Real-Time Estimates of Differential Signal Phase for Spaceborne Systems Using FPGAs

Vishwas Vijayendra, Paul Siqueira, Harikrishnan Chandrikakutty,
Akilesh Krishnamurthy, and Russell Tessier
Department of Electrical and Computer Engineering, University of Massachusetts, Amherst

Abstract

The high cost and inaccessibility of spaceborne systems necessitates the adaptation of computing resources based on changes in environmental conditions. In this paper, a new, adaptive phase difference detection circuit that can be implemented in reconfigurable hardware is described. This circuit measures both phase changes of a single signal over time and phase drifts between two associated signals. This information can be used for anomaly detection for a variety of spaceborne applications, eliminating the need to transmit data for further terrestrial processing. We demonstrate the benefit of this adaptive circuit using a new state-of-the-art data acquisition system developed for the NASA Surface Water Ocean Topography (SWOT) project. Phase calculations with an error of less than 2% are determined using our adaptive approach.

1. Introduction

In the upcoming decade, a number of remote sensing satellite systems planned by NASA will rely on the use of microwaves (300 MHz to 30 GHz) to determine physical characteristics of the Earth from the vantage point of space. The National Research Council's decadal survey [1] specifically highlighted three projects: the Tier-1 L-band (1.2 GHz) DESDynI mission, which is implementing a spaceborne synthetic aperture radar, the Tier-2 Ka-band (35.75 GHz) SWOT mission which is implementing a cross-track interferometer for measuring surface water and ocean topography, and the Tier-3 GRACE II follow-on mission, which will measure minute changes in Earth's gravitational field. While each of these satellite missions relies on the application of microwaves in a different way, a fundamental characteristic of the microwave signals for all three projects is the signal's phase, which is associated with the round-trip distance between the signal's source and destination.

For the DESDynI mission (Deformation, Ecosystem Science and Dynamics of Ice), signal phase is used both to form a synthetic aperture which focuses observations into high resolution imagery of the Earth's surface and to analyze a signal's electric field polarization upon return. Polarization is a particularly important metric used for inferring structural characteristics from the scattering target. The improved measurement of phase for a mission such as DESDynI would allow for improved focusing of the end-product data and better discrimination of the received signal polarization [2].

For the SWOT (Surface Water and Ocean Topography) mission, a common signal is reflected from the Earth's surface and received by two antenna panels (Figure 1) co-mounted on a satellite that are separated by a fixed distance called the baseline (nominally 10m). The differential phase of the received signal by the two ends of the baseline is used to infer the angle of arrival of the reflected signal. Since the viewing geometry is known, it is possible to determine the topography of the ocean and inland waters to a high degree of accuracy [3]. The precision to which this phase difference can be measured directly affects the intended science product of height. Hence, an improved phase measurement capability can be used to improve overall system performance, or can be traded against other system parameters such as the baseline length, to reduce the size of the spacecraft structure and dramatically impact the overall system cost.

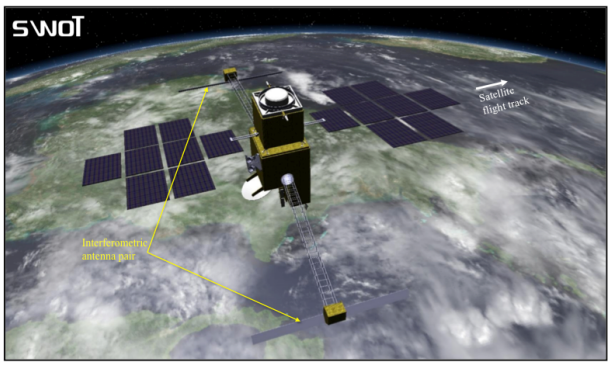

Figure 1. Artistic concept of the SWOT satellite mission. The two antennas that make up the interferometric pair are highlighted. The phase difference of signals measured between these antennas is used to determine the topography of the ocean, rivers, and inland bodies of water.

For the GRACE-II (Gravity Recovery and Climate Experiment) follow-on mission, which is still in the planning stages, either a microwave or a laser signal will be used to monitor the distance between two co-orbiting satellites. Because satellite orbits are affected by the earth's gravitational field, a mapping of this changing

distance as a function of orbit location can be translated into a map of the gravitational field and used to monitor changes in the field. The very successful GRACE mission launched in March of 2002 [4], which uses 24 GHz and 32 GHz cross-linked signals to accurately monitor the distance between the two satellites, has successfully employed the approach. While there are many factors that ultimately contribute to the GRACE mission's acuity in measuring gravitational changes, the measurement accuracy of microwave phase has a direct impact on the final end product due to its fundamental role in determining the distance between the two satellite platforms.

In all three systems, it is clear that signal phase measurements play an important role in fundamental satellite system operation. As a result, it is important to measure and monitor a signal's phase from a measurement point of view and in terms of its use as a characterization of the overall system's performance [5]. Performance characterization may occur in the final stages of the spacecraft assembly or as part of routine on-orbit operations. A typical phase characterization may involve the injection of a known signal into the nominal science data path, the recording of the characteristics of the known signal, and a comparison against a standard. Overall, this process can be best described as system calibration.

When performed prior to launch, this calibration is used to assure that the overall satellite operation is able to perform up to specification. In this pre-launch configuration, access to key internal workings of the satellite becomes more challenging as the system achieves higher levels of integration because the configuration more closely mimics that which will ultimately deployed in space. During pre-launch, access the large volumes of data and intermediate products necessary to accurately measure phase becomes difficult and it is more desirable to measure the science end-product directly on the satellite system.

In the post-launch configuration it is often necessary to monitor overall system performance. Due to data rate limitations it is desirable to perform corrections as necessary on-board to obviate the need for transmitting calibration data intermediate products to the ground. In this instance, a health-monitoring system is implemented on the satellite. This unit provides system-level metrics of overall system health and, if necessary, records and transmits intermediate products that can be used to perform detailed system diagnostics.

Due to the importance of phase calculations for NASA missions and the data rate and access limitations apparent in present and planned satellite systems, it is desirable to implement low overhead phase stability calculations in a satellite's signal path. This information can then be used to monitor the overall health of the satellite in real time. Increasingly, spaceborne data systems are incorporating significant computational power directly into the signal path in a process that improves the efficiency of the overall satellite system. This implementation affords the opportunity to continuously monitor system health at time scales commensurate with those of the raw science data, which may be on the order of milliseconds to seconds.

In this paper, we present an adaptive implementation of a hardware circuit which allows for the continuous monitoring of phase. This implementation is migrated from a phase monitoring software algorithm [5] that has previously been shown to provide high accuracy. The new hardware circuit has been directly implemented into a field-programmable gate array (FPGA) that is interfaced to two high-speed analog-to-digital converters (ADCs). The FPGA-based system has been designed to perform computation in the signal path for the SWOT mission. By measuring and monitoring the differential phase of a known signal fed into the two analog-to-digital converters, a low-level check can be performed on the science data integrity on a pulse-to-pulse basis. The results are subsequently passed down and incorporated into a standard data telemetry data stream. On-board FPGA monitoring of these results can be used as a "watch dog" to flag problematic data and to record intermediate products that can be saved and sent to the ground for further analysis. By utilizing an adaptive component such as an FPGA for this process, the system can be configured over the lifetime of the satellite to alter the threshold at which detailed and data intensive analysis takes place. As a result, the platform provides an efficient and insightful approach for monitoring overall satellite system performance.

Through experimentation, we show that the FPGA circuit required to perform phase monitoring easily fits within FPGA resources (~20,000 logic blocks) and can achieve high precision (within 2% accuracy). Our approach is demonstrated in real-time using a signal generator and an FPGA-based system.

The rest of this paper is organized as follows. Section 2 provides background on the need for real-time phase detection and its possible use in the SWOT system. In Section 3, the details of our FPGA phase detection circuit are described in detail. The experimental approach used to generate results is presented in Section 4 while the results themselves are provided in Section 5. Section 6 concludes the paper and provides directions for future work.

2. Background

This work was motivated by our effort to construct an integrated Ka-band (35 GHz) two-channel receiver with a digital sampling system (Figure 2) for the SWOT mission. The system utilizes two 3 GSamp/sec analog-to-digital converters and a Xilinx Virtex-4 FPGA to perform a final stage of downconversion of a 62 MHz signal from 1.2 GHz to baseband [6]. The construction and integration of

the RF and digital components for the receiver is meant to provide a mechanism to digitally compensate for thermally-induced phase and amplitude variations in the signal chain by the RF portion of the receiver. Because of the high frequency at the input to the receive chain (35.75 GHz, or equivalently, a wavelength of 8.4 mm), small differences in temperature between the two receive channels will manifest themselves as changes in the electrical path length. As a result, a differential phase not associated with the measurement of topographic height (e.g. Figure 3), will be recorded potentially inducing a large error source in the overall system.

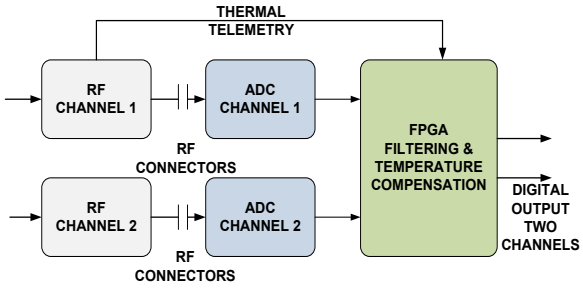

Figure 2. A block diagram of the planned SWOT RF and digital receive chain. Analog downconversion from 35 GHz to 1.2 GHz takes place prior to sampling by two 3 GSamp/sec A/D converters, which is then read in by a Xilinx Virtex-4 FPGA. Further downconversion (digital) and signal compensation meant to account for thermal variations in the RF portion of the receiver is performed in the FPGA.

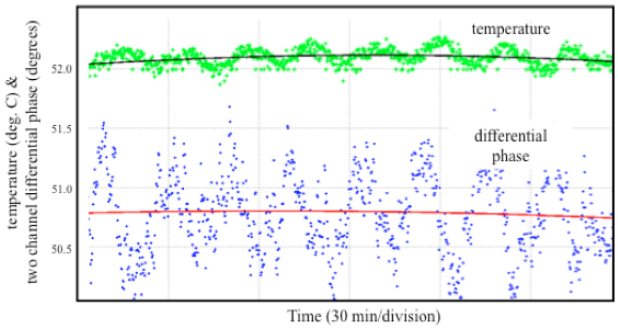

Figure 3. Illustration of the relationship between receiver thermal temperature (green) and the differential phase of two signals being downconverted (blue). The periodic variation in the temperature due to the testing environment is clearly visible in the differential phase. Data for the plot was taken from [7].

In order to compensate for this error, a prototype receive system (RF and digital) is being constructed to first characterize the RF system's performance as a function of temperature, and second, to compensate for these thermally-induced errors in real time, through the implementation of corrective signal conditioning that will occur within the architecture of the FPGA. This conditioning consists of both amplitude and phase corrections as a function of the signal frequency within the 200 MHz passband of the transmitted signal.

In addition to the implementation of a digital downconverter on the FPGA, the capability exists to perform calibration measurements of the relative phase between the two channels of the downconverter. These measurements provide a direct measure of the electrical path length differences over time. Because this digital board sits directly in the line of input science data, computing speed and efficiency are key components in the system design. Through experimentation, the combination of the two single-channel ADCs and the FPGA has been shown to be capable of handling the 4.4 kHz pulse repetition frequency (PRF) and 30 sec sampling window for both channels sampling at the full rate of the ADCs (3 GSamp/sec).

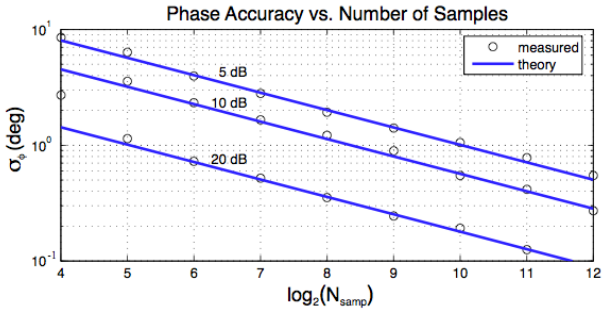

Figure 4: Measured and theoretical phase accuracy versus the number of samples.

An algorithm for monitoring system phase has previously been implemented and tested in software [6]. The most basic form of the algorithm is based on the monitoring of a single-tone signal fed into an analog-to-digital converter. The resulting waveform is compared to a reference waveform of the expected frequency and phase. Differences between the observed waveform and the reference waveform are used to determine the gain and phase characteristics of the observed waveform's signal path. The accuracy of the gain and phase characteristics is dictated by the number of data points and the signal to noise ratio (SNR) of the observed signal. Increasing either the SNR or the number of points increases the accuracy of these estimates in a mathematically predictable way, and the estimation of these numbers can be optimized to achieve a desired level of measurement accuracy.

Figure 4 illustrates measured and theoretic phase accuracy versus sample count. The theoretical phase measurement accuracy (standard deviation in degrees) versus the number of samples is shown on a log scale (on this scale, the number 10 represents 2^{10} samples, or 1024 samples). Simulations using Matlab were performed for varying signal to noise ratios (5 dB, 10 dB, and 20 dB) and for varying numbers of samples. An excellent

agreement between the theory and simulations can be seen. Similar results have also been demonstrated with measured data in the laboratory.

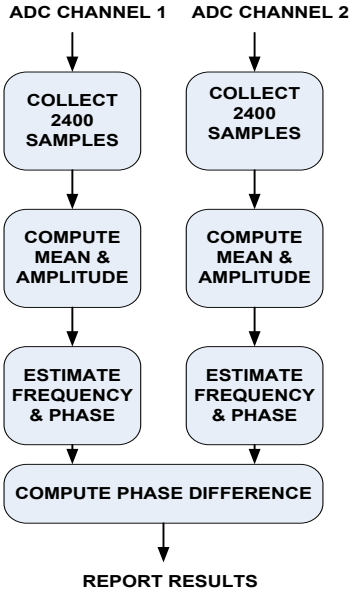

Figure 5. Block diagram for the FPGA differential phase determination algorithm. The number of samples (2400 in the above) can be changed to increase the accuracy of the overall estimation process. The choice of this number is ultimately dependent on the desired estimate accuracy and the signal to noise ratio of the input data.

By implementing this algorithm (as shown in Figure 5) in hardware directly in the intended data stream for the science data, intermediate observations, such as the observations of the known signal, can be discarded after the calculation and the results reported at data rates that are several orders of magnitude smaller than the input data. In the event of a detected anomaly by the hardware-based algorithm, the data can instead be flagged and made available to transmission outside of the signal path, thus providing the opportunity for more detailed inspection by a diagnostic system or engineer.

3. FPGA-Based Phase Testing

The implementation of phase testing in the FPGA takes place as a sequence of steps performed on stored data. Operations take place in the following sequence:

Storing *M* consecutive samples in a FIFO - As shown in Figure 6, a series of consecutive data samples are stored in a FIFO after collection from an ADC. A subset of *M* samples are taken from the FIFO and stored in the RAM bank shown in Figure 6. As values are collected from the FIFO they are converted from 8-bit signed magnitude representation to two's complement representation by the subtraction of 127. The array values stored in RAM (called *v[]*) are located in adjacent memory locations. The size of the array (*M*) should be greater than 1000 and should cover multiple cycles of the input sinusoidal waveform.

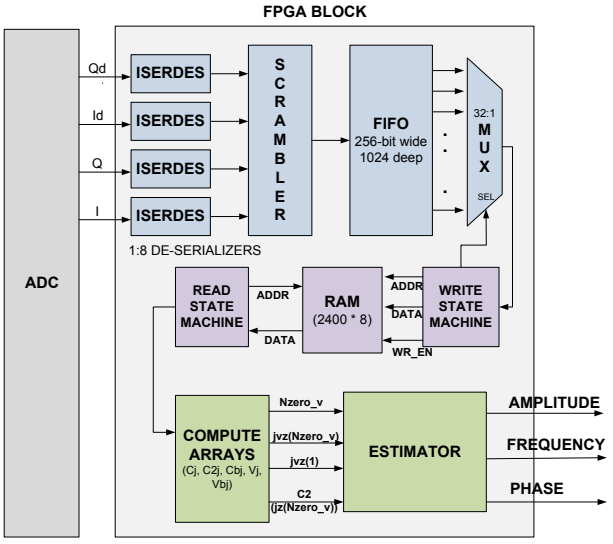

Figure 6. Block diagram of the hardware used in Data FPGA to implement the algorithm that estimates amplitude, frequency, and phase of the reference cosine signal.

Determine and count signal zero value transitions – A total of *M* data values (*v[]*) are fetched one at a time from consecutive RAM locations starting with the first value in an array of *M* values. Each sequence of three consecutive *v[]* values are then averaged using a three-point boxcar filter to create a new sequence of values $vb[j] = \frac{vb[j-1]+ vb[j]+ vb[j+1]}{3}$. The generated *vb[]* values are used to determine positive-to-negative and negative-to-positive transitions in the sinusoidal data stream. The number of these zero transitions (*Nzero_v*) in the *vb[]* array is determined along with the index in *vb[]* where the first (*jvz[1]*) and last (*jvz[Nzero_v]*) zero transition in the data stream occurs.

Determine a running sum of the square of the signal magnitudes - Data values (*v[]*) are also used to calculate a running sum of the square of the magnitude of all points such that $C2[j] = \sum_{i=0}^{j}(v[i])^2$. The intermediate values *vb[]*, *Nzero_v*, *jvz[Nzero_v]*, and *c2[]* are calculated in parallel in the FPGA. These intermediate values are used to calculate estimated amplitude, frequency, and the phase of the first sample in the *v[]* array.

Estimated amplitude calculation - In terms of the values noted above, the amplitude of a signal can be estimated by the following equation.

$$A = \sqrt{\frac{2}{Nzero\_v}} \times \sqrt{C2[jvz[Nzero\_v]]} \quad (1)$$

The numerator in the expression includes the sum of squared magnitudes of all points up to the index of the last data zero transition. The denominator includes the number of zero transitions of the sinusoidal signal. A square root is taken of the resulting value to determine the RMS amplitude value.

Estimated frequency calculation - The estimated frequency of the signal is determined using the following formula:

$$freq = \frac{(Nzero\_v - 1)}{jvz[Nzero\_v - 1] - jvz[1]} \times (F_{SAMP}) \quad (2)$$

Where F_{SAMP} is the system sampling frequency of the analog-to-digital converters.

Calculation of corrected initial phase - The goal of this calculation is to estimate the phase of the initial sample in the array sequence. The calculation of this initial phase takes place in a series of steps. The calculation of initial phase is represented by the following equation:

$$phii = \left(2\pi \times (Nzero\_v)\right) + \left(\frac{\pi}{2}\right) - \left(2\pi \times \frac{freq}{F_{SAMP}} \times jvz[Nzero\_v - 1]\right) \quad (3)$$

This initial phase is then used in a series of iterations to minimize the error involved in the calculations. The equations associated with this analysis are:

$$phic = phii - \frac{\sum_{m=1}^{M} \left(sin(\Theta_m) \times \left(v[m] - Acos(\Theta_m)\right)\right)}{A \sum_{m=1}^{M} (sin(\Theta_m))^2} \quad (4)$$

such that

$$w = 2\pi \times freq \quad (5)$$

$$\mathrm{dt} = \frac{1}{\mathrm{F_{SAMP}}} \quad (6)$$

$$\Theta_{\mathrm{m}} = w \times dt \times (m - 1) + phii \quad (7)$$

In the first iteration, the *phii* value determined in Equation (3) is used as the input to Equations (4) and (7). In subsequent iterations, the *phic* value found in the immediately previous iteration is used as the equation input in the *phii* locations. In Equation (4), the corrected initial phase is effectively refined by evaluating estimated phase values across all *M* data points in *v[]*. As discussed in Section 5, a total of 3 iterations were used to estimate phase values in our experimentation.

To evaluate phase drift, a series of corrected initial phase values (*phic*) can be determined for a series of *M* point data blocks sampled from a single input sinusoidal signal. These values can then be compared to expected phase values to determine if a measurable phase drift can be determined for the sequence. An example of this analysis is presented in Section 5.

4. Experimental Approach

The phase detection circuit described in Section 3 has been implemented in the lab using the experimental setup shown in Figure 7. An arbitrary waveform generator is used to create a 62 MHz signal which is used as a reference signal. The signal is sent through a splitter to generate two equivalent signals. These signals are then input into a state-of-the-art FPGA-based data acquisition system that was designed by the author's research group. This hardware, used to characterize the single tone signal, was built to process L-band signals fed from a down conversion circuit.

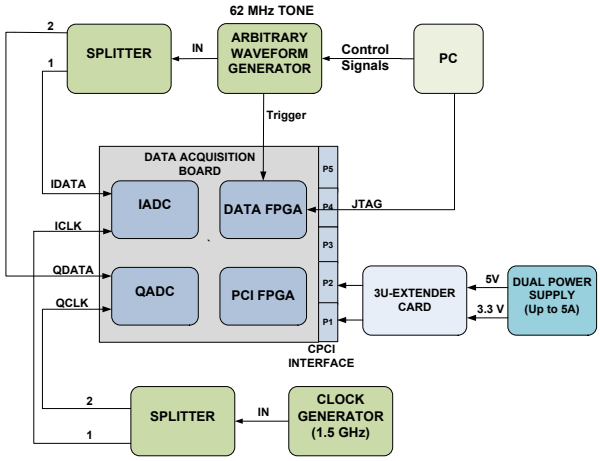

Figure 7. Block diagram of the experimental setup for the phase detection circuit implemented in the laboratory.

A block diagram of the data acquisition system (as shown in Figure 8) shows the major components of the system. The main components include 2 National Semiconductor ADC08D1520 3 GSPS analog-to-digital converters and a Xilinx XC4VFX140 Virtex 4 FPGA (labeled Data FPGA in the figure). Data can be streamed off the board using SATA (serial ATA) and SFP (small form factor pluggable) connectors which handle 1 Gbps data rates from Xilinx RocketIO Multi Gigabit Transceivers (MGT).

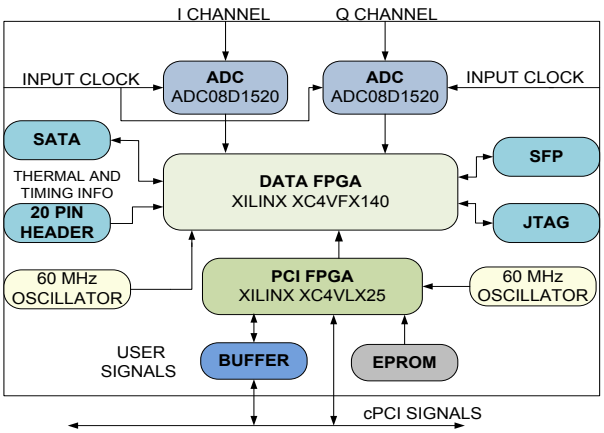

Figure 8. Block diagram of the data acquisition system used to characterize single tone signal for amplitude, frequency, and phase estimations.

As shown in Figure 6, sampled data is transferred from the ADCs to the FPGA as four separate 375 MHz data streams operating at dual data rate (DDR). These data values are demultiplexed by 1:8 SERDES circuitry inside the FPGA. Individual data words are reconstructed using the SCRAMBLER block shown in Figure 6 and input into a 256x1024 FIFO. The 1.5 GHz input clock for the ADCs is also created by a signal generator and splitter pair shown in Figure 7.

A set of steps was followed to collect data for our experiments

- The 1.5 GHz clock is enabled to start ADC sampling
- The 62 MHz input signal is enabled
- A start trigger is sent from the signal generator to the FPGA.
- This trigger enables the FIFOs to store state.
- Processing starts once the input FIFO is full.

The algorithm that estimates the amplitude, frequency and phase of the signal (Section 3) was implemented in the Data FPGA. An accurate measurement of the phase of the signal and the phase difference between the two signals fed into the data acquisition system enables the Data FPGA to perform a low-level signal integrity check. During deployment, this information would be passed along with the telemetry data back to the Earth for further analysis. In our experimental setup, the information is transferred to a PC via an RS-232 serial cable.

Inside the Data FPGA, a 256x1024 FIFO is used to hold a total of 32,768 data samples. There are two state machines employed in the FPGA. The write state machine reads the values from the FIFO, splits it into 32 8-bit samples and writes to successive locations of a RAM. The read state machine reads data at appropriate addresses of RAM and passes the values to a compute stage.

5. Results

Two sets of experiments were performed using the experimental setup described in Section 4. First, the corrected initial phase (*phic*) of each individual channel was determined for a series of *M* point blocks. The actual initial phase (the phase of the first sample of the first sequence) was then determined using Matlab for comparison. In a second set of experiments, the corrected initial phases for two channels were compared to each other for a series of data point blocks. These results were also compared to Matlab calculated results.

The Data FPGA includes two estimators for the two signals feeding the data acquisition system. Table 1 shows the hardware utilization of single and dual channel estimators implemented in Data FPGA.

Table 1. Hardware utilization of single and dual channel estimators implemented in Data FPGA

FPGA Logic Resources	Single Channel	Dual Channel
LUTs	18,599	32,670
Flip Flops	11,906	20,666
RAM Blocks	65	66
DSP Blocks	15	27

The hardware implementation of Equations (1-7) includes a mix of fixed and floating point formats. Values *Nzero_v*, *v[]*, *vb[]*, and *jvz[]* are represented using 8-bit 2's complement values. Values c2[] are represented using 26-bit 2's complement values. The amplitude calculation in Equation (1) requires the conversion of *c2[]* and *Nzero_v* values to floating point. This conversion is performed in the FPGA hardware. Xilinx floating point multiply, divide, and square root operations [8] are performed on 32-bit floating point values. An amplitude result (Equation (1)) is generated as a 32-bit integer. The frequency calculation (Equation (2)) and corrected initial phase calculation (Equations (3-7)) similarly require the conversion of input values to floating point and the use of a floating point divider. These calculations also result in 32-bit integer results. The corrected initial phase (*phic*) calculations additionally require the use of 32-bit Xilinx Cordic blocks to perform *sin* and *cos* operations [9]. In the Virtex 4 architecture, each RAM block contains 18 Kbits of data and multipliers perform 18x18 2's complement operations.

For our initial single channel experiments, a 600 mV peak-to-peak signal of 62 MHz was fed to the I channel ADC (IADC) and a total of 32,768 8-bit consecutive samples were stored in the FIFO shown in Figure 6. These samples were then considered as a series of

consecutive M = 2400 sample blocks. The phase of the first sample of the first sequence was calculated from the samples in post processing (using algorithms written in Matlab) for reference.

For the first 2400 value sample block (block *1*), the comparison of the calculated $phic_1$ and the post-processed value is straightforward. After calculation using Equations (3-7), the comparison can be made directly. For subsequent sample blocks, the calculated values using Equation (4) determine the phase of that block's (i.e. block m) sequence, phic_m. To adjust this value to allow comparison to the phase of the *first* sequence, a phase equivalent to the time delay of 2400 samples at a sample rate of 3 GSamp/sec was subtracted from the current sample block.

$$phic_1 = phic_m - 360\times \Delta T \times (62MHz) \qquad (8)$$

This approach can be extended to monitor/estimate the initial phase of the signal in real time for several seconds/minutes. The incoming data stream is collected in the FIFO continuously for the entire receive window duration and the algorithm is run to estimate the corrected initial phase of the signal at a variety of time points, indicating the possible presence of phase drift

Table 2 shows the FPGA algorithm's estimated phase for consecutive blocks of 2400 samples each measured at a sample rate of 3 GSamp/sec (Equations (3-7)) for the I-channel ADC being read into the FPGA. As described in Section 3, three iterations of *phic* calculation using Equation (4) were necessary to converge to the result shown in Table 2. The signal's true starting phase (55.03 degrees) has been determined in post-processing, and is accurate to within 0.01 degrees. The starting phase itself, while somewhat arbitrary, should stay constant for each subsequent block. The FPGA estimated phases, *phic*, are within 3.1 degrees of the true value and better than 1% accurate overall (out of 360 degrees of phase), thus validating our approach. The small remaining error is expected to be due to round off error in the data processing and we are currently investigating how to make the measurement accurate to within its theoretical bounds of 0.01 degrees.

A similar experiment was performed for a 62 MHz input signal for the Q channel ADC (QADC). Table 3 shows the phase of the signal monitored for 7 successive sample blocks in the FIFO. For both the results in Table 2 and 3, the clock speed of the FPGA circuitry was 60 MHz. A total of 0.37 ms was needed to determine *phii* and 10.3 ms was required to determine *phic*. These execution times were determined starting from the initial sample read from RAM.

The second experiment used to validate the corrected initial phase estimator involved a comparison of initial phase for signals arriving simultaneously on both I and Q channels. In this experiment, *phic* is determined for both channels and subtracted to determine the phase difference. As shown in Figure 7, a single signal from the arbitrary waveform generator is split into I and Q channels. Since equal length 2m cables were used for both channels, minor phase differences were expected. While there are various reasons that the phase might be different for the two channels (e.g. timing differences between the two ADCs) the phase difference between the two channels as a function of time, if performed accurately, shows path-integrated temperature differences between the two channels that feed into the ADCs.

Table 2. Corrected initial phase estimation (*phic*) for IADC versus actual phase determined by post-processing analysis of the first sample block 1 which determined that the actual phase was 55.03 degrees, measured to an accuracy better than 0.01 degrees. Estimate errors in percentages are taken as a proportion of the measured phase error out of 360 degrees.

Sample block	Estimated phase, phic	Estimated error	
1	58.1	-3.0 deg	-0.8 %
2	53.1	+2.0 deg	+0.5 %
3	53.0	+2.0 deg	+0.6 %
4	55.7	-0.7 deg	-0.2 %
5	55.5	-0.4 deg	-0.1 %
6	58.1	-3.1 deg	-0.9 %
7	53.1	+1.9 deg	+0.5 %

Table 3. Corrected initial phase estimation for QADC versus actual phase (45.43 degrees) determined by post-processing analysis of sample block 1.

Sample block	Estimated phase, phic	Estimated error	
1	47.7	+2.3 deg	+0.7 %
2	47.8	+2.4 deg	+0.7 %
3	45.2	-0.2 deg	-0.0 %
4	42.6	-2.8 deg	-0.8 %
5	42.8	-2.6 deg	-0.7 %
6	47.7	+2.3 deg	+0.7 %
7	47.8	+2.4 deg	+0.7 %

Actual (determined from post-processing) and FPGA estimated phases for a split 62MHz signal input into both I and Q ADC channels are shown in Table 4. A slight error is seen in the phase estimation due to round off errors and sampling accuracy, which at times manifests itself as a seven degree error when an entire sample is dropped (see blocks 4 and 5). With the exception of the errors induced from a dropped sample, it can be seen that the phase difference measures between the post-processed and FPGA processed data is consistent to within 0.03°, an

indication that the FPGA phase measurement errors are systematic (and not random), and therefore cancel one another when calculating phase differences.

Table 4: True (Matlab) versus FPGA measured phase differences for both I and Q ADC channels. Difference errors reflect the difference between the true and measured phases.

Sample block	True ($phic_I$-$phic_Q$)	FPGA ($phic_I$-$phic_Q$)	Difference Error	Difference Error
1	29.22°	29.78°	0.56°	0.2%
2	29.22°	29.77°	0.55°	0.2%
3	29.20°	29.74°	0.54°	0.2%
4	29.23°	22.18°	-7.05°	1.9%
5	29.20°	22.28°	-6.92°	1.9%
6	29.18°	29.78°	0.60°	0.2%
7	29.19°	29.77°	0.58°	0.2%
8	29.18°	29.74°	0.55°	0.2%

A final experiment was performed to evaluate an estimate of the signal frequency using Equation (2). Using a 2400 sample block, an estimate of 61.9964 MHz was determined for the 62.0000 MHz signal.

6. Conclusions

For many spaceborne systems, the on-board rapid calculation of input signal phase differences can provide important diagnostic information. The presence of high performance adaptive hardware in the signal processing datapath provides an opportunity to quickly and efficiently evaluate signal phase and identify phase drift. In this project a phase estimation circuit has been implemented in FPGA hardware and experimentally tested. We demonstrate the approach using a new state-of-the-art data acquisition system developed for the NASA Surface Water Ocean Topography (SWOT) project. Phase calculations with an error less than 2.0% were determined using our adaptive approach.

In the future we plan to improve the accuracy of our circuit by using longer sample block lengths and improving precision for hardware algorithms implemented on the FPGA.

Acknowledgments

This research was funded by NASA's Earth Science Technology office under a grant from the Advanced Component Technology program (grant #ACT-08-0048). The authors wish to thank National Semiconductor, Xilinx Corporation, and MS Kennedy for their donation of circuit components. The assistance of Andrew Boissonault and Eric Bovell is also appreciated.

References

[1] National Research Council, Earth Science and Applications from Space: National Imperatives for the Next Decade and Beyond, Space Studies Board, 2007.

[2] J. Curlander and R. McDonough, Synthetic aperture radar - Systems and signal processing, John Wiley and Sons, 1991.

[3] P. Rosen, S. Hensley, I.R. Joughin, F.K. Li, S.N. Madsen, E. Rodriguez, and R.M. Goldstein, "Synthetic Aperture Radar Interferometry," IEEE Proc., 88(3), 333-382, 2000.

[4] B. Tapley, S. Bettadpur, M. Watkins, and C. Reigber, "The gravity recovery and climate experiment: Mission overview and early results," Geophys. Res. Let., 31, L09607, 2004.

[5] P. Siqueira, R. R. Ahmed, J. Wirth, A. Bachmann, "Variable Precision Two-Channel Phase, Amplitude, and Timing Measurements for Radar Interferometry and Polarimetry," IEEE Trans. Micw. Theor. Tech., 55(10), 2248-2256, 2007.

[6] P. Siqueira, R. Tessier, A. Swochak, K. Fu, D. Esteban-Fernandez, and B. Heavey, "A single-stage, two- channel Ka-band to digital, thermal compensating receiver for SWOT," NASA Earth Sci. Tech. Forum, 4 pp., Arlington, VA, 2010.

[7] P. Siqueira, K. Srinivasan, E. Insanic, and R. Ahmed, "A Cross-Track Ku-Band Interferometer for Topographic and Volumetric Depth Measurements," *Aerospace Conference Big Sky, Mt.,* pp.1-8, March 2007

[8] Xilinx Corportation, Floating Point Operator v5.0, DS 335 data sheet, June 2009.

[9] Xilinx Corportation, Cordic v4.0, DS 249 data sheet, April 2009.