instrument) [5].

Some of the headers may be corrupted, because the RS decoder has not been implemented yet. However, RS coding is transparent (adding special parity symbols behind the input message; not altering the message itself) and the data are readable in most cases (depends on the degree of corruption).

Figure 3-24 illustrates the overall recovery time (starting from system reset). The CADU headers start coming consistently after 190 ms.

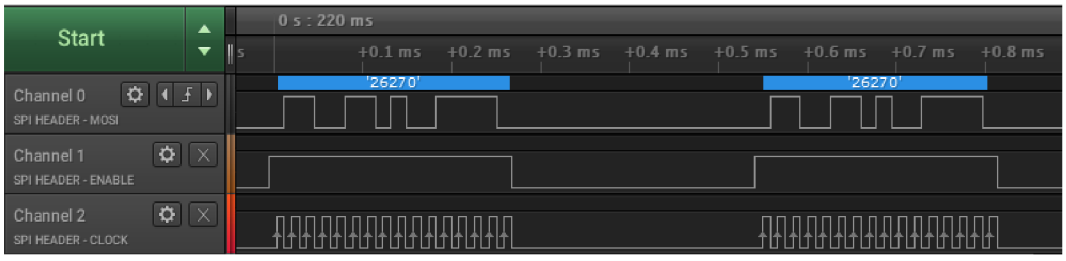


Figure 3-23: CADU headers displayed using the Saleae logic analyzer software.

Figure 3-24: CADU headers illustrating the overall recovery time.

3.5.3 FPGA Resource utilization

Complete resource utilization of the receiver design is shown in table 3-2. The FPGA cell structure view (floor-planning view) is depicted in figure 3-25. Most of the BRAM resources (brown cells) have been used for the recording storage. It is no surprise that from the main design blocks the CS loop and the STS loop are the biggest ones.

Considering only one quarter of total resources used, the next iteration may increase the sampling rate resulting in more robust performance of the CS loop and STS loop.

Table 3-2: FPGA resource utilization.

Resource	Utilization	Available	Utilization %
LUT	1269	53200	2.39
LUTRAM	91	17400	0.52
FF	2860	106400	2.69
BRAM	78	140	55.71
DSP	24	220	10.91
IO	17	125	13.60
BUFG	3	32	9.38
PLL	1	4	25.00

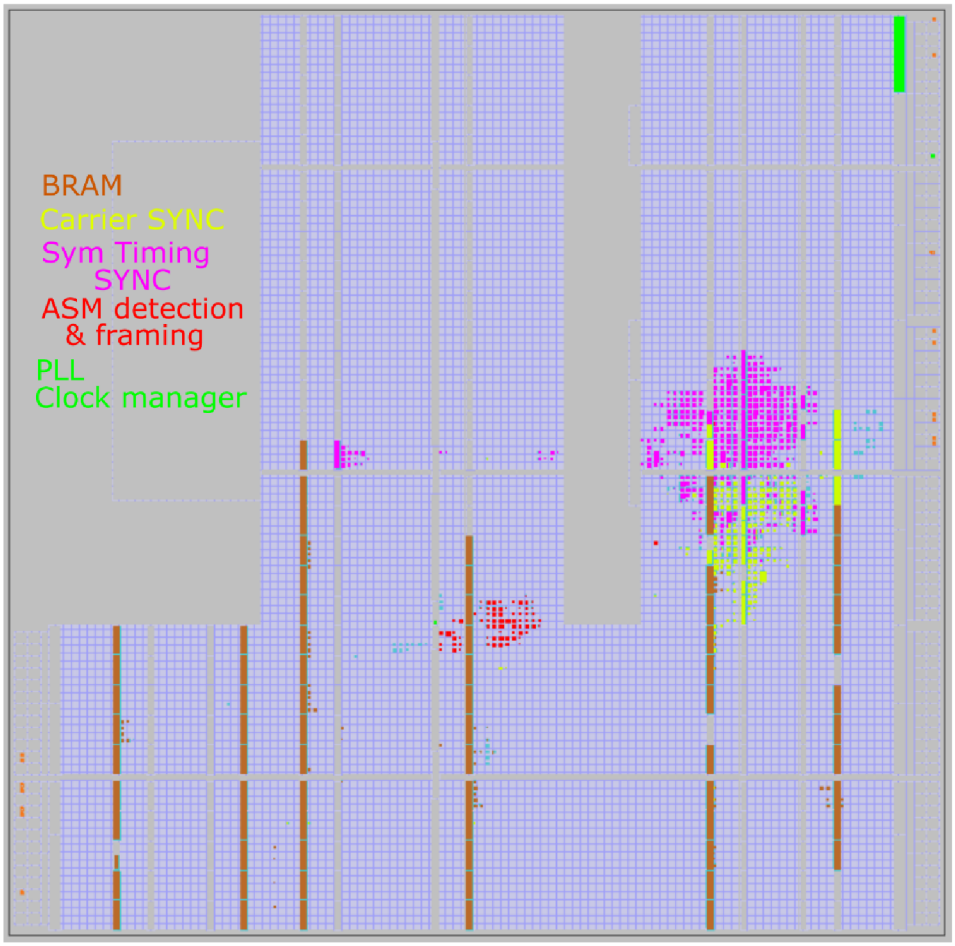


Figure 3-25: FPGA floor-planning view after implementation.

CONCLUSION

A brief study of the available X-band broadcasting spacecrafts has been made In chapter 1, highlighting the METOP, EOS-AM-1 (Terra) and EOS-PM-1 (Aqua) satellites. The Aqua satellite has been chosen as the target of the digital receiver design due to the highest amount of information available.

In Chapter 2 the digital receiver functionality has been successfully simulated in MATLAB. The input signal has been modelled using limited information from [3][4][5] and in order to reduce the FPGA implementation resources, the symbol sampling ratio was chosen to be 20/9 samples per symbol (16.666 MSPS). The idea was to reduce the 1st Nyquist zone to cover the main spectral lobe of the O-QPSK signal (15 MHz). Another reason for the non-integer ratio was cooperation with an SDR which had a fixed sampling rate of 50 MSPS (3rd multiple of 16.666 MSPS).

Due to the obsolete information contained in [3] it was assumed (mistakenly) that the actual link encoding scheme is supposed to be NRZ-L. However, the real signal measurements confirmed NRZ-M as the true link encoding scheme.

A simple case study about the maximal expected Doppler frequency shift has been done with the result $\pm 183.5\ kHz$.

A sophisticated method of the signal-aided frequency offset compensation using the Ballanced Quadricorrelator [6][7] has been simulated and its functionality has been proven. However, a limitation in minimal frequency offset residual has been found and it has been described by the empirical formula (1). Another empirical formulas (2)(3) describe the process of the offset frequency compensation and those can be used to calculate the total BQ process duration. The BQ mechanism later showed up not to be essential for the offset frequency compensation since the carrier synchronization block combined with the initial frequency preset were proved to be sufficient.

The carrier phase synchronization has been maintained by the O-QPSK-modified Costas loop. It was found that the Costas loop does not need any pulse shaping filtration (after the CMPY block) because the spectrum of the useful signal occupies all the 1st Nyquist zone (any multiplication products get aliased into the useful spectrum). Because of the input signal fractional symbol sampling rate (20/9 samples per symbol) the fractional delay FIR filter implementing the Lagrange interpolation (see Appendix 1) has been used to delay the I-channel signal by half of the symbol period. The feedback loop behavior has been analyzed and presence of the beat frequency and the pull-in effect have been elaborated. The loop tracking speed has been compared to the Doppler frequency shift steepness related to the SC flight speed and it was found satisfactory. The influence of the O-QPSK carrier phase ambiguity has been discussed, later to be resolved by the ASM detection block.

The symbol timing synchronization block was designed to provide the symbol re-sampling from fractional (20/9) to integer (2.0) symbol sampling ratio. The Farrow interpolation structure has been used for this purpose implementing the quadratic Lagrange polynomial interpolation. Gardner-type TED [] has been used to ensure the optimal symbol sampling moments and the STS loop filter operating at half of the sampling rate has been used to close the STS loop.

The final block of the receiver chain is the ASM detection block. It is based on the

complex correlation with the ASM and together with NRZ-M decoder it helps to resolve the carrier phase ambiguity (see Appendix 2) and to start the PRBS decoding and data framing process.

Chapter 3 describes most of the design blocks in detail so they can be implemented into Xilinx ZYBO z7 development board. The schematics contain detailed information about signal bit-widths and latencies which are important especially for the closed-loop digital systems. First, some of the elementary DSP building blocks and IP cores are described (Fixed-point arithmetics, Complex multiplier, DDS…). In order to match this design with the Matlab simulations, the Costas loop coefficient formulas (see Appendix 3) have been defined.

The Farrow structure implementation focused on proving the concept of reducing the number of multipliers. However, the FIR filters were designed as IP cores, downgrading the total resource cost. In the next iteration of this project the FIR filters used in the Farrow structure should be implemented more economically (using adders and bit-shift operations).

Since the internal signals are 16 bits wide and run at 16.666 MHz or 15 MHz, some kind of auxiliary design blocks had to be designed to act as special kinds of logic analyzers. These blocks probe the carrier synchronization loop filter (its input and output) and put the output serial signal to the FPGA debug pins where they can be captured by the conventional logic analyzers.

Each design block has been tested and verified with Matlab simulations before it had been combined with others. In order to test the overall design performance and preserve the simplicity, a compromise had to be made concerning the source of the satellite signal. A recording, containing 81920 samples of the SDR baseband signal had to be stored into the Block RAMs and re-played continuously, simulating the real signal reception. The concatenation of the recording is not completely seamless, causing periodic spikes in the Costas loop filter output.

Verifying the overall function is done by monitoring the Costas loop filter signals (detecting the locked state) and by extracting the first 16 bits of each CADU header. The implementation results show that majority of the Zynq FPGA structure remained unused. More importantly, the implementation results made it clear about what kind of FPGA (number of LUTs, BRAMs, DSP blocks…) should be used in the next iteration of this project.

REFERENCES

[1] EUMETSAT (2012). MISSION ANALYSIS OF METOP-A END-OF-LIFE OPERATIONS. [online] Darmstadt, Germany. Available at: http://issfd.org/ISSFD_2014/ISSFD24_Paper_S17-3_Righetti.pdf [Accessed 3 Dec. 2018].

[2] NASA Goddard Space Flight Center (1998). Direct Access System User's Guide for the EOS–AM Spacecraft. [online] Greenbelt, Maryland. Available at: https://directreadout.sci.gsfc.nasa.gov/documents/satellite_gen/DAS_UG.pdf [Accessed 3 Dec. 2018].

[3] Wende, C. and Dodge, J. (1999). The Direct Broadcast Service on Office of Earth Science Spacecraft. [online] Available at: https://directreadout.sci.gsfc.nasa.gov/links/rsd_eosdb/PDF/DBUWPfall982.pdf [Accessed 3 Dec. 2018].

[4] NASA Goddard Space Flight Center (2002). INTERFACE DESCRIPTION DOCUMENT FOR EOS AQUA X-BAND DIRECT BROADCAST. [online] Greenbelt, Maryland. Available at: https://directreadout.sci.gsfc.nasa.gov/links/rsd_eosdb/PDF/IDD_Aqua-DB.pdf [Accessed 3 Dec. 2018].

[5] NASA Goddard Space Flight Center (2002). EOS PM-1 SPACECRAFT TO EOS GROUND SYSTEM INTERFACE CONTROL DOCUMENT. [online] Greenbelt, Maryland. Available at: https://directreadout.sci.gsfc.nasa.gov/links/rsd_eosdb/PDF/ICD_Space_Ground_Aqua.pdf [Accessed 3 Dec. 2018].

[6] A. N. D'Andrea and U. Mengali, "Performance of a quadricorrelator driven by modulated signals," IEEE Transactions on Communications, vol. 38, no. 11, pp. 1952–1957, 1990.

[7] A. N. D'Andrea and U. Mengali, "Design of quadricorrelators for automatic frequency control systems," in IEEE Transactions on Communications, vol. 41, no. 6, pp. 988-997, June 1993.

[8] Simon, M. (1998). Carrier Synchronization of Offset Quadrature Phase-Shift Keying. [online] Jet Propulsion Laboratory, California Institute of Technology. Available at: https://pdfs.semanticscholar.org/5985/aec8cdd7f93a285f01d0d8d9a71be7b0bf25.pdf [Accessed 3 Dec. 2018].

[9] M. Gardner, Floyd. (2005). Phaselock Techniques: Third Edition. Phaselock Techniques: Third Edition. 70-74. 10.1002/0471732699.

[10] A. C. Kelly, P. L. Coronado, W. F. Case and A. G. Franklin, "Terra, Aqua, and Aura Direct Broadcast-providing Earth Science data for real-time applications," 2010 IEEE International Geoscience and Remote Sensing Symposium, Honolulu, HI, 2010, pp. 1438-1441.

[11] A. Savtchenko, R. Kummerer, P. Smith, S. Kempler, and G. Leptoukh, "A-Train data depot - bringing Atmospheric measurements together," in 2007 IEEE International Geoscience and Remote Sensing Symposium, 2007.

[12] H. Li, G. Torfs, T. Kazaz, J. Bauwelinck and P. Demeester, "Farrow structured variable fractional delay lagrange filters with improved midpoint response," 2017 40th International Conference on Telecommunications and Signal Processing (TSP), Barcelona, 2017, pp. 506-509.

[13] Rice, M. (2009). Digital Communications: A Discrete-Time Approach. Upper Saddle River: Pearson Prentice Hall, pp.462-471.

[14] Rice, M. (2009). Digital Communications: A Discrete-Time Approach. Upper Saddle River: Pearson Prentice Hall, pp.382-391.

[15] M. Gardner, Floyd. (2005). Phaselock Techniques: Third Edition. Phaselock Techniques: Third Edition. 189-193. 10.1002/0471732699.

[16] Mengali, U. and D'Andrea, A. (1997). Synchronization techniques for digital receivers. New York: Plenum, pp.362-371.

[17] Mengali, U. and D'Andrea, A. (1997). Synchronization techniques for digital receivers. New York: Plenum, pp.393-395.

LIST OF ABBREVIATIONS AND SYMBOLS

AIRS	*Atmospheric Infrared Sounder*
ASM	*Attached Synchronization Marker*
BRAM	*Block Random Access Memory*
BQ	*Balanced Quadricorrelator*
CADU	*Channel Access Data Unit*
CCSDS	*Consultative Committee for Space Data Systems*
CE	*Complex Envelope*
CIC	*Cascaded Integrator-Comb*
CMPY	*Complex Multiplication*
CS	*Carrier Synchronization*
dB	*decibel*
DDS	*Direct Digital Synthesis*
EOS	*Earth Observing System*
EPGS	*EOS Polar Ground Stations*
FIR	*Finite Impulse Response*
GS	*Ground Station*
LO	*Local Oscillator*
LPF	*Low Pass Filter*
LUT	*Look-Up Table*
MA	*Moving Average*
MODIS	*Moderate Resolution Imaging Spectroradiometer*
NCO	*Numerically Controlled Oscillator*
PD	*Phase Detector*
PLL	*Phase Locked Loop*
PRBS	*Pseudo-Random Bit Sequence*
RCF	*Raised Cosine Filter*
RS	*Reed-Solomon*
RTL	*Register-Transfer Level*
SC	*Spacecraft*
SCID	*Spacecraft Identifier*
SPI	*Serial Peripheral Interface*

SRRC	*Square-Root Raised Cosine*
STS	*Symbol Timing Synchronization*
TED	*Timing Error Detector*
VCID	*Virtual Channel Identifier*
β	*Roll-off factor of the Raised Cosine filter*
$\hat{x}$	*complex number*

APPENDIX 1

The design process of the fractional delay FIR filter using the 2nd order polynomial interpolation begins with substituting

$$\mu T = x_2 - x, T = x_2 - x_1 = x_1 - x_0$$

in the Lagrange basis polynomials, where T is the sampling period which cancels out during the simplifying process. The Lagrange basis polynomials are given by

$$l_0 = \frac{(x_2-x)\cdot(x_1-x)}{(x_2-x_0)\cdot(x_1-x_0)} = \frac{\mu T\cdot(\mu T-T)}{2T\cdot T} = \frac{\mu\cdot(\mu-1)}{2},$$

$$l_1 = \frac{(x_2-x)\cdot(x_0-x)}{(x_2-x_1)\cdot(x_0-x_1)} = \frac{\mu T\cdot(\mu T-2T)}{T\cdot(-T)} = \mu\cdot(2-\mu),$$

$$l_2 = \frac{(x_1-x)\cdot(x_0-x)}{(x_1-x_2)\cdot(x_0-x_2)} = \frac{(\mu T-2T)\cdot(\mu T-T)}{(-2T)\cdot(-T)} = \frac{(\mu-2)\cdot(\mu-1)}{2}.$$

The FIR filter impulse response is equivalent to the Lagrange basis polynomials for given fractional delay μ.

$$h_{FD}(n) = l_2\cdot x(n) + l_1\cdot x(n-1) + l_0\cdot x(n-2).$$

APPENDIX 2

In order to fully understand the phase ambiguity resolution method, it must first be explained what does carrier phase offset do to the complex baseband signal. Assuming the Costas loop is in locked state and the NCO frequency matches the carrier frequency of the received signal. According to the S-curve there are 4 equally probable stable points for the phase φ to settle in: 0°, 90°, 180° and 270°. The Costas loop output baseband signal $\hat{c}_{cost}$ is related to the signal at the TX side $\hat{c}_{TX}$ according to

$$\hat{c}_{cost} = \hat{c}_{TX} \cdot e^{j\varphi},$$

where $\hat{c}_{TX} = I_{TX} + jQ_{TX}$ and $\hat{c}_{cost} = I_{cost} + jQ_{cost}$. Table A2-1 illustrates the influence of the phase on the I and Q parts of the input signal.

Table A2-1: Carrier phase offset influence.

Phase	I_{cost}	Q_{cost}
0°	I_{TX}	Q_{TX}
90°	$-Q_{TX}$	I_{TX}
180°	$-I_{TX}$	$-Q_{TX}$
270°	Q_{TX}	$-I_{TX}$

Apparently, phases 0° and 180° are different only in the polarity (the same goes for 90° and 270°). At this stage the signal is still NRZ-M encoded and changing the polarity of both channels (I and/or Q) will not manifest after the NRZ-M decoding. Table A2-2 shows the situation for one of the channels (2.0 samples per symbol) where NRZ-M decoding process is done for both polarities of the signal.

Table A2-2: NRZ-M decoding process for both signal polarities.

I(n)	-1	-1	1	1	-1	1	-1	1	-1
I(n-2)	x	x	-1	-1	1	1	-1	1	-1
$Y_1(n)$	x	x	-1	-1	-1	1	1	1	1
-I(n)	1	1	-1	-1	1	-1	1	-1	1
-I(n-2)	x	x	1	1	-1	-1	1	-1	1
$Y_2(n)$	x	x	-1	-1	-1	1	1	1	1

Both results are identical, proving that NRZ-M encoding devaluates the signal polarity as a factor influencing the ASM correlation.

Next, the 2-sample offset in the I-channel during the correlation buffering (figure 2-32) has to be explained. This is due to the O-QPSK modulation technique. The Q-channel is first delayed by ½ of the symbol on the TX side. It is then compensated by delaying the I-channel by ½ of the symbol on the RX side. However, if the Costas loop gets locked with 90° phase offset, the input I/Q channels are swapped and the Q channel gets delayed by ½ of the symbol once more, making it 1-symbol delay (2 samples).

APPENDIX 3

First, the relationship between the integration coefficient K_2 (Matlab) and $NI1$ and $NI2$ is described. In the process of numeric integration, following statements apply:

- FPGA $I_{SAT2} - I_{SAT1} = \Delta I_{SAT} = err \cdot 2^{-NI1}$ (A2.1)
- MATLAB $I_2 - I_1 = \Delta I = err \cdot K_2$ (A2.2)

where index 2 symbolizes the most recent value and index 1 the value from the previous cycle. The err signal has the same values (and range) for both FPGA and MATLAB. Phase increment at the input of the DDS equals

$$\Delta\theta_{int} = I_{SAT2} \cdot 2^{-NI2}. \quad \text{(A2.3)}$$

In MATLAB the phase increment is given by

$$\Delta\theta_{rad} = 2\pi \cdot I_2 / f_{CLK}. \quad \text{(A2.4)}$$

Applying the mutual phase increment formula

$$\Delta\theta_{int} = \Delta\theta_{rad} \cdot \frac{2^{15}}{\pi} \quad \text{(A2.5)}$$

and substituting $\Delta\theta_{rad}$ from the (A2.4) yields

$$\Delta\theta_{int} = 2^{16} \cdot I_2 / f_{CLK}. \quad \text{(A2.6)}$$

Placing right-hand sides of equations (A2.3) and (A2.6) into equivalence yields

$$I_{SAT2} \cdot 2^{-NI2} = 2^{16} \cdot I_2 / f_{CLK} \quad \text{(A2.7)}$$

Substituting variables I_{SAT2} and I_2 for their differentials modifies the above equation and makes the coefficient relationship clear. Using (A2.1) and (A2.2) in (A2.7) yields

$$err \cdot 2^{-NI1} \cdot 2^{-NI2} = 2^{16} \cdot err \cdot K_2 / f_{CLK},$$

and after simplification the result is

$$K_2 = 2^{-NI1-NI2-16} \cdot f_{CLK}.$$

The relationship between the proportional coefficient K_1 and NP is described below. Similar to (A2.3) there is the proportional path contribution to the phase increment at the DDS input

$$\Delta\theta_{int} = err \cdot 2^{-NP}. \quad \text{(A2.8)}$$

In MATLAB the proportional path phase increment in radians equals

$$\Delta\theta_{rad} = 2\pi \cdot K_1 \cdot err / f_{CLK} \quad \text{(A2.9)}$$

Substituting $\Delta\theta_{rad}$ into (A2.5) yields

$$\Delta\theta_{int} = 2^{16} \cdot K_1 \cdot err / f_{CLK}. \quad \text{(A2.10)}$$

Placing the right-hand sides of (A2.8) and (A2.10) into equivalence yields the final expression

$$K_1 = 2^{-NP-16} \cdot f_{CLK}.$$