A real-time UAV INSAR raw signal simulator for HWIL simulation system

Wei Li, Houxiang Zhang, Hans Petter Hildre
Faculty of Maritime Technology and Operations
Aalesund University College
N-6025, Aalesund, Norway
E-mail: {weli & hozh & hh}@hials.no

KEYWORDS
INSAR, HWIL, FPGA

ABSTRACT

In this paper, an FPGA based UAV INSAR raw signal simulator is designed to address high computational complexity. It is based on a time domain raw signal generation algorithm and can compute in real time. This signal simulator is designed for Hardware-in-the-loop (HWIL) UAV INSAR simulation which can be used for UAV operator training and system verification. Multi-FPGAs are used in this simulator with optimisation methods to improve FPGA resource costs, including a modified non-restoring algorithm for slant range computing, as well as pipelined FFT and IFFT processors for a fast convolution method.

INTRODUCTION

Interferometric Synthetic Aperture Radar (INSAR) systems are special types of radar that produce three dimensional high resolution images (comparable to optical sensors) in all weather conditions, night and day (Madsen, S. N et al.1998). High performance INSAR systems utilise sophisticated signal processing algorithms and need complicated and costly radar electronics and processing units.

In recent years, a new remote sensing technology based on Unmanned Aerial Vehicle (UAV) INSAR has emerged. This technology provides a great potential for detailed monitoring and surveillance of areas covering of up to a few thousand square kilometres with a relatively low cost. However, the high performance UAV INSAR is challenging because of the highly dynamic platform and the baseline error caused by antenna oscillation. The raw signal simulator (A. Mori et al. 2004) is useful for testing and verifying the function and performance of the UAV INSAR system.

SAR raw signal generating algorithms can be classified in two types: frequency domain (FD) and time-domain (TD) (G. Franceschetti 1998 and A. Mori et al. 2004). The TD algorithm can easily consider the real trajectory of the platform and other effects such as mechanical structure oscillation and orbital deviations, considerable variation of the velocity vector in the case of an UAV platform which is ideal for a Hardware-in-the-loop (HWIL) simulation system with real time measurement methods.

However, most TD INSAR simulators do not work in real-time because of the high computational load for the raw signal generating algorithm and the ultra low delay requirement. FPGA (Field-programmable gate array) is an ideal device for implementing the HWIL simulator, thanks to the advantages of high parallel computing performance, low latency and flexible I/O interfaces.

In this paper, multiple FPGAs are used to design a UAV INSAR raw signal simulator for the HWIL simulation system. In order to achieve real-time processing, some optimisation is presented including slant range calculation, fast convolution processor and memory distribution methods. The paper is organised as follows: first, the HWIL simulation system is introduced; second, the INSAR TD raw signal algorithm is described; third, the system architecture of simulator is presented; fourth, the processor design for the raw signal generation is proposed with FPGA design results.

HWIL UAV INSAR simulation system

The UAV INSAR raw signal simulator can be used for HWIL UAV simulation systems, as shown in Figure 1. It is mainly used to design and verify the function and performance of UAV INSAR systems. In this HWIL simulation system, parts of the virtual model are replaced by the actual physical model. For example, the UAV motion simulator is used to simulate UAV dynamics model and is responsible for sending commands and parameters related to the simulation. The INSAR raw signal generator is used to simulate the raw signal according to the radar and platform parameters. The UAV INSAR processor is connected with these devices and works just like in the real environment.

The INSAR target signal simulator is a key part of this HWIL simulation system, which has real-time generation of SAR raw signal and closed-loop UAV simulations. Different from ordinary radar simulation systems, the INSAR raw simulation system needs a large amount computation and a high accuracy. As such, the SAR simulator has strong computing performance and a highly parallel and optimised software design to take full advantage of the hardware computing resources.

Proceedings 28th European Conference on Modelling and Simulation ©ECMS Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani (Editors)
ISBN: 978-0-9564944-8-1 / ISBN: 978-0-9564944-9-8 (CD)

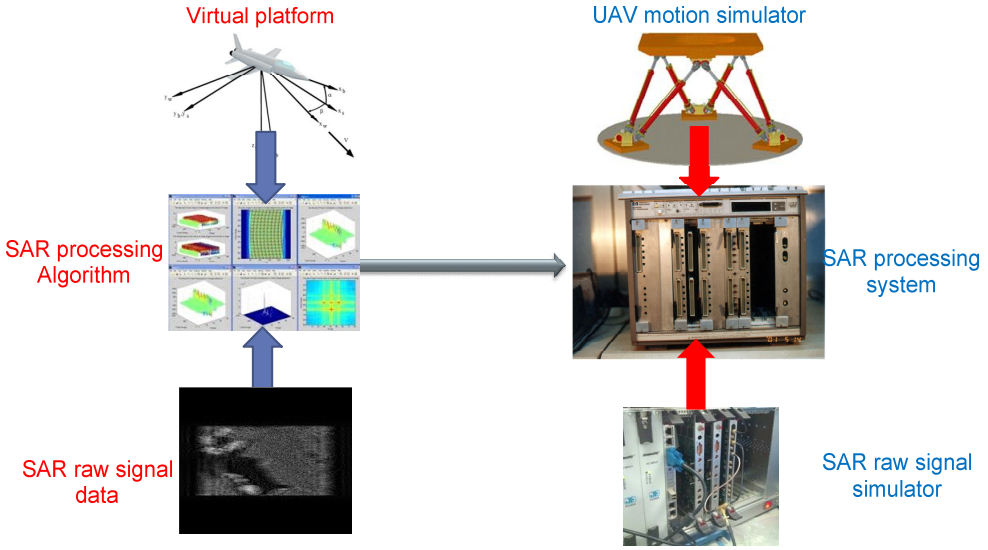


Figure 1: SAR HWIL simulation system

INSAR raw signal simulation algorithm

The geometry of an INSAR system is shown Figure 2, where S1 and S2 are the master and slave antennae. Every PRT, the radar transmits pulse signal in each azimuth position. The raw signal received by the antennae can be described by Eq. 1:

$$s_r = \sum_{i,j} G_{ij} RCS_{ij} \exp\left[-j4\pi R_s / \lambda\right] \cdot \delta\left[r - R_s\right] \otimes \frac{2}{cv} s_t \quad (1)$$

where

i, j index of sampling in azimuth and range

RCS_{ij} backscattering coefficient of scatter (i,j)

G_{ij} pattern antenna weight

λ wavelength

R_s slant range between a point scatter and the antenna phase centre

$\delta(\cdot)$ pulse envelope

c speed of light

v velocity of platform

s_t transmitted signal

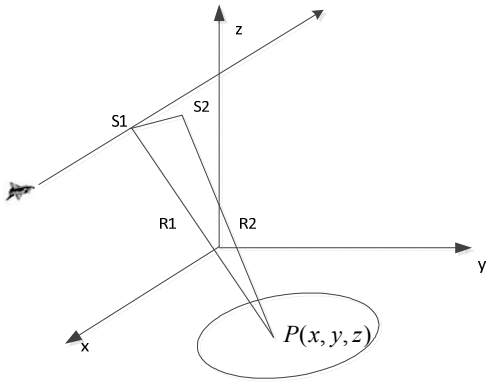


Figure 2: Geometry of an INSAR system

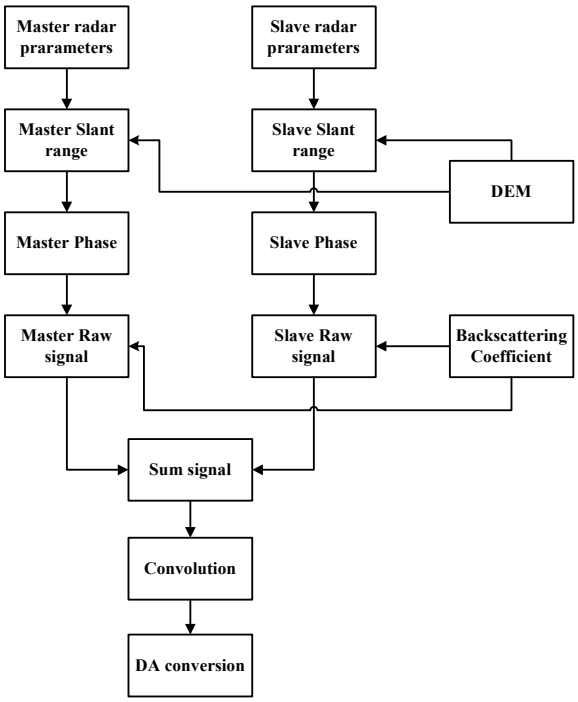


Figure 3: INSAR raw data generating procedure

The raw data generating procedure is based on the block diagram shown in Figure 3. The steps are: 1) Receiving the simulation or test parameters, such as the platform position and velocity; 2) Computing the slant range of scatters in the scene for master and slave antennae; 3) Computing the azimuth phase and multiplied with the RCS; 4) Accumulating the return signal of the same range cell; 5) Convolution with the transmit signal; 6) Converting the digital signal to an analogue signal.

SYSTEM ARCHITECTURE

In order to satisfy the need for processing the TD INSAR raw signal simulation algorithm in real-time, in this paper, multiple FPGAs are used in parallel. The simulator provides master/slave channels and each channel signal is generated using one computing board. As shown in Figure 4, the simulator is based on CompactPCI bus architecture with one main computing board, one slave computing board, an analogue signal generating board and a controller board. The controlling communication is based on PCI bus architecture and the high speed transfer of data between the modules is made possible via backplane LVDS bus architecture.

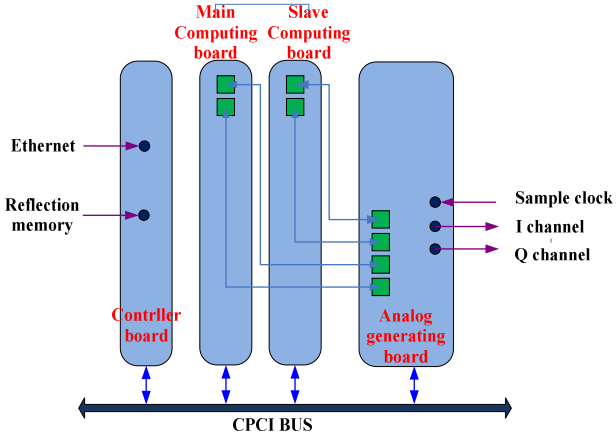
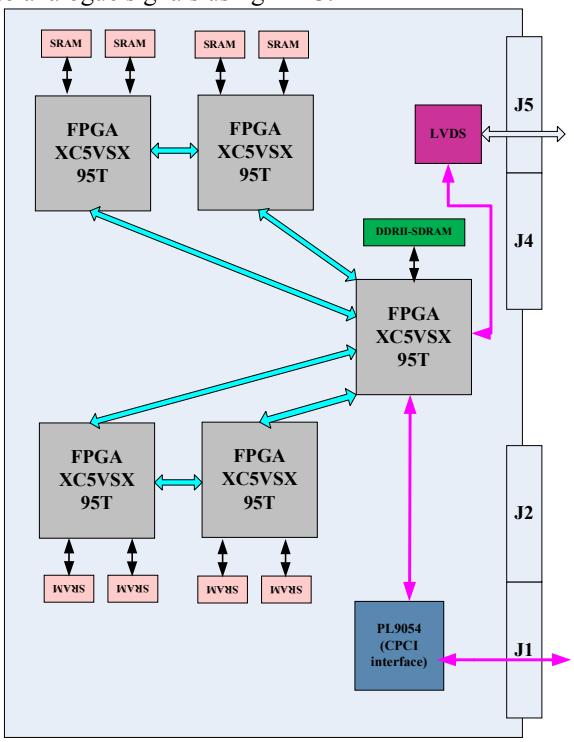


Figure 4: Simulator system architecture

The hardware structure and implementation of the computing module is shown in Figure 5, which was designed and implemented by the authors. The main computing board and slave computing board are of the same structure and are reconfigurable. Five XC5VSX95T FPGAs are used in each computing module and DDRII-SRAM are used as external memory for Digital elevation model (DEM) data and backscattering coefficients because of their high speed, low delay and relatively simplified controller interface. The memory width is 144bits and memory capacity is 16MB for every FPGA. The analogue signal generation module is composed of one XC5VSX95T and one XC4VSX55T which is shown in Figure 6. The FPGA XC4VSX55 is used to transmit simulation parameters to the computing module. The FPGA XC5VSX95T is used to receive the digital raw signal and converts to analogue base band signal through DAC AD9736.

PROCESSOR DESIGN FOR INSAR RAW DATA SIMULATION

According to the system architecture, there is one master and one slave computing module for the digital raw signal generating. The processor architectures are similar in these two modules when multiple FPGAs work in parallel. The main architecture and function of the simulator are shown in Figure 7. The computing boards are mainly used to compute the slant range between radar and targets, phase and sine/cosine value radar return signals. The signals are summed up along the slant range and then sent to the analogue converting board. The analogue converting board is used mostly for convolution with the transmission signal and conversion to analogue signals using DAC.



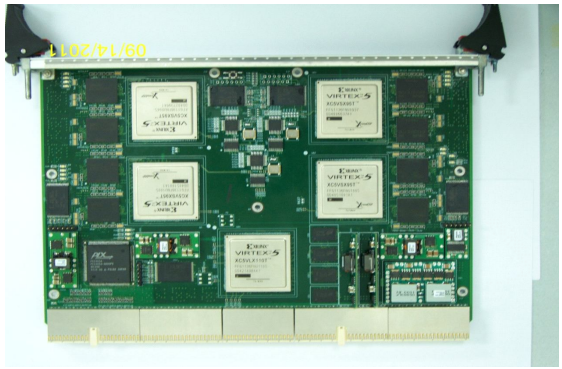


Figure 5: Hardware structure of computing module

In this section, details about the processor design will be described.

Parallel computing architecture

In each computing board, there are four FPGA for main computing and one FPGA for data summing and communication. Each computing FPGA is equipped with four 36bit DDRII-SRAM as external memory for the backscattering coefficient and DEM. The whole SAR imaging area is distributed to every FPGA external memory so that every FPGA can work in parallel without excess communication. According to the radar radiation pattern the area can be divided in the range or azimuth direction. In this simulator, the scene is divided into 4 parts in the range direction corresponding with 4 processing FPGAs. Every FPGA has 24 cores to compute the raw signal. When the processing of signal coherent accumulation is complete, the result of every

processing unit needs to be accumulated for each processing module.

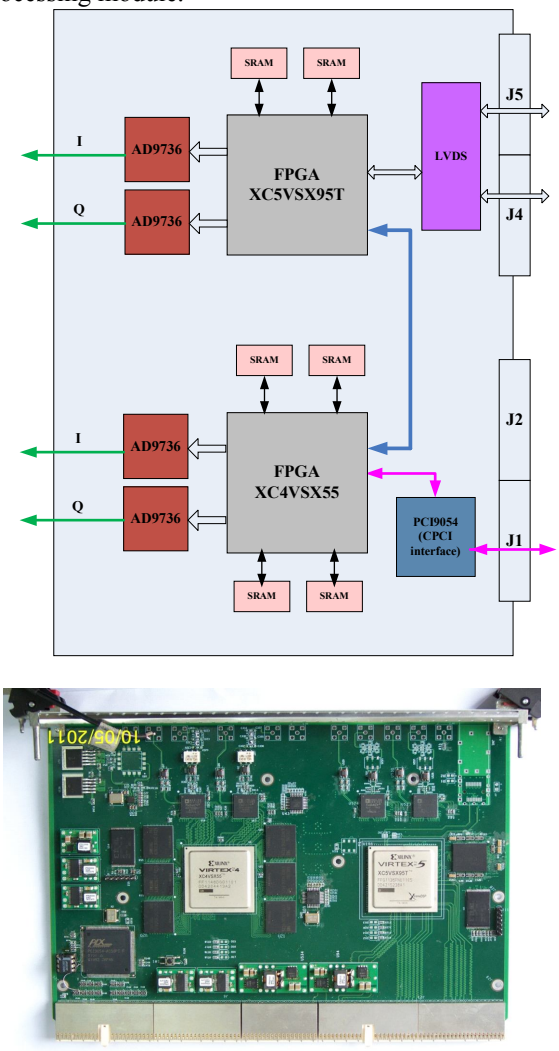


Figure 6: Hardware structure of analogue signal generation module

Slant range computing is one of the main computing steps with highest logic resource cost of FPGA. It is essential for high quality INSAR raw signal generating with high phase accuracy.

$$R_s = \sqrt{(x-X_0)^2+(y-Y_0)^2+(z-Z_0)^2}$$

$$\Phi_s = \exp\left[-j4\pi R_s / \lambda\right] \quad (2)$$

As shown in Eq. (2), the phase error is directly related to the slant range error. For example, with the centre frequency of 10GHz and the phase error of 3 degrees, the range error should be below 0.0012 metres. If the slant range is computed using (2), a double floating point square root operation is needed. The FPGA resource cost for floating point processing is heavy and the latency greatly increases with precision. As such, the square root is calculated in fixed-point in this paper.

There are mainly three kinds of square root methods, which are Newton-Raphson, SRT-Redundant and non-restoring techniques. In this paper, a modified non-restoring pipelined architecture is used to optimise the hardware resource usage by taking advantage of the FPGA internal CLB structure, which is optimised for adder realisation and by using the RTL approach directly. In every pipelined stage of the non-restoring algorithm the adder and subtracter can be multiplexed using a complement adder. The hardware resource report lists of the 64 bit input square root processors are shown in Table 4 according to the Xilinx ISE12.4 using FPGA XC4VSX55. It can be seen that the LUTs decrease when comparing processors (T.Sutikno.2011 and S. Samavi 2008).

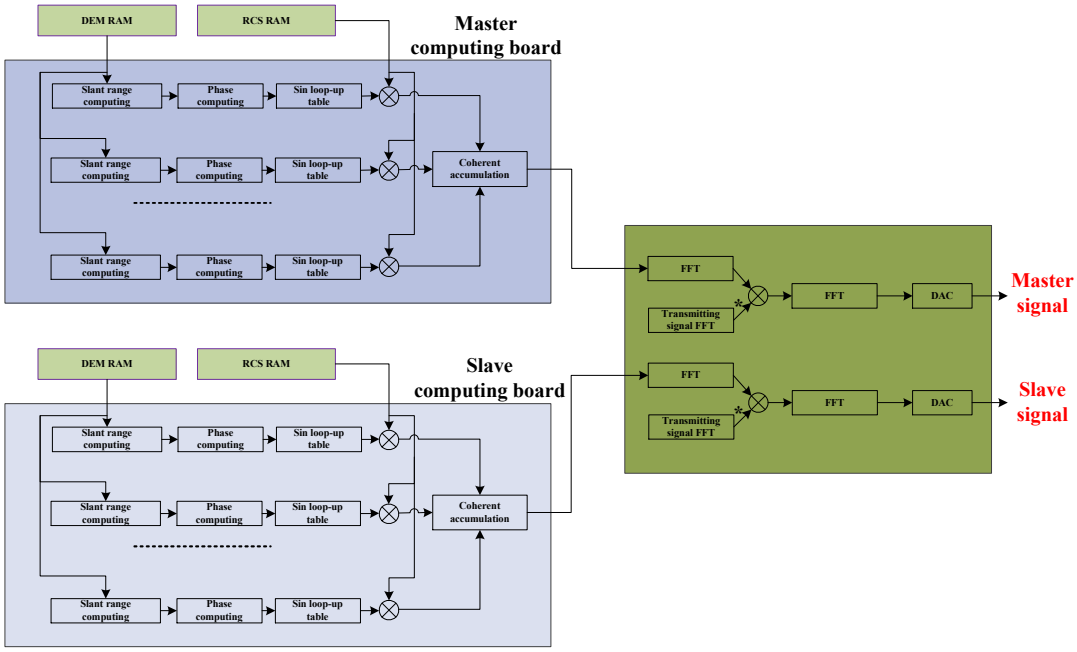


Figure 7: Implementation architecture for raw signal simulation

Table 1 Resource report

Resource Use	Proposed
Slice Flip Flops	1150
Slice LUTs	648
Slices	637

Convolution method

As shown in Figure 8 the signal processed by the main computing modules needs to be convoluted with the transmitted radar signal to generate the raw signal. The convolution can be performed by direct time domain convolution or the fast convolution method using FFT. The time domain convolution method needs lots of resources but has less latency when the operation can be performed in parallel. The resources needed of the latter are reduced, but the latency for the first output is increased. In this paper R2^{2}SDF pipelined FFT processor (S. He et al. 2001) are chosen because of the high speed and medium resource cost. It can process N FFT points in N clock periods. The R22SDF output order is bit-reversed. If the same structure is used for the FFT and IFFT, additional memory is required to reorder the output result of the FFT and the latency is greatly increased. So the DIT and DIF structures are presented for the FFT and IFFT processors respectively, which are shown in Figure 9 and Figure 10. The hardware resources – especially memory– decrease when comparing the convolution processor using only DIF FFT. The resource report list is shown in Table 2 according to the Xilinx ISE12.4 using an FPGA XC5VSX95T.

Table 2: Resource report

Resource Use	used
Slices	2340
Block RAM (18Kb)	24
DSP48E	40

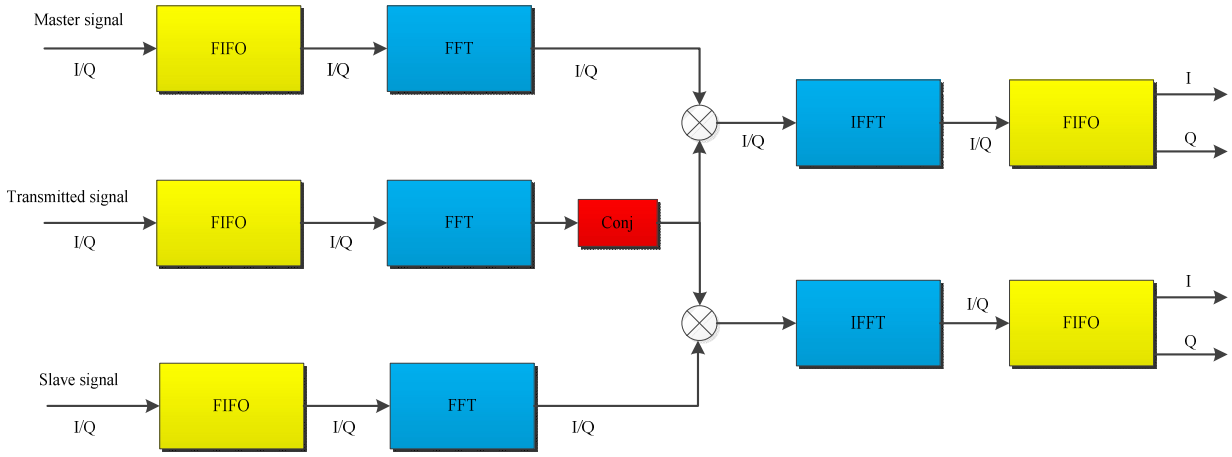


Figure 8: Convolution method

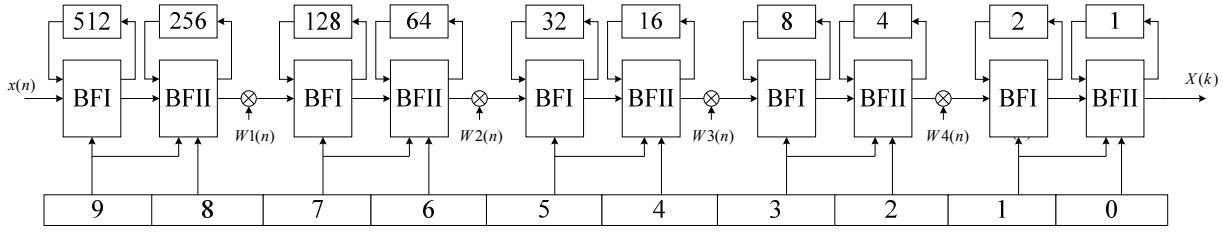


Figure 9: R2^{2}SDF DIF structure

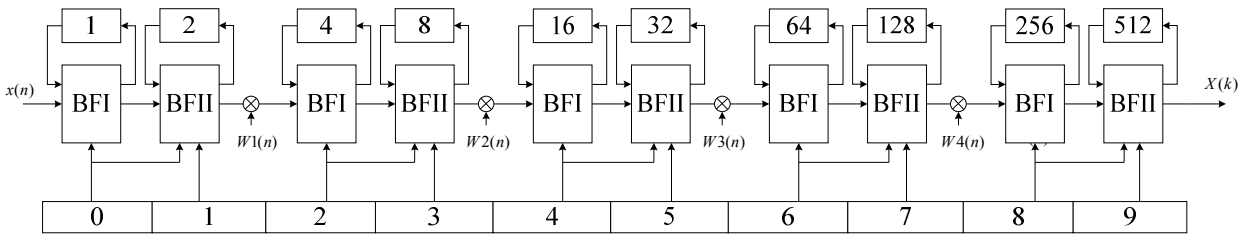


Figure.10: R2^{2}SDF DIT structure

Analogue signal generation

The analogue signal generation is based on FPGA and DAC. The raw SAR analogue signal is generated according to the time radar signal sequence. The PRF pulse signal is a global synchronisation signal for radar transmission and raw signal generation. After the raw signal convolution, the signal data is stored in the FPGA buffer and is generated according to the PRF and slant range centre. In order to generate the analogue signal and compute simultaneously, dual ping-pong rams are used. As shown in Figure 11, when the computing signal data is stored in RAM A, the data from RAM B is read and sent to DAC, while in the following PRF, the order is reversed.

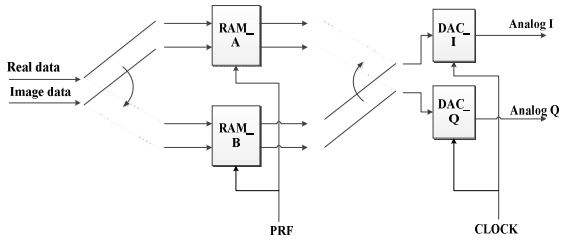


Figure.11: Analogue signal generation structure

FPGA performance

As shown in Figure 6, there are two main kinds of FPGA modules in the system. One is the master/slave computing module the other is the analogue signal generating module. We designed the FPGA modules according to the radar parameters listed in Table 5.

On the computing board, there are four FPGAs for main computing and one FPGA for data summing and communication. Every main computing FPGA has 24 processors which can carry out pipelining tasks, including range computing, phase computing, signal accumulation and memory communication.

We use the Xilinx ISE12.4 for synthesis, place and route. The cost for the main computing FPGA is listed in table iii and the clock speed is 100MHz.

Table 3: Resource report of main computing FPGA

Logic Utilization	Used	Available	Utilization
Registers	35383	58880	60%
Slice LUTs	34358	58880	58%
Block RAM	125	244	49%
Number of DSP48Es	364	640	56%

On the Convolution board, there is one XC5VSX95T for convolution and analogue signal generation, and one XC4VSX55 for communication. The cost for the convolution and analogue signal generating FPGA is listed in table iii and the clock speed is 100MHz.

Table 4: Resource report for the convolution FPGA

Logic Utilization	Used	Available	Utilization
Registers	19471	58880	33%
Slice LUTs	18279	58880	31%
Block RAM	46	244	18%
DSP48Es	66	640	10%

The simulation test is performed with the parameters listed in Table 5. The continuous computing time of every part is below one PRF (Pulse Repetition Time 25us） and needs less than 51.2ms for 2048-pulse simulation.

Table 5: INSAR simulation parameters

Frequency (GHz)	10
Azimuth points	400
Range points	400
Pulse width (μs)	1.5
Bandwidth(MHz)	100

CONCLUSION

In this paper we designed an FPGA-based system for real time INSAR raw signal generating. The simulator is based on the TD algorithm and optimisation methods are presented, such as the slant range computing algorithm, parallel processing architecture and the convolution method. It is designed for UAV INSAR HWIL simulation system which can be used for UAV operator training and system verification. In the future the system will be improved by including real-time backscattering coefficient computing and shadow effects.

REFERENCES

Madsen, S. N., and Zebker, H. A. (1998). Principles and Applications of Imaging Radar. Wiley.

Ian G. Cumming, Frank H. Wong. (2005). Digital Processing of Synthetic Aperture Radar Data. Artech House.

Z. Xujin, Z.Zhaoda. (2007). SAR Echo Simulation Based on Hardware-in-loop. Modern Radar. Vol.29, Sept. 2007

Zheng Xiang, Kaizhi Wang, Xingzhao Liu, Wenxian Yu. (2009). A GPU based Time-domain Raw Signal Simulator for Interferometric SAR. IGARSS 2009, 25-28.

F Zhang, Zheng Li, B Wang, M Xiang and W Hong. (2012). Hybrid general-purpose computation on GPU (GPGPU) and computer graphics synthetic aperture radar simulation for complex scenes. International Journal of Physical Sciences, 7, 1224-1234.

Zhang F, Bai L, Hong W. (2008). INSAR imaging geometry simulation based on computer graphics. ISPRS 2008, 781-784.

Zhang F, Wang BN, Xiang MS. (2010). Accelerating INSAR

raw data simulation on GPU using CUDA. IGARSS 2010, 2932-2935.

Zhihua He, Feng He, Zhen Dong, Diannong Liang. (2012). Real-Time Raw-Signal Simulation Algorithm for INSAR Hardware-in-the-Loop Simulation Applications. IEEE Geosci. Remote Sensing Lett, 9, 134-138.

G. Franceschetti, A. Iodice, M. Migliaccio, and D. Riccio. (1998). A novel across-track SAR interferometry simulator. IEEE Trans. Geosci. Remote Sensing, 36, 950–962.

A. Mori and F. De Vita. (2004). A time-domain raw signal simulator for interferometric SAR. IEEE Trans. On Geosci. Remote Sensing, 42, 1811–1817.

L. Yamin and C. Wanming. (1997). Implementation of Single Precision Floating Point Square Root on FPGAs. IEEE Symposium on FPGA for Custom Computing Machines, Napa, California, USA, 1997, 226-232.

T.Sutikno. (2011). An efficient implementation of the non-restoring square root algorithm in gate level. Internantional Journal of Computer Theory and Engineering, 3, 1793-8201.

S. Samavi, et al. (2008). Modular array structure for non-restoring square root circuit. Journal of Systems Architecture, 54, 957-966.

S. He, M.Torkelson. (2001). Designing pipeline FFT processor for OFDM (de)modulation. ISSSE 2001, 257-262.

L.Wei, W. jun, L. shaohong. (2007). The GPS code acquisition based on pipelined FFT processor. The Second International Conference on Space Information Technology 2007.

AUTHOR BIOGRAPHIES

WEI LI is a researcher on signal processing at the Faculty of Maritime Technology and Operations, Aalesund University College,Norway. Email: weli@hials.no.

HOUXIANG ZHANG is a professor on Robotics and and Cybernetics at the Faculty of Maritime Technology and Operations, Aalesund University College,Norway. Email: hozh@hials.no.

HANS PETTER HILDRE is a professor on product and system design at the Faculty of Maritime Technology and Operations, Aalesund University College,Norway.
Email: hh@hials.no.