UNSYS digital

Journal of Unmanned System Technology

SAR Image Processing Based on the LEON3 Multiprocessor for CP-SAR onboard Microsatellite

Good Fried Panggabean[†‡], Josaphat Tetuko Sri Sumantyo[†], and Bambang Setiadi[↗]

[†]Center for Environmental Remote Sensing, Chiba University, 1-33, Yayoi, Inage, Chiba 263-8522, Japan

[‡]Faculty of Informatics and Electrical Engineering, Del Institute of Technology, Toba Samosir, Indonesia

[↗]Research Center for Geotechnology, Indonesian Institute of Sciences, Bandung, Indonesia

***Abstract*—** The SAR imaging processor is an essential part in SAR system to produce the final SAR image quality that used for various remote sensing applications. In this paper, we propose the design and describe the development of onboard SAR image processor based on LEON3 softcore multiprocessor implemented on FPGA board. The proposed SAR image processor is targetted to used for future onboard Unmanned Aerial Vehicle (UAV) and microsatellite of Josaphat Microwave Remote Sensing Laboratory (JMRSL). The system-on-chip (SoC) approach has been used for highly configurable, easy to integrate with other IP core through an AMBA on-chip bus and low-cost implementation. The open hardware LEON3 multiprocessor has been configured to meet the system requirement to aim at high-performance processing. The implementation of the proposed SAR image processing is targetted on an Altera Cyclone IV FPGA family and Xilinx Artix-7 FPGA AC701 Evaluation Kit. The RDA algorithm is implemented as software to run on the LEON3 Multiprocessor, and the functionality validated using the ALOS PALSAR raw data. The preliminary results show that the functionality of the system successfully works and can compress the SAR raw data into an SAR image, whereas the processing speed needs to be improved. The proposed design promises to answer the problem of SAR image processing in the future with a low price.

***Keywords*—** SAR image processor; LEON3; AMBA; RDA; SoC; Microsatellite

DOI: 10.21535/just.v5i2.988

I. INTRODUCTION

ONE important application of Synthetic Aperture Radar (SAR) is monitoring a natural disaster area because observation can be performed in day-night time and all-weather conditions. Currently, Josaphat Microwave Remote Sensing Laboratory (JMRSL) is developing a Circularly Polarized SAR (CP-SAR) sensor using L, C and X band frequency [1][2]. The CP-SAR sensor plans to deploy on an Unmanned Aerial Vehicle (UAV) for ground experimental calibration and validation. The future, the sensor will use on Boeing 737-200 aircraft for further CP characteristic research and then L band CP-SAR will be utilized on a microsatellite.

The CP-SAR sensor application will be implemented in the various field. For the initial SAR experiment, the research will be focused on the exploration of elliptical scattering wave from the land surface including vegetation, grass, soil, sand, rock, and ice. The axial ratio image (ARI), ellipticity and tilted angle extracted from elliptical scattering wave will be investigated to examine the relationship between the physical characteristics of vegetation, soil, and so on. In the field of SAR application, the CP-SAR sensor will be used for land cover mapping, ocean monitoring, Cryosphere surveillance and disaster monitoring [3].

The design of the CP-SAR sensor on board microsatellite aims is to develop low-cost, lightweight (100 kg class), low-power sensor that configured to transmit and receive left-handed circular polarization (LHCP) and right-handed circular polarization (RHCP) signal. Microsatellite CP-SAR. The sensor antenna is proposed to use a meshed parabolic antenna to meet weight requirements as a 100kg-class microsatellite. The proposed model of Microsatellite CP-SAR is shown in Figure 1. The model consists of a parabolic mesh antenna, GPS-RO and an electronic density temperature probe (EDTP) sensor that will be used to monitor ionospheric phenomena as a submission.

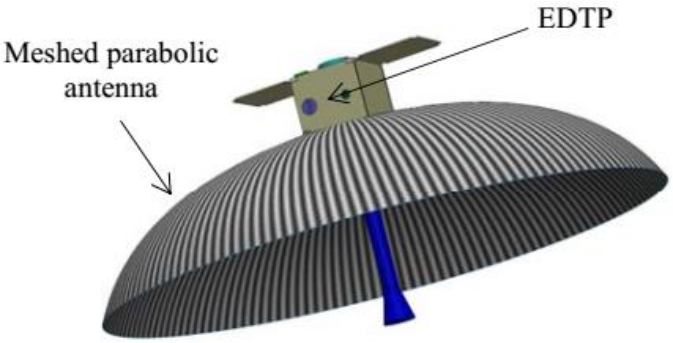

Figure 1 The meshed parabolic antenna for the CP-SAR sensor onboard microsatellite.

Corresponding author: Good Fried Panggabean
(e-mail: goodfp@chiba-u.jp)

This paper was submitted on Oct. 24, 2017; revised; and accepted on Nov 2, 2017.

The Microsatellite CP-SAR will be orbited at altitude in 500-800 km with 97.6° inclination angle and operate at 1.27 GHz, the diameter of meshed parabolic antenna 3.6 m, the beamwidth elevation and azimuth 5° respectively, off-nadir angle 20°-30°, the target ground resolution 30 m and the ground swath width about 45 km. The CP-SAR sensor transmits chirps pulse with PRF 4200 – 6000 Hz, pulse width 20 µs, the bandwidth 15 kHz and peak power output about 1500 Watts. The characteristics of the CP-SAR onboard microsatellite are summarized in Table 1.

Table 1 Comparison of ALOS Palsar [24] and CP-SAR GAIA II Characteristics

Parameters	ALOS PALSAR	Microsatellite CP-SAR
Frequency	1.27 GHz	1.27 GHz
Wavelength	0.24 m	0.24 m
Altitude	691.65 km	500-800 km
Inclination angle	98.16 ^{O}	97.6^{O}
Polarization	HH, HV, VH, VV	LL, LR, RL, RR
Off-nadir angle	9.9 ^{O} – 50.8^{O}	20 ^{O} - 29 ^{O}
Swath Width	70 km (single/dual-pol) 30 km (quad-pol)	45 km
Target reolution	Range = 10 m. Azimuth = 10 m	Range = 30 m Azimuth = 30 m
Antenna width (Elevation)	3.1 m	3.6 m
Antenna length (Azimuth)	8.9 m	3.6 m
Gain Antena	0 - 44 dB	33.1 dB
Pulse Length	27 µs	20 µs
Bandwith (BW)	28 MHz (single-pol) 14 MHz (dual, quad-pol)	15 MHz
Selected PRF	1500 – 2500 Hz	5000 Hz
Peak power	2000 W	1500 W
Platform size		1 m × 1 m × 1 m
Weight	4000 kg	100 kg

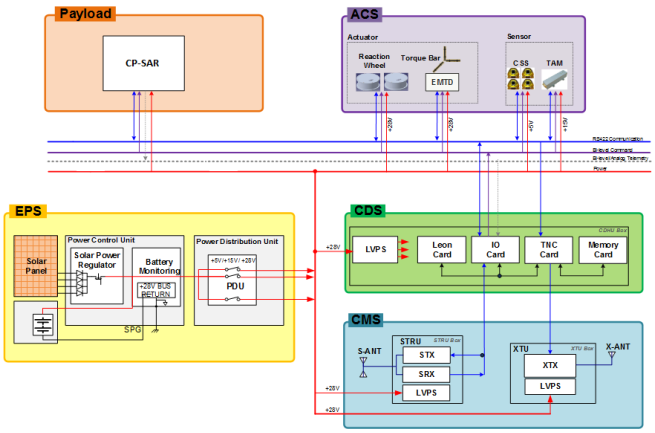

Figure 2 Modules of CP-SAR onboard microsatellite [2].

The modules of CP-SAR onboard microsatellite is shown in Figure 2. The modules consist of five main sub-modules: payload (CP-SAR sensor and EDTP), Altitude Control Subsystem (ACS), Communication Subsystem (CMS), Electrical Power Subsystem (EPS), and Command and Data Handling Subsystem (CDS). ACS that provides altitude information and maintains satellite altitude during the mission is composed of LVPS, actuators, sensors (coarse sun sensor-CSS and three-axis magnetometer-TAM) and GPS Receiver-GPSR. Telemetry, tracking and command system supported by CMS subsystem that composes of S-band transmitter, receiver, and antenna for telemetry, X-band transmitter, and an antenna for data downlink. EPS consists of solar panel, solar power regulator, power supply card, power distribution card, Li-Ion battery pack and battery control card. The CDS is the brains of the satellite that control orbiter and the whole of satellite function. The module employs an Onboard Computer (OBS) using LEON3 processor, memory card, payload IO card, IO card and low power supply card (LVPS). The CDS subsystem is shown in Figure 3(a) and the single board computer for space application with the triple modular redundant of the LEON3 processor is shown in Figure 3(b).

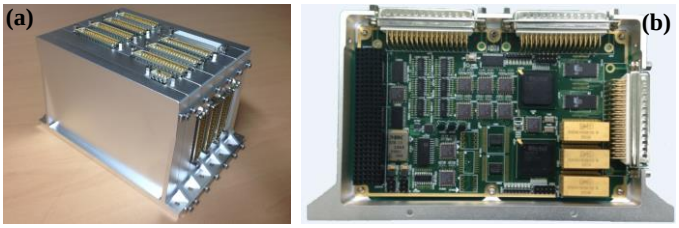

Figure 3: (a) CDS Subsystem, (b) LEON3 card

The CP-SAR sensor onboard microsatellite will be used for land surface and disaster monitoring. The experimental CP Differential Interferometry SAR (CP-DInSAR) will be applied to observe earthquake area, forest fire, peatland fires, volcano activity, landslide and flood monitoring. For disaster monitoring, it is necessary to distribute the observation data rapidly from disaster area to the disaster management center. The data is needed to estimate the damage of area and support the rescue operations. With performing SAR image processing onboard is a solution to reduce the time of distribution thus image area observed can be directly transmitted to the ground station.

The SAR image processor has an important role in SAR system to produce the real or near real time SAR images. It is the highly computational application that implemented on wide computing technology. The High-Performance Computing (HPC) in grid environments is used to implement the high-performance computing platforms with parallel methods [4]. Collaboratives computing using multiple central processing units (CPUs) and GPUs to accelerating processing is explored extensively at [5]-[8]. Near real-time SAR processing using general purpose CPU [9] and onboard SAR image processing uses DSP or a FPGA system [10],[11].

SoC (System on Chip) is proliferating and has been widely used in industrial applications. The design of SoC-based platforms offers the rapid system development, flexibility and reprogrammable with low cost [12]. With SoC design, the whole system can be developed on a single device and realize the system on a single FPGA device. The LEON3 is a 32-bit soft processor compliant with the SPARC V8 architecture suitable for SoC design. The LEON3 processor usages in aerospace applications are currently widely used in European space programs.

The implementation of the proposed SAR image processor based on the LEON3 multiprocessor platform is described in this paper. The main contribution of this paper is to present hardware and software solutions based on the LEON3 multiprocessor for the SAR image processor. The remainder of the paper organize as follow, section II describes some background on the LEON3 processor and the RDA algorithm. Section I introduces the proposed design of SAR imaging processor based on the LEON3 multiprocessor. Implementation and experiment result is presented in section IV. Due to the unavailability of raw data of CP-SAR, the experiment was conducted using ALOS PALSAR data which has similar characteristics with Microsatellite CP-SAR. Finally, the conclusion is left to the section V.

II. LEON3 Processor and The RDA Algorithm

A. *LEON3 Processor*

The LEON3 is a synthesizable VHDL model of a soft processor that highly configurable, and particularly suitable for system-on-a-chip (SoC) designs [13],[14]. The LEON3 processor is distributed as part of the GRLIB IP library developed by Gaisler Research. The GRLIB IP library provides the template design that can be used to implement a LEON3 system for various FPGA boards. The LEON3 processor and the other IP cores will be connected through an on-chip bus AMBA-2.0 AHB/APB as shown in Figure 4.

The Advanced Microcontroller Bus Architecture (AMBA) is an open standard bus designed to the high-performance embedded system [15]. The Advanced High-performance Bus (AHB) is intended for high-performance, high-bandwidth operation and act as backbone bus for connecting the LEON3 processor with on-chip memory and off-chip external memory interface. The bus supports up to 256-bits data transfer and support for multiple bus masters.

The Advanced Peripheral Bus (APB) is designed as a local secondary bus for low bandwidth control accesses and optimized for low power consumption. This bus is intended as an interface to peripherals that do not require high performance pipelined bus such as UART JTAG, timers, interrupt controller and I/O port. The AHB/APB Bridge is provided to transform AHB transfers into the appropriate format for the slave devices on the APB.

B. *The Range-Doppler algorithm*

The SAR image processing is a program that converts the SAR data into single look complex (SLC) image. The SAR processing will compress the reflected signal spreads in the range and azimuth direction into a signal pixel to construct the target that illuminated by SAR sensor. The Range-Doppler Algorithm (RDA) is an algorithm that commonly used for processing SAR data into an image [16],[17]. There are several important SAR image and well-known in addition to RDA those are the Chirp Scaling Algorithm (CSA) [18], the Omega-K Algorithm (ωKA) [19] and Spectral Analysis SAR Algorithm (SPECAN) [20].

The RDA algorithm was initially used in 1978 to process spaceborne SEASAT SAR image data. The efficiency and simplicity of the block processing algorithm are achieved by performing operations on the frequency domain in both range and azimuth furthermore maintaining a one-dimensional operation [21]. The Range Cell Migration Correction (RCMC) operation is effected in the range time and azimuth frequency domain or Doppler domain range that additionally accommodates block processing efficiency. The implementation efficiency of RDA algorithm can be obtained because all matched filter convolution operations are performed as multiplies in the frequency domain.

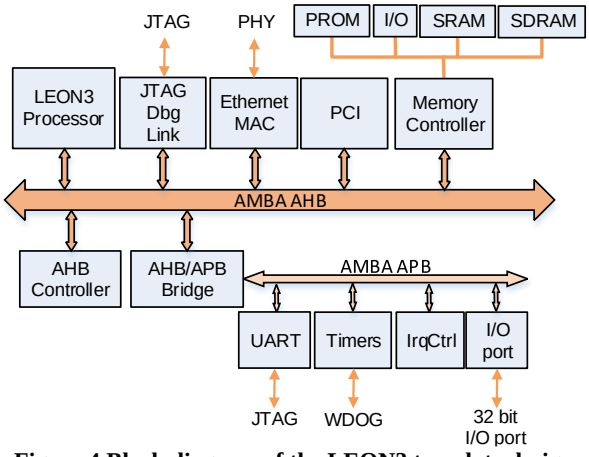

Figure 4 Block diagram of the LEON3 template design

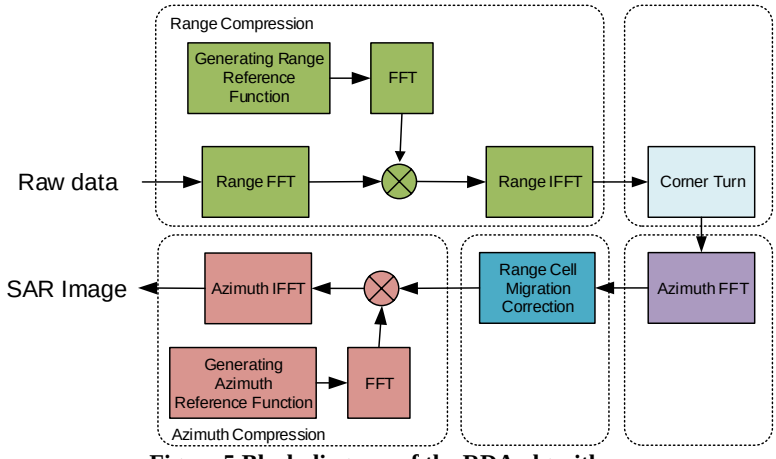

Figure 5 Block diagram of the RDA algorithm

The block diagram of the RDA algorithm is illustrated in Figure 5. There are five significant steps in the RD algorithm namely range compression, corner turn, azimuth FFT, range cell migration correction, and azimuth compression. The range compression is performed the matched filter multiply on every row of range FFT raw SAR data and the subsequent operation of range IFFT. The range matched filter work as a range reference function and are performed on each row of raw data during range compression process. The corner turn operation is needed to convert the row-based of the range compressed data to become column-based data for the following the step of azimuth FFT. The RCMC is performed in the range-Doppler domain where a group of the trajectory at the same range are converted into a single trajectory utilizing interpolation operation. The final step is azimuth compression performed azimuth matched filtering multiply at each range gate followed by inverse azimuth FFT.

The raw data is the modulated baseband data received from the radar system. The modulated radar signal received from a point target can be expressed as (1).

$$s_0(\tau,\eta) = A_0\omega_r\left[\tau - \frac{2R(\eta)}{c}\right]\omega_a(\eta-\eta_c)e^{\left\{-j\frac{4\pi f_0 R(\eta)}{c}\right\}}e^{\left\{j\pi K_r\left(\tau-\frac{2R(\eta)}{c}\right)^2\right\}} \quad (1)$$

where A_0 is an arbitrary complex constant, τ is range constant, η is azimuth time referenced to closest approach, η_C is beam center offset time, ω_r is range envelope, ω_a is azimuth envelope, f_0 is radar center frequency, and K_r is range chirp FM rate. The $R(\eta)$ is instantaneous slant range is defined as $R(\eta) = \sqrt{R_0^2 + V_r^2\eta^2}$, where R_0 is the closest slant range to the target and V_r is the effective radar velocity.

The range compression performed by applying a range Fourier transform to $s_0(\tau,\eta)$ and multiply with the frequency domain the matched filter. The output of the range matched filter can be expressed as (2).

$$s_{rc}(\tau,\eta) = IFFT_\tau\{S_0(f_\tau,\eta)G(f_\tau)\} \quad (2)$$

where $G(f_\tau) = exp\left(-\frac{j\pi f_\tau^2}{K_\tau}\right)$ is the matched filter in the frequency domain. The azimuth FFT is performed transformation for each range compressed data into range-Doppler domain as follow

$$S_1(\tau, f_\eta) = FFT_\eta\{s_{rc}(\tau,\eta)\} \quad (3)$$

The next step of RDA algorithm is RCMC. The RCMC is applied using the range interpolation operation in the range-Doppler domain. The RCMC operation is defined as (4)

$$S_2(\tau, f_\eta) = RCMC\{S_1(\tau, f_\eta)\} \quad (4)$$

where he amount of range to be corrected is expressed as (5).

$$\Delta R(f_\eta) = \frac{\lambda^2 R_0 f_\eta^2}{8V_r^2} \quad (5)$$

The final step of RDA algorithm is azimuth compression. The data after RCMC are multiplied with the azimuth matched filter in frequency domain as (6).

$$S_3(\tau, f_\eta) = S_2(\tau, f_\eta)H_{az}(f_\eta) \quad (6)$$

where azimuth matched filter $H_{az}(f_\eta)$ is defined as (7).

$$H_{az}(f_\eta) = e^{\left(-j\pi\frac{f_\eta^2}{K_a}\right)} \quad (7)$$

where K_a is the azimuth FM rate as a function of R_0 that defined as $K_a \approx \frac{2V_r^2}{\lambda R_0}$. The final result of azimuth compression is the inverse Fourier transform of S_3 can be expressed as (8)

$$\begin{aligned} s_{ac}(\tau,\eta) &= IFFT\{S_3(\tau, f_\eta)\} \\ &= A_0 p_r\left(\tau - \frac{2R_0}{c}\right)p_a(\eta)e^{\left(-j\frac{4\pi f_0 R_0}{c}\right)}e^{(j2\pi f_{\eta c}\eta)} \end{aligned} \quad (8)$$

where p_a is the amplitude of the azimuth impulse response.

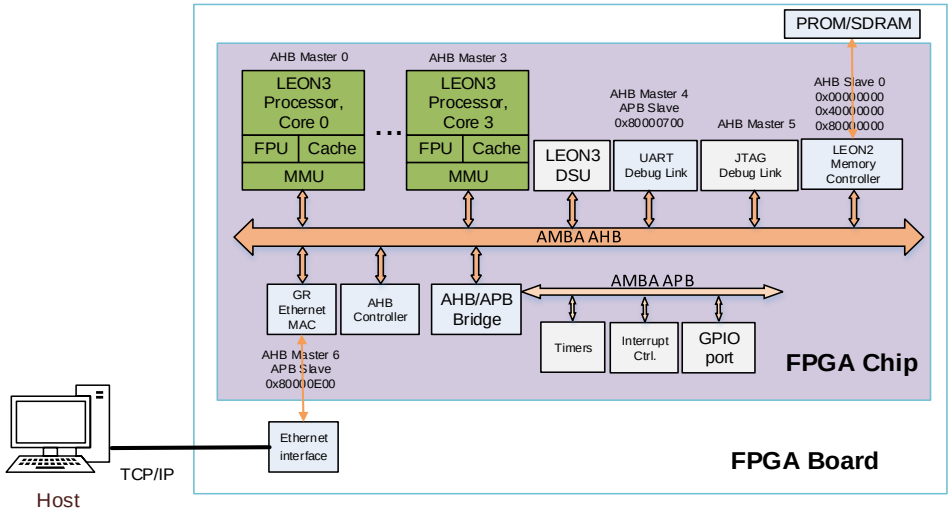

Figure 6 Multicore LEON3 processor architecture for the proposed SAR image processor

III. Implementation of The Proposed SAR Image Processor

This section presents the hardware and software implementation of the proposed SAR image processor based on the LEON3 processor and parallelization approach to utilize multi-processing capabilities.

A. The LEON3 Processor Configuration

In this section is described the hardware implementation of the proposed design based on the LEON3 processor. The proposed SAR image processor architecture using multicore LEON3 processor is shown in Figure 6. The multicore LEON3 processors connect to the Universal Asynchronous Receiver-Transmitter (UART) and the Joint Test Action Group (JTAG) debug link, debug support unit (DSU), LEON2 memory controller for interfacing with SRAM and SDRAM and Gaisler Research (GR) Ethernet MAC through the AHB bus. Several peripheral devices such as General-Purpose Input/Output (GPIO), UART, timers and interrupt controller is connected to the LEON3 processor via APB bus.

The LEON3 processor is customized to aim the high-performance multicore SoC system. The LEON3 template design that available on GRLIB open-source version is configured using a graphical tool to satisfy the requirements. The customizations include using a level-2 cache to reduce the effects of memory latency, using the high-performance GRFPU floating-point unit to provide floating-point operation, configuring to support for SPARC V8 MUL/DIV instructions using a 32×32 pipelined multiplier and enabling Memory Management Unit (MMU).

The FPGA board is connected to a personal computer (PC) through a UTP cable. This cable is used to transmit segment of raw data from the PC to the LEON3 processor and send back the segment of image result from the LEON3 processor to the PC. The communication protocol used between the PC and the LEON3 processor is the TCP/IP protocol. The PC and the FPGA board are also connected using the USB port that functioned as the JTAG to download the executable binary code from the host computer to the LEON3 processor and to debug the program running on the board (Figure 7). The host computer in the system functioned as flow and data control and storing the SAR raw data and the image processing result.

The target implementation of the proposed SAR image processor is based on two types of FPGA boards, the Altera Cyclone IV DE2-115 (DE2) and Xilinx Artix-7 FPGA Evaluation Kit (AC701). The configuration and system specification used in this experiment is shown in Table 2. The various values of clock multiplication factor were attempted to obtain the clock frequency of the LEON3 processor in the range of 50 to 75 MHz. The number of cores implemented on DE2 board is three cores due to resource limitations while AC701 can be implemented up to 4 cores.

The LEON3 multiprocessors run the eCos operating system. It is the real-time operating system that supports symmetric multi-processing (SMP). The SMP model needs only one operating system running and controlling the whole of the processor. The main advantage of this model is the ability to distribute multi-threaded applications on the available processor's node automatically. Others operating system is Real-Time Executive for Multiprocessor Systems (RTEMS) broadly used in space applications. It works using the Asymmetric Multiprocessing (AMP) model that runs separate operating systems on each processor. Every processor executes its image and communicates with another processor by message passing via shared memory, From a software development standpoint, software partitioning is required to fit the underlying hardware architecture. In this case, the SMP model is chosen in a multi-processor system, and the eCos operating system is used to utilize multi-processing capabilities.

The Altera Quartus II Web Edition 14.1 tools are used to synthesize the design of the LEON3 processor with the customized hardware components for DE2 board and Xilinx Vivado 2015.4 for AC701 board. Table 3 and Table 4 show the occupancy of the hardware resource that used to implement the LEON3 multi-processor for DE2 and AX701. We can see that the system use only a few of hardware resources. The system uses only 30% of the total available logic elements for a single core.

Table 2 The LEON3 processor configuration and system specifications

Parameter	Values	
	DE2	AC701
Clock Frequency	50 MHz – 75 MHz	100 MHz
Integer Unit	Yes	Yes
Floating Point Unit	Yes	Yes
RAM	SDRAM 128M	DDR3 1 GB
Cores Maximum	3	4
Ethernet	Gaisler Research 100 Mbps	
Compiler	Ver. 21-Jan-2015 from Gaisler	
Operating System	RTEMS Ver 4.10 from Gaisler	
	Ecos Ver 1.0.8 from Gaisler	

Table 3 Percentage of DE2 hardware resource utilization

Resources	Number of Cores		
	1	2	3
Logic Elements	34.21	61.74	99.48
Memory bits	7.01	19.34	37.73
Emb. Mul. 9-bit	7.33	14.66	29.32
PLLs	25	25	25

Table 4 Percentage of AC701 hardware resource utilization

Resources	Number of Cores			
	1	2	3	4
LUT	31.71	49.76	67.91	85.56
LUTRAM	3.34	3.63	3.93	4.23
FF	9.36	13.07	16.91	20.45
BRAM	8.90	16.03	23.15	30.27
DSP	0.54	1.08	1.62	2.16
IO	39.75	39.75	39.75	39.75
BUFG	21.88	21.88	21.88	21.88
MMCM	10.00	10.00	10.00	10.00
PLL	30.00	30.00	30.00	30.00

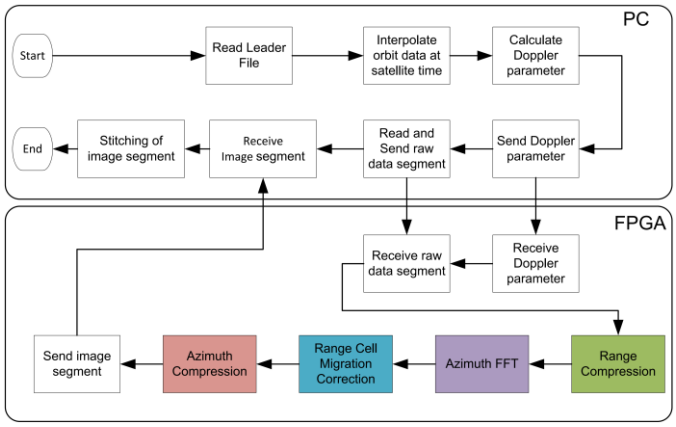

Figure 7 Flow of implemented software

A. Implementation of the Range-Doppler algorithm

In this section, the implementation of software for SAR Image processor is presented. The flowchart of the implemented software is illustrated in Figure 7. The software consists of two parts. The first part of the software is the application that runs on the PC and the second part is the implementation of the range-Doppler algorithm was ported to the LEON3 processor. The application running on the host computer is functioned as data flow control. It will initiate the system to start the processing and sends the SAR parameters to read from the leader file. Then the application sends each segment of data to be processed and receive the result. The SAR parameters are needed for the RDA computation.

In Figure 7, the range-Doppler algorithm starts with generating a range reference function using the parameters received from the host computer. The raw data in range direction transformed to frequency domain using FFT and then multiplied with the range reference function. The range compressed data is transformed back to range time domain. Azimuth compression is done by multiplying the range compressed data in azimuth direction with azimuth reference function and then applying the inverse FFT.

In this implementation, we need to consider the size of the segment in the range and azimuth direction due to the limitations of the FPGA memory. There is a trade-off in the selection of segment size because a segment processed overlapped with the previous segment and the next segment. If the segment size is reduced, the number of segments to be processed will increase, and the overlapping samples will increase but if the segment size is enlarged demand for memory increases. For example, the size of the raw data for ALOS PALSAR at area Mt. Fuji contains 10344 pixels and 35421 lines. If the segment size used is 8192 pixels × 16384 lines, the whole image must be partitioned into six segments (see Figure 6). There are 915 pixels overlap in range direction, and 5823 pixels overlap in the azimuth direction. It takes about 1024 MB of memory to store one segment of data (I and Q) when using 32-bit floating point data type. The data is stored using half-precision floating point format (16-bit) to reduce memory consumption.

The range-Doppler algorithm implemented in c language has been ported to the LEON3 processor. The first, we need to modify configuration file and configured driver manager to defining the appropriate drivers. The next work, the application needs to cross-compile and link the program using sparc-rtems-gcc to produce an executable binary code. Then, the executable binary code is downloaded to the LEON3 processor using a USB port that functioned as JTAG. The JTAG link is also used to debug the program running on the board as shown in Figure 8.

B. Parallelization Approach

Several considerations must be considered when implementing parallel processing. First of all, it must be ensured that the processing load can be distributed proportionally in each core so that the workload evenly on each core, this way parallel program performance can be improved completely. The second consideration is the partition granularity rational program to achieve a high degree of parallelism. A final consideration is scalability and resource availability. The first issue to be resolved is the partition granularity programs and ensuring balanced workloads when processed in multicore parallel.

Two parallelization approaches to utilize multiprocessor can be proposed. The first method, parallelization conduct at the segment level of range/azimuth compression as shown in Figure 8. Each processor compresses every segment parallelly and computes every step of the RDA algorithm sequentially. This approach consumes much memory because each segment processed parallelly must be allocated to a different memory location. Moreover, each segment independent with other segments of the line of segment and partly overlaps with the other segment as needed by the algorithm. The second approach conduct at the line level of the range and azimuth processing. A data segment is divided into sub-segment data and processed by each processor parallelly. For example, a segment with size 8192×8192, processor core one will perform the range/azimuth compression for line 1 up to 4096 and the core 2 for line 4097 up to 8192. This approach consumes less memory because only one segment should be allocated to the memory.

Figure 8 Debugging LEON3 processor

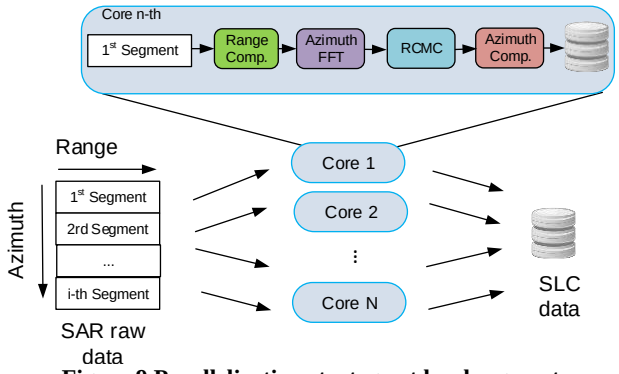

Figure 9 Parallelization strategy at level segment

The number of parallel processes depends on the number of cores available. Due to resource constraints, only three processors can be implemented on the DE3 board and four processors on the AC701 board.

IV. Results and Discussion

In this section, we have implemented the proposed designs based on LEON3 processor. Performance metrics measured is the time of execution. First, we have conducted a testing to obtain FFT execution time because RDA uses a lot of FFT operations. Figure 10 shows the execution time of FFT for a different complex number of samples. The experiments were conducted to distinct configurations of LEON3 processor clock frequency. The FFT execution time for the 1024 point complex numbers requires about 28 milliseconds (compare if using FFT IP core requires only 52 µs [8]).

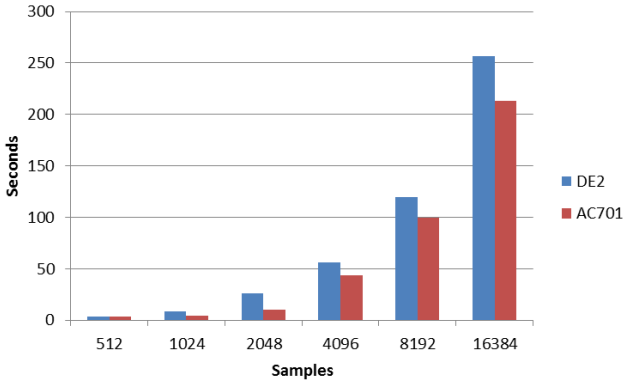

Figure 10 FFT Average Execution Times (milliseconds) of distinct number of complex samples

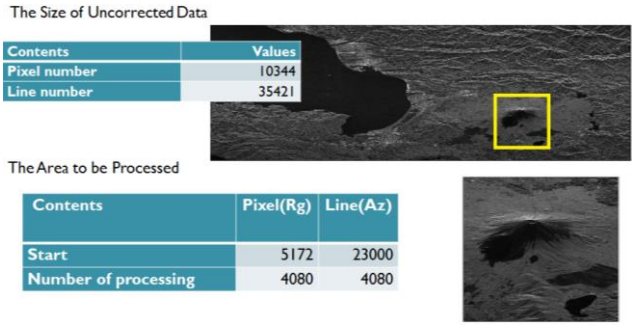

Figure 11 PALSAR Sample picture taken from [22]

Table 5 Scene Parameters

Parameter	Value
Satellite	ALOS PALSAR
Frequency	1.27 GHz/23.6 cm
Polarization	HH
Observation Mode	Very high-resolution mode 1 wave (FBS)
Observation Area	Mt.Fuji
Observation date	2008/11/18
Off-nadir angle	34.30^{O}
PRF	2159.83 Hz
Pulse length	27 µs
Sampling Rate	32 Mhz
Sampling Window Start Time	5.651683×10^{-3} s
Satellite time	4.755617×10^{4} s
Total scene size	20688 x 35421 (range x azimuth)

The second test was conducted in this experiment using PALSAR sample picture with very high-resolution observation mode (FBS) at area Mt. Fuji. The scene parameters are shown in Table 5. The raw data level 1.0 is processed with size 4536 pixels (range) × 4728 lines (azimuth) as shown in Figure 11. The processing of the complete image (4536×4728) at a time is not possible due to memory limitation. For the fine process, FFT length using 2048 points for range direction and 8192 points for azimuth direction. If complex numbers are stored with 16-bit half-precision floating point format, then one segment requires 64 MB of memory. The whole data image is partitioned into eight segments (4 patches in the range direction and two patches in the azimuth direction) and will be processed segment by segment.

The average execution times for each segment for various clock frequencies are shown in Figure 12(d). By increasing the LEON3 processor clock frequency of 5MHz will decrease the execution time of about 5%. Parallelization performance is benchmarked by measuring the computation speed-up and the parallelization efficiency of the proposed SAR image processor. The speedup is defined as the ratio of the processor latency single LEON3 processor as compared to the processor latency on N processors. Figure 12(a) shows the execution time of 1, 2, 3, and 4 processors implementation for DE2 and AC701 boards. As shown in Figure 12(b), the computation speedup almost doubled by doubling the number of cores for DE2. For AC701 board only speedup 1.79 times for 2 cores, 1.93 times for 3 cores and 2.18 times for 4 cores. The highest efficiency is reached about 99% for DE2 and 81% for AC701 when running on two processors.

The estimated power consumption measured with Xilinx Power Estimator for the entire system are 3.73 W, 4.215 W, 4.661 W, and 5.092 W respectively for 1,2,3 and 4 core processors. For the single core, the LEON3 processor consumes 465 mW. The most consuming components are the DDR2 memory controller (1.727 W), the clock generators (138 mW) and GR Ethernet MAC (105 mW).

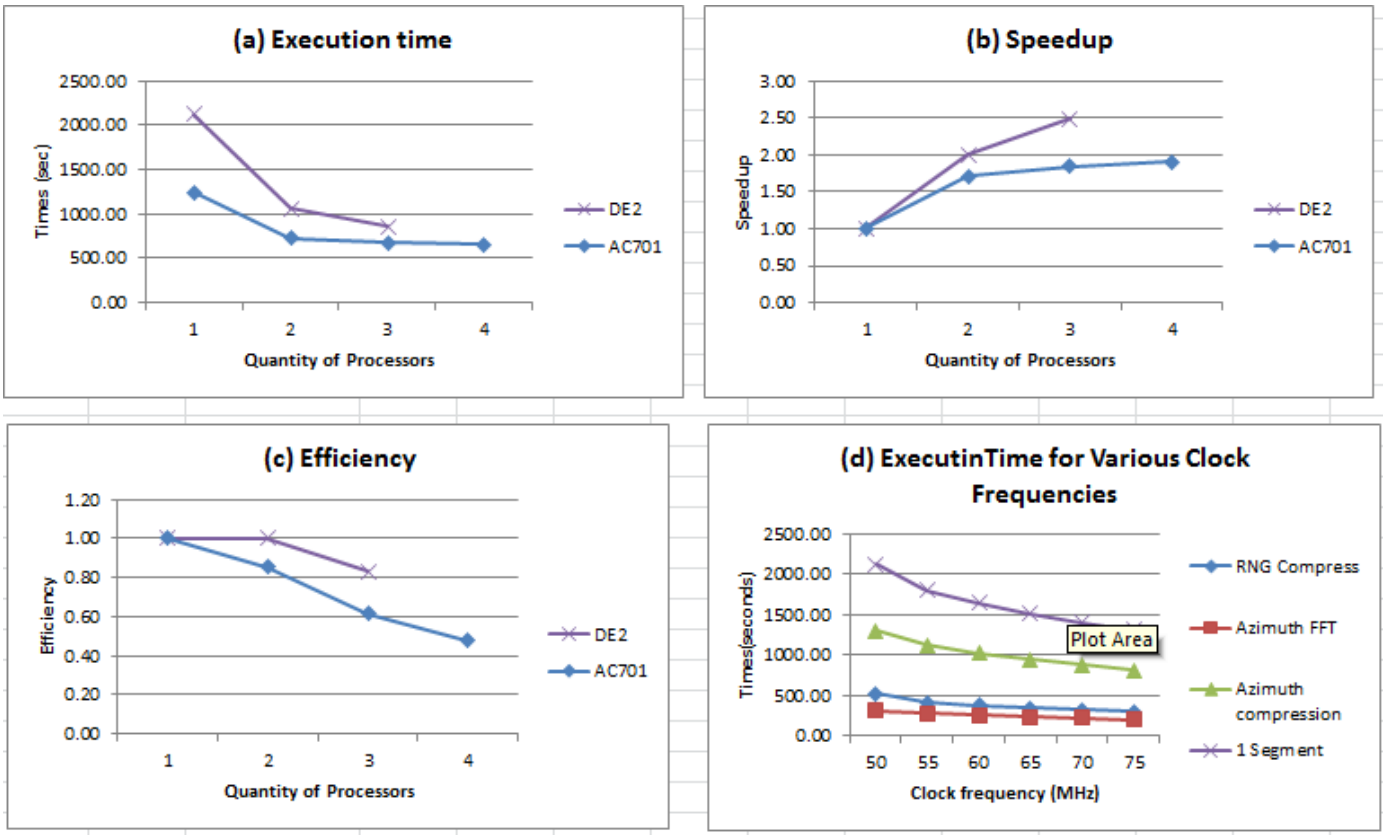

Figure 12 Average execution times

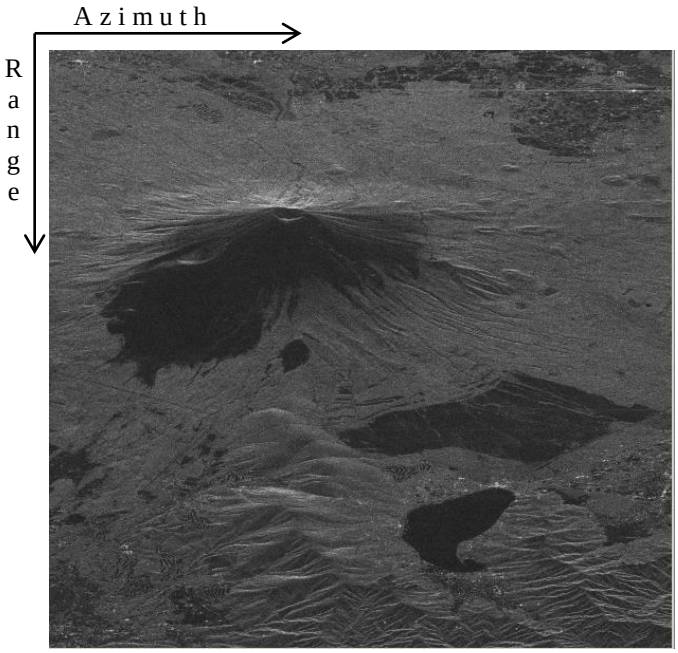

Figure 13 Image result with size 4536 pixels (range) × 4728 lines (azimuth)

V. Conclusion

In this paper, we introduce the implementation of the proposed SAR image processor based on the LEON3 multiprocessor using ALTERA DE2-115 and Xilinx Artix-7 FPGA AC701 Evaluation Kit FPGA. The RDA algorithm is implemented to utilize parallel processing. The experimental results are showed that functionally, the system could process SAR raw data into an SAR image although processing speed needs to be improved. The proposed SAR image processor is high potential for the SAR processing on-board microsatellite. The future work we will implement FFT IP core and designed LEON3 multiprocessor to increase the system performance.

VI. Acknowledgements

This research was supported by Chiba University Strategic Priority Research Promotion Program FY2016-FY2018. We also thank to the Del Foundation, Tanoto Foundation and Citramas Foundation for funding of the study.

VII. REFERENCES

[1] Sumantyo, J.T.S., "Progress On Development Of Synthetic Aperture Radar Onboard Uav And Microsatellite", Geoscience and Remote Sensing Symposium (IGARSS), IEEE International, 2014.

[2] Josaphat Tetuko Sri Sumantyo; Nobuyoshi Imura," Development of circularly polarized synthetic aperture radar for aircraft and microsatellite", IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016.

[3] Sri Sumantyo, J. T. Circularly Polarized Synthetic Aperture Radar Onboard Unmanned Aerial Vehicle (CP-SAR UAV). In Autonomous Control Systems and Vehicles; Nonami, K.; Kartidjo, M.; Yoon, K.-J.; Budiyono, A., Eds.; Springer Japan; pp. 175–192, 2013.

[4] M. Cafaro, I. Epicoco and S. Fiore et al. Near real-time parallel processing and advanced data management of SAR images in grid environments, Journal of Real-Time Image Processing, 4(2009), 219-227.

[5] C. Clemente, M. di Bisceglie, M. Di Santo, N. Ranaldo, and M. Spinelli, ''Processing of synthetic aperture radar data with GPGPU,'' in Proc. IEEE Workshop Signal Process. Syst., Tampere, Finland, Oct. 2009, pp. 309–314.

[6] Bambang Setiadi, Good Fried Panggabean, Josaphat Tetuko Sri Sumantyo, and Koo Voon Chet, "Development of raw data processing system for JX-2 UAV using mobile heterogenenous computing," Journal of Unmanned System Technology.

[7] Di Bisceglie, M., M. Di Santo, C. Galdi, R. Lanari, and N. Ranaldo, "Synthetic aperture radar processing with GPGPU", IEEE Signal Proc. Mag., Vol. 27, No. 2, 69–78, Sept. 2010.

[8] Di Bisceglie, M., M. Di Santo, C. Galdi, R. Lanari, and N. Ranaldo, "Synthetic aperture radar processing with GPGPU", IEEE Signal Proc. Mag., Vol. 27, No. 2, 69–78, Sept. 2010.

[9] M. Cafaro, I. Epicoco and S. Fiore et al. Near real-time parallel processing and advanced data management of SAR images in grid environments, Journal of Real-Time Image Processing, 4(2009), 219-227.

[10] C.-M. Huang, C.-M. Wu, C.-C. Yang, S.-L. Chen, C.-S. Chen, J.-J. Wang. K.-J. Lee, C.-L. Wey, "Programmable System-on-Chip for Silicon Prototyping", Industrial Electronics, IEEE Transactions, Vol 58, Issue 3, pp. 830-838, March 2011.

[11] Xin Xiao, Rui Zhang, Xiaobo Yang and Gang Zhang, "Realization of SAR real-time processor by FPGA", IEEE International Geoscience and Remote Sensing Symposium, vol.6, 2004, pp. 3942-3944.

[12] C.-M. Huang, C.-M. Wu, C.-C. Yang, S.-L. Chen, C.-S. Chen, J.-J. Wang. K.-J. Lee, C.-L. Wey, "Programmable System-on-Chip for Silicon Prototyping", Industrial Electronics, IEEE Transactions, Vol 58, Issue 3, pp. 830-838, March 2011.

[13] Gaisler Aeroflex:GRLIB IP Library User's Manual, version 1.3.7, 2014.

[14] Gaisler Aeroflex: GRLIB IP Core User's Manual, version 1.3.7, 2014.

[15] ARM, AMBA Specification. Rev. 2.0,1999.

[16] Cumming and J. R. Bennett, "Digital processing of SEASAT SAR Data," in Proc. IEEE Int. Con$ Acoust., Speech, Signal Processing, WA, 1979, pp. 4547.

[17] A. M. SMITH, "A new approach to range-Doppler SAR processing", International Journal of Remote Sensing, Volume 12, Issue 2, 1991.

[18] Raney, R.K., Runge, H., Bamler, R., Cumming, I.G., Wong, F.H., Precision SAR processing using chirp scaling. IEEE Transactions on Geoscience and Remote Sensing 32 (4),786 – 799, 1994.

[19] L.M.H. Ulander and H. Hellsten, "System Analysis of Ultra-Wideband VHF SAR," RADAR'97, Edinburgh, UK, Included in conference publication no. 449, pp. 104-108, IEE, London, October 1997.

[20] A.A. Thompson, J.C. Curlander, N.S. McLagan, T.E. Feather, M. D'Iorio, J. Lam, "ScanSAR Processing Using the Fastscan System," Proc. IGARSS '94. pp. 1187-1 189. 1994.

[21] Cumming, Ian G," Digital Processing of Synthetic Aperture Radar Data: Algorithms and Implementation", Artech House,2005.

[22] Remote Sensing Technology Center: PALSAR Sample Picture, http://www.alos-restec.jp/en/staticpages/index.php/service-sampledata-03, 2009.

[23] Altera, "FFT MegaCore Function User Guide", Version 1.02, March 2001.

[24] Rosenqvist, A; Shimada; Watanabe, M ALOS PALSAR: Technical outline and mission concepts, 4th International Symposium on Retrieval of Bio and Geophysical Parameters from SAR Data for Land ApplicationsInnsbruck, Austria, November 16-19, 2004.