FPGA Implementation of a Runtime User Configurable Chirp Signal Generator for Multi Mission Synthetic Aperture Radar Imaging

Hamza Mansoor Manya and Ashfaq Amjad Khan

***Abstract*— The concept of pulsed RADARs is associated with Chirp Signal Transmission. Chirp signal provides high precision in target resolvability with the flexibility of lower average power. The transmitted Chirp Signal defines the resultant image resolution, which varies from mission to mission. The objective of this paper is to discuss the implementation of a FPGA based runtime user configurable Chirp Signal Generator using the Xilinx System Generator. The designed system is capable of transmitting three different types of Chirp Signal; i.e. Unidirectional Up-Chirp, Unidirectional Down-Chirp and Bidirectional Chirp. The system can generate an output Chirp of Bandwidth between 10 to 100MHz with a transmitted pulse duration of between 1 to 10 Micro Seconds and a Pulse Repetition Frequency of between 100 to 10000Hz. The sampling frequency of the system is kept constant at 150MHz.**

***Index Terms*— Bandwidth, Chirp, FPGAs, Pulse Repetition Frequency, Pulsed RADAR, SAR, System Generator, Transmitted Pulse Duration**

I. Introduction

Synthetic Aperture RADAR (SAR) is a common tool in remote sensing. It finds its applications in global positioning, disaster management, agricultural monitoring e.t.c. SAR was first proposed by Carl Wiley in 1951 [1] which described the use of Doppler frequency analysis to improve radar image resolution. SAR has many advantages over optical imaging, for example SAR can be used for soil monitoring.

Originally the concept of RADAR imaging was based on continuous wave RADARs. The echoes were hard to resolve due to overlapping and the average power of the system was very high. This gave birth to the idea of RADAR imaging based on pulsed RADARs. [2] However, with this new concept came many new problems, for example; the echoes from objects with lower RADAR Cross Section (RCS) in presence of objects with larger RCS became impossible to detect. Such problems were resolved by frequency modulated (FM) pulsed RADARs Chirp. This concept greatly relaxed the processing complexity as the echoes became relatively spaced out as compared to continuous wave RADARs. The concept of FM Pulsed RADAR has its own pros and cons. [3]

The focus of this paper is discuss a FPGA based Runtime User Configurable Chirp Signal Generator for multi mission Synthetic Aperture Radar Imaging Payload. The design requirements for the desired SAR Payload System are illustrated by the Table I.

TABLE I
SAR Payload system parameters

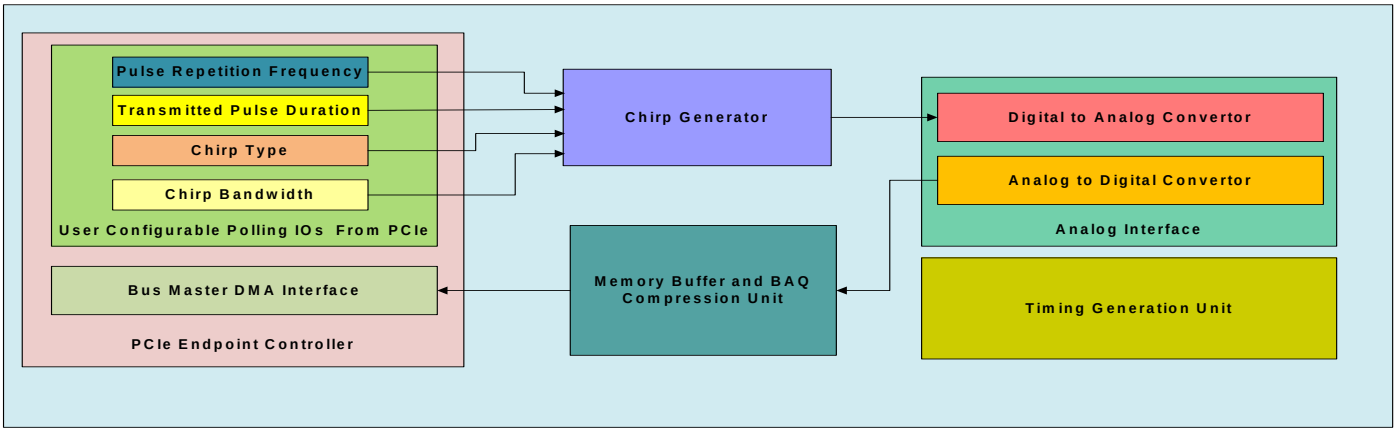
System Parameters	Specifications
Mode of Operation	Stripmap
Operating Frequency	4-6 GHz (C-Band)
Chirp Bandwidth	10-100MHz
Azimuth Resolution	0.05-1m
Range Resolution	1.5-15m
Platform Speed	100-150m/s
Altitude of Platform	1-5km

The requirement of a runtime user configurable chirp signal generator arises from the system requirements of the payload. The sampling frequency is kept 150MSPS to ensure that the resultant chirp signal is smooth. The systems is capable of generating chirp signal of all three type, bidirectional, unidirectional upward and unidirectional downward, however a degraded output is observed at bandwidths beyond 60MHz.

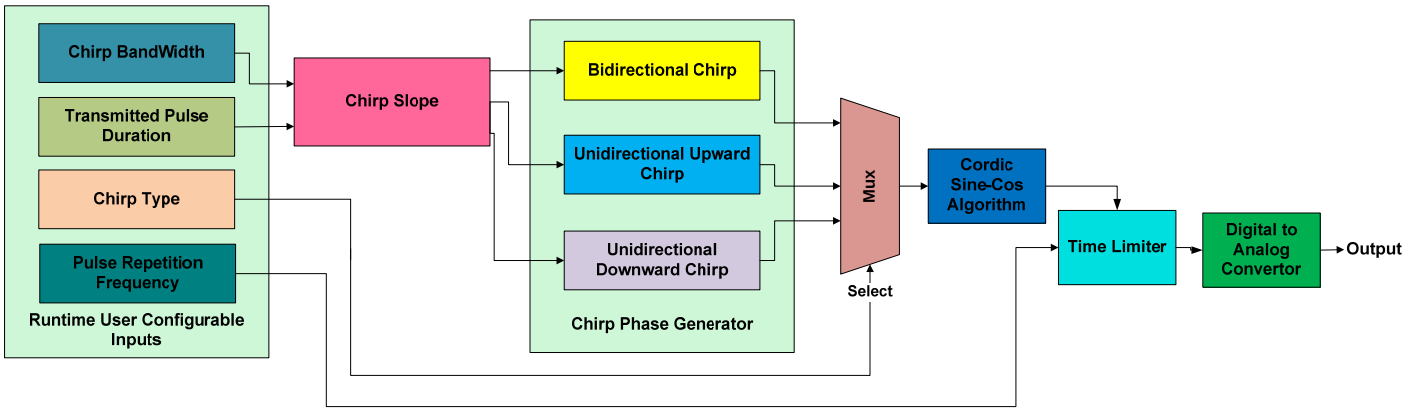
The PRF is also kept user configurable because it is dependent upon platform speed, which is variable in this case. The average transmitter power of the system changes with a change in transmitted pulse width. When altitude is higher a higher average power is required to ensure that the SNR of the received signal is in allowable range, therefore, with a variable platform altitude a requirement for variable transmitted pulse duration arises.

II. Related Work

A similar approach is presented in [4] which is based on the design of a FPGA based reconfigurable chirp signal generator for Unmanned Air Vehicle SAR Payload. However, the system is user configurable in the pre-synthesis phase and only generates an unidirectional upward chirp. The author discusses two approaches in this paper; Memory Based approach in which the signal is stored in a memory and the second a DDS compiler based approach. In this paper a non memory based approach is used. However, the signal generation is based on Cordic Algorithm.



A. Runtime User Configurable SAR Acquisition Controller



B. Runtime User Configurable Chirp Signal Generator

Fig. 1. Proposed Design of the System

The pros and cons of using chirp signals in pulse radar imaging are discussed in [5]. The paper discusses the basic concepts of synthetic aperture radar imaging and matched filtering approaches.

The issues in SAR imaging include processing latency, which is addressed in [6]. The idea is how to effectively divide the acquired data frame into tiles to increase the processing latency by processing these tiles in parallel without having an intense effect on the processed image quality.

Similarly, SAR antenna simulation is a highly time consuming and mathematically complex process. This simulation performed in [7].

III. Principles of Signal Generation in Linear Frequency Modulated Pulsed RADARs

Synthetic Aperture RADAR systems are frequently associated with the use of pulsed linear frequency modulated signals (chirp). Chirp signal incorporates numerous sinusoid frequency components in a single pulse.

The frequency slope of a Chirp signal can be of three different types i.e. unidirectional upward frequency slope, unidirectional downward frequency slope and bidirectional frequency slope. Generally in SAR the preferred Chirp type is bidirectional frequency slope. The advantage of using this type of chirp signal is, to attain a particular bandwidth in a Chirp pulse the highest frequency component theoretically required is half of bandwidth edge while in a unidirectional chirp the highest frequency component should at least be the edge of the bandwidth. The use of a bidirectional frequency slope therefore immensely relaxes the sampling frequency requirements and consequently the hardware requirements.

The generation of all three types of signals only differs in instantaneous phase generation. The preceding section elaborates the processes of phase generation of all three types.

IV. Design Description

The Fig. 1 (B) illustrates the block level detail of the runtime user configurable chirp signal generator unit design. The runtime user configurable chirp signal generator unit will become a integral part of the Runtime User Configurable SAR Acquisition Controller. The block level detail of the Runtime User Configurable SAR Acquisition Controller are illustrated in Fig. 1 (A).

The scope of this paper is to discuss the design detail of the runtime user configurable chirp signal generator unit. The system was designed using the Xilinx System Generator. The details of the different blocks of the runtime time user configurable chirp signal generator are illustrated below:

A. Runtime User Configurable Inputs:

The system allows the user to select a transmitted pulse duration between 1Micro-Sec to 10 Micro-Sec. The Bandwidth can be selected in the range 10 and 100MHz.

The system also allows the user to select from the different types of chirps (uni-directional up chirp; uni-directional down chirp; bidirectional down-up chirp) according to user defined mission requirements. The pulse repetition frequency is dependent on the velocity of the platform and can be varied in the range of 100-10000Hz.

B. Chirp Slope Generator:

The Chirp slope generator takes the chirp Bandwidth and the transmitted pulse duration (Chirp Sweep Time) as input and computes the slope of the output signal.

The Chirp Slope for a bidirectional chirp is computed by dividing the input bandwidth by twice of Chirp Sweep time:

$$Slope\ \_B = \frac{BW}{2 * Tp} \qquad (1)$$

In case of a unidirectional chirp the slope is computed by dividing the input bandwidth by Chirp Sweep time:

$$Slope\ \_U = \frac{BW}{Tp} \qquad (2)$$

C. Chirp Phase Generator:

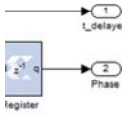
The Chirp Phase generator takes Chirp Slope as input and computes the instantaneous phase of the output signal. In this block the phase of all three chirps are computed. Firstly the time bin array is computed by linearly spacing out the sweep time into intervals of sampling time.

This logic is simply implemented using a counter which resets when count times sampling time equals to sweep time. The value of each time bin is computed by multiplying the count by sampling time. The internal structure of this block is illustrated by Fig. 2:

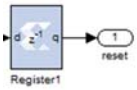
The phase of a bidirectional chirp is given by the formula:

$$Phase\ \_B = t * (t - Tp) * \prod * Slope\ \_B \qquad (3)$$

The implementation of this part of the system is elaborated by the Fig 3:



The reset logic of the phase generator comprises of a comparator, which compares the current value of sample count times sampling frequency to the sweep time. When the two become equal the output of this comparator becomes 1, which resets the sample count of the phase generator block. The logic is illustrated in Fig. 4.



The phase of a upward Unidirectional is given by:

$$Phase\ \_UD = t^2 * \prod * Slope\ \_U \qquad (4)$$

The implementation of this part of the system is elaborated by the Fig. 5:

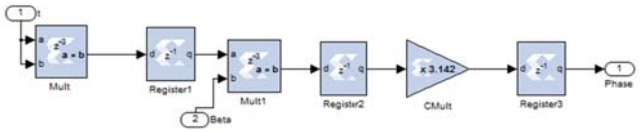


Fig. 5. Phase generator of Upward Unidirectional Chirp

The phase of a downward Unidirectional is given by:

$$Phase\ \_UD = t(BW - t) * \prod * Slope\ \_U \qquad (5)$$

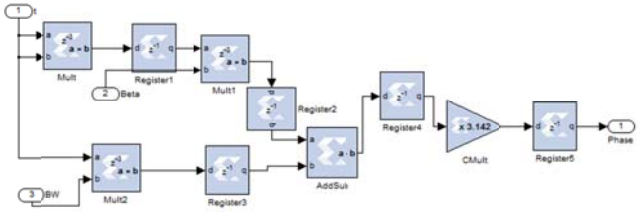


Fig. 6. Phase generator of Downward Unidirectional Chirp

The implementation of Eq (5) is illustrated by the system in figure 6.

D. Cordic Sine Cos Block:

The function of this block is to generate the sine and cosine of the input phase, which is routed to the block through a multiplexer. The function of this multiplexer is to forward the phase of the user desired chirp for further processing.

The logic of the Sine Cos Generator were implemented through a predefined Cordic Sine Cos Algorithm Block in System Generator. The block can only generates valid outputs at phase inputs of between -π and π, however, the phase generator logic generates the phase in multiples of π. In order to cater with this problem a phase limiter logic was implemented, which maps the input phase into a range between -π and π. The implementation of this logic is elaborated in Fig. 7.

The logic converts the input phase to an integer by multiplying it by a constant number and divides it with another integer number which is double the value of the constant multiplier. The idea behind this logic is the first constant represents π and the second 2π. The fractional output of the divider generator is between -π and π.

The output of the phase limiter was simply fed to the Cordic Sine Cos Block and a continuous Chirp wave of the desired Sweep time (transmitted pulse duration) was observed at the output of this block. This was then fed to the time limiter block.

E. Time Limiter Block:

The time limiter block coverts the input continuous wave chirp signal to pulsed chirp signal. The logic is to compare the time bin count times sampling time to the transmitted pulse duration. The output of this comparator is fed to a multiplexer which forwards the Chirp Signal to the output of the block from t=0 till t=Tp and forwards zero to the output of the block from t=Tp+Δ till t=(1/PRF)-Δ. After this the whole system resets and the next transmission cycle starts. The implementation of this logic is illustrated in Fig. 8.

F. Digital to Analog Convertor:

The System Generator Design was migrated to ISE System Navigator by Verilog Netlist conversion and integrated with the Core of the digital to analog convertor (DAC) Controller. The sampling frequency of DAC was kept constant at 150 MSPS.

V. Implementation Requirements

This design of the runtime user configurable chirp signal generator will become an integral part of a Synthetic aperture Radar Data Acquisition Controller, which requires PCIe and DDR3 support. Hence this design was optimized for a Virtex 6 family FPGA device. Table II. illustrates the implementation results of this design on a Virtex 6 FPGA.

TABLE II
Implementation Results

Parameter	
FPGA Family	Xilinx Virtex 6
Maximum Frequency	351MHz
Slices	759

VI. Testing Methodology & Results

The testing methodology used in this paper is hardware in the loop co-simulation in MATLAB Simulink Environment, Using the Xilinx System Generator. System Generator, in this environment, provides support for hardware software co-simulation. The software provides the stimulus to the JTAG co-simulation block, which acts as an interface between the Simulink model and the design running on the FPGA.

In response to this stimulus the design on the FPGA generates some outputs, which are displayed on Simulink Scopes or displays. When using this methodology for functional verification, there is no need to perform traditional HDL simulations.

Once the functional verification is done, System Generator is capable of generating HDL equivalents of the models in either VHDL or Verilog. These HDL source codes are highly optimized, as the design optimization and reduction phase is iterative. The HDL equivalents of the models can then be integrated with any design in Xilinx ISE environment.

Four sets of tests were performed for design validation by varying one parameter at a time. The results of these tests are presented in the proceeding sections:

A. Effect of Change in Bandwidth

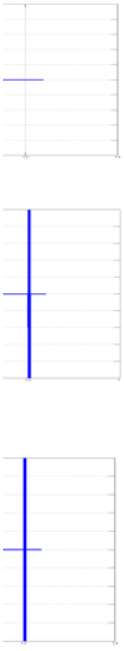
The system was firstly tested by varying the bandwidth. Three datasets at 30, 45 and 60MHz were compiled to observe the effect of change in bandwidth. The analysis was performed by keeping the chirp type bidirectional, Tp 3 micro seconds and PRF 1000Hz. Fig. 9, Fig. 10 and Fig. 11 illustrate the outputs results in system generator and MATLAB, and the Spectrum plot of the output.

ɔt

ıds Tp

B. Effect of Change in Pulse Repetition Frequency

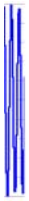
The system was tested by varying the pulse repetition Frequency (PRF). Three datasets at 1000, 5000 and 10000Hz were compiled to observe the effect of change in PRF. The analysis was performed by keeping the chirp type bidirectional, Tp 3 micro seconds and Bandwidth 60MHz. Fig. 12 illustrates the outputs results in system generator.

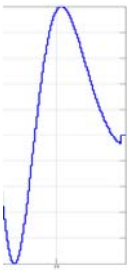


ncy

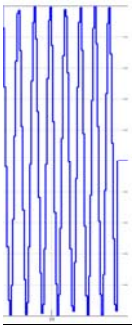
C. Effect of Change in Transmitted Pulse Duration:

With a change in transmitted pulse duration the output chirp signal contracts and expends in time domain. In order demonstrate this effect a bidirectional chirp of 60MHz Bandwidth was recorded at Tp equals to 3Micro Seconds, 5Micro Seconds and 7Micro Seconds. The results are illustrated by in Fig. 13:





Seconds Tp



econds Tp

D. Effect of Change in Chirp Type:

In order to validate the functionality of change in output chirp type the bandwidth was kept constant at 10MHz , the PRF was kept constant at 1000Hz and the transmitted pulse duration was kept constant at 3 Micro Seconds. Fig. 14, Fig. 15 and Fig. 16 illustrate the outputs:

Fig 14: Bidirectional 10MHz Chirp with 3 Micro Seconds Tp

CONCLUSION

With the evolution and availability of off the shelf reconfigurable digital systems, like FPGAs, the trend for implementing digital data acquisition and signal processing systems has increased by the day. The digital systems offer a greater precision than their analog counterparts when it comes to reconfiguration.

In this paper an architecture of a runtime user configurable digital chirp generator is discussed. The architecture of the system is based on Cordic Sine generation and is non memory based. The merits of using a bidirectional chirp signal in SAR digital transceivers are also discussed.

The requirements which led to this work required pin point precision and runtime reconfigurabilty. The designed system was tested with a wide range of stimuli and has proven to fulfill all system requirement.

This system is however just a integral part of the synthetic aperture payload which currently in design phase. This system was ported to its HDL equivalent and is now being employed in the overall SAR digital transceiver design.

REFERENCES

[1] Wiley, C. A., "Pulse Doppler RADAR Methods and Apparatus," United States Patent, No. 3, 196, 436, Filed, August 1954, 1965.
[2] Mehrdad Soumekh “Synthetic Aperture RADAR Signal Processing”, 1999.
[3] A.A. Khan and H.M. Manya, "Analysis of the Effect of Chirp Parameters on Image Quality of Synthetic Aperture Radar", Proceeding of the International Conference on Emerging Technologies 2012, 89-92, 2012
[4] M. Y. Chua and V. C. Koo, "FPGA-Based Chirp Generator For High Resolution UAV SAR", Proceedings of the Progress In Electromagnetics Research, PIER 99, 71-88, 2009
[5] M. Y. Chua and V. C. Koo, " An introduction to Synthetic Aperture Radar", Proceedings of the Progress In Electromagnetics Research, Vol. 2, 27–60, 2008
[6] Einar-Arne Herland “ERS-1 SAR Processing with CESAR”, 2007.
[7] Matthew Schlutz “Synthetic Aperture Radar Imaging Simulated in MATLAB”, June, 2009.