RDA사용 위성기반 SARP 주요설계기법

정회원 홍 인 표*

A Critical Design Method of the Space-Based SARP Using RDA

In Pyo Hong* ***Regular Member***

ABSTRACT

The design method of synthetic aperture radar processor (SARP) in the critical design stage is to describe the processing algorithm, to estimate the fractional errors, and to set out the software (SW) and hardware (HW) mapping. The previous design methods for SARP are complex and depend on HW. Therefore, this paper proposes a critical design method that is of more general and independent of HW. This methodology can be applied for developing the space-based SARP using range-Doppler algorithm (RDA).

Key Words : critical design, SAR processing, fractional error, SW and HW mapping, design sequence, SARP, RDA

Ⅰ. Introduction

The generation of synthetic aperture radar (SAR) images from the raw radar data is computationally demanding, and depends on the radar platform and the chosen processing parameters[1]. The SARP is the primary image-generating component of the SAR system.

Generally, there are three design stages for designing SARP: conceptual, preliminary, and critical design stage in sequence. This paper focuses on the critical design stage. The design sequence of SARP is illustrated in Fig. 1. We assume that the parameters and requirements of the SAR system and SARP have been already selected and determined in the preceding design stages: conceptual and preliminary design stages. These are used as the input data of the critical design process for the SARP. Work during the preceding design stages set out the preliminary functional definitions of the SAR processing algorithm. These definitions constitute the starting point of this paper. There are three design steps to be done in the critical design stage: These are to describe the processing algorithm, to analyze the fractional errors, and to map SW and HW of the SARP by using processing parameters. Each step of critical design stage is described at one of the following sections.

Also, it is assumed a space-based SAR system using range-Doppler algorithm (RDA) for simulation and called the experimental-SAR (EX-SAR), which is not a real but designed system based on the SAR system design criteria[2, 3]. The mission of the EX-SAR is the scientific research and observation of earth properties. We choose X-band for EX-SAR because X-band SAR has some advantages such as high resolution, good capability for separating forest from snow, etc. The simulation to the EX-SAR processor (ESARP) is performed to prove the practicability and simplicity of the method. Estimations for the magnitude of any processing errors are summarized, and their impact on performance is assessed. Processing load and rate is estimated, input and output(I/O)

* 국방과학연구소 (hip7777@naver.com)
논문번호: KICS2005-09-378, 접수일자 2005년 9월 19일

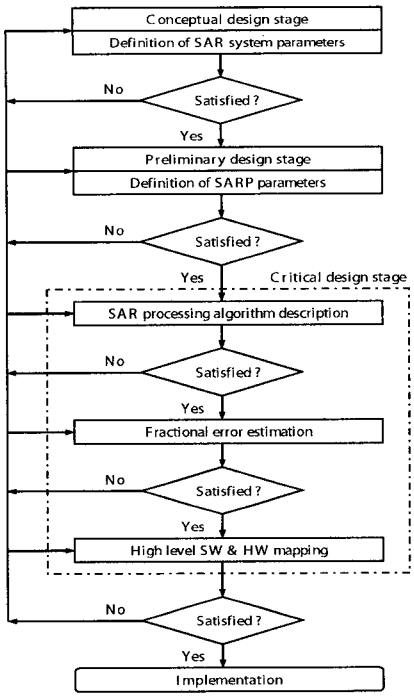

Fig. 1. Design sequence of the SARP.

data rate is discussed, and parallel processing is applied to the HW mapping. Section 3 introduces the conditions and parameter values of the ESARP for simulation.

Ⅱ. Proposed Design Method

2.1 SAR Processing Algorithm Description

To design the SARP in detail, it should be described firstly the SAR processing algorithm. RDA[4, 5] is often chosen in a SARP because it has ability to compress a point target accurately, and because it can be implemented as a two one-dimensional matched filtering process: range and azimuth compression. Thus, RDA is selected for the SAR processing algorithm in this paper. The range and azimuth compression is performed with FFT correlation algorithm because of computational efficiency and capability of an optimal compression filter to compensate for pulse distortion. Fig. 2 shows the functional flow diagram of compression. The input data to the compression sequence have been checked and unpacked to floating point format by the unpack processes in preprocessor, and then they are synchronized data. Table 1 illustrates the main parameters for each processing steps of RDA in Fig. 2. The detailed description contents for each processing step of RDA consist of I/O parameters, operations using pseudo-code, and comments: These are described distinctly in [6].

Table 1. Main processing parameters of RDA

Processing steps	Main Parameters	Remark
FFT	FFT length, block size	Range & azimuth
IFFT	IFFT length	Range & azimuth
Range reference function multiplication & filtering	Replica data, scaling factor	These parameters are required to generate the range reference function.
RCMC & resampling	Geometric parameters, number of looks, block size	RCMC: Range Cell Migration Correction
Azimuth reference function multiplication & filtering	Azimuth chirp, Doppler filter, azimuth antenna pattern correction parameters, output scaling factor	These parameters are required to generate the azimuth reference function.
Azimuth deskew	Deskew parameters, azimuth pixel spacing	
Range resample	Product parameters	This processing is carried out if the image product is detected.
Image data	Product format parameters	Single look complex and detected, multi-look detected

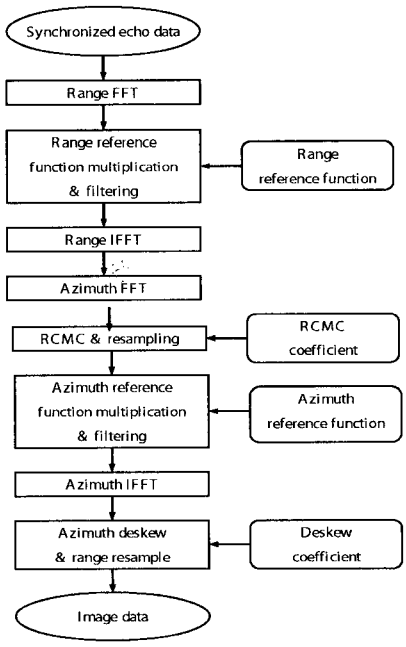

Fig. 2. Main processing flow of RDA.

2.2 Fractional Error Estimation

After design of RDA, it should be performed the fractional error estimation of the SARP for analyzing whether the design is proper or not. Fractional errors of the SARP are due to the imperfections in processing algorithm or in its implementation. These come from processing steps of the algorithm. It is assumed that SAR system errors are not considered to be part of the fractional error of SARP because they have been already considered and included in the preceding design stage. In order to estimate the fractional errors of SARP, we should follow the calculation sequence of Fig. 3. Each step in Fig. 3 indicates fractional error sources and relationship between parameters and can be calculated by using equations in Table 2. Table 3 illustrates the parameters used in Table 2 and Fig. 3. For example, calculating

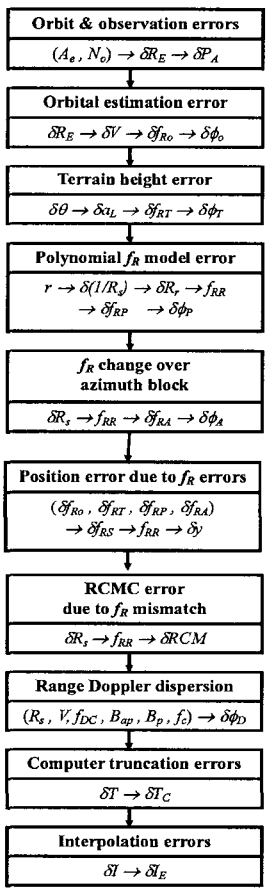

Fig. 3. Fractional error estimation sequence.

Table 2. Fractional error equations

Fractional error sources	Fractional error equations
Orbit & observation errors	$\delta R_E = A_e / (N_o)^{0.5}$
	$\delta P_A = \delta R_E / \tan i$
Orbital estimation error	$\delta V = (V\, \delta R_E) / 2\, R_E$
	$\delta f_{Ro} = (4V\, \delta V) / (\lambda R_s)$
	$\delta \phi_o = 180\, \delta f_{Ro}\, (\tau_s / 2)^2$
Terrain height error	$\delta\theta = (h \sin i)/R_s$
	$\delta \alpha_L = \{V^2 / (r_E + h_p)\} \sin\theta \{(h_t / R_s) \sin i\}$
	$\delta f_{RT} = (2/\lambda) \{V^2 / (r_E + h_p)\} (h_t / R_s) \sin\theta \sin i$
	$\delta \phi_T = 180\, \delta f_{RT}\, (\tau_s / 2)^2$
Polynomial f_R model error	$r = \{(\text{swath width}) \times \sin i\}/2$
	$R_s = R_m - r$
	$\delta(1/R_s) = 1/R_s - (1/R_m + r/R_m^2 + r^2/R_m^3)$
	$\delta R_r = \delta(1/R_s)/(1/R_s)$
	$f_{RR} = (2/\lambda R_s) \{V / (1 + h_p / r_E)^{0.5}\}^2$
	$\delta f_{RP} = f_{RR} \times \delta R_r$
	$\delta \phi_P = 180\, \delta f_{RP}\, (\tau_s / 2)^2$
f_R change over azimuth block	$\delta R_s = (A_b/2) \times \{c_s / (2 \times A_R)\}$
	$\delta f_{RA} = (\delta R_s / R_s) \times f_{RR}$
	$\delta \phi_A = 180\, \delta f_{RA}\, (\tau_s / 2)^2$
Position error due to f_R errors.	$\delta f_{RS} = (\delta f_{Ro}^2 + \delta f_{RT}^2 + \delta f_{RP}^2 + \delta f_{RA}^2)^{0.5}$
	$\delta y = V_g\, (f_{DC} / f_{RR}^2)\, \delta f_{RS}$
RCMC error due to f_R mismatch	$\delta RCM = (\delta R_s / R_s) \{\lambda / (4 f_{RR})\}$
	$\{(B_{ap} / 2 + f_{DC})^2 - f_{DC}^2\}$
	If yaw steering is perfect, f_{DC} is zero.
	If yaw steering is not perfect, then $f_{DC} \neq 0$.
Range Doppler dispersion	$\delta \phi_D \approx (\pi R_s\, c_s / 2) [\{((f_{DC} + 0.5 B_{ap})^2$
	$- f_{DC}^2)\, B_p^2\} / \{V^2 f_c^2\, (B_p + f_c)\}]$
Computer truncation errors	$\delta T_C \approx \delta T \times (N_c)^{0.5}$
Interpolation errors	$\delta I_E \approx \delta I \times (A_b)^{0.5}$

orbit & observation errors is as follows: δR_E is derived from Ae and N_o. So, δP_A, fractional across track error due to orbit altitude error, can be calculated by using δR_E.

To derive the fractional error equations and values, the following contents are considered. Keplerian orbit[7, 8] and elliptic earth model are used. Regression is fitted to Keplerian orbit. To get the terrain height error, we assume terrain height (h_t) is 1 km. It is also assumed that correlator uses a 32-bit central processing unit (CPU) using IEEE format. It stores data of 2^{22} bit precision. After each operation, there is an error of 1/2 bit in 2^{22}, a fractional error (δT) of

Table 3. Parameters used in Table 2 and Fig. 3

Parameters	Description	Parameters	Description
R_E	Distance from earth center to the SAR platform	δf_{RS}	Focus errors due to f_R
A_e	Equipment accuracy	c_s	Speed of light
N_o	Number of observations	B_{ap}	Azimuth processing bandwidth
B_p	Pulse bandwidth	f_c	Carrier frequency
i	Incidence angle	λ	Wave length
δP_A	Fractional across track error due to orbit altitude error	δR_E	Fractional position error due to observations
V	SAR platform velocity	N_e	Number of computing errors
R_s	Slant range	τ_s	Synthetic aperture duration
A_R	Analog to Digital Converter (ADC) rate	δf_{Ro}	Fractional f_R error due to orbital estimation error
r_E	Radius of the earth at nadir	$\delta\phi_b$	Fractional phase error due to orbital estimation error
h_p	SAR platform altitude to the nadir point	δf_{RT}	Fractional f_R error due to terrain height error
θ	Look angle	$\delta\phi_T$	Fractional phase error due to terrain height error
f_R	Doppler frequency modulation	δf_{RP}	Fractional f_R error due to polynomial f_R model error
R_m	Slant range at mid of swath	$\delta\phi_P$	Fractional phase error due to polynomial f_R model error
r	Slant range variation relative to swath center	δf_{RA}	Fractional f_R error due to f_R change over azimuth block
δR_r	Relative error in $1/R_s$	$\delta\phi_A$	Fractional phase error due to f_R change over azimuth block
$\delta(1/R_s)$	Real estimation error	δy	Along track displacement error due to f_R errors
f_{RR}	f_R in the case of $1/R_s$ dependence	δRCM	Fractional RCMC error due to f_R mismatch
A_b	Azimuth block size	$\delta\phi_D$	Fractional phase error due to range Doppler dispersion
a_L	Look direction component of orbital acceleration	δT_C	Fractional RMS error due to computer truncation
V_g	Ground velocity	δI_E	Fractional RMS error due to interpolation
f_{DC}	Doppler center frequency		

1 in 10^7[9]. A sinc interpolation used to interpolate band-limited data with a uniform spectrum gives a fractional mean square error (δI): As an example, 5 points sinc interpolation is 0.012, 7 is 0.006, and 9 is 0.003 dB. The detailed work about how to derive the fractional error equations is described distinctly in [6] and [10].

2.3 High Level SW & HW Mapping

If there is no problem during the fractional error estimation, then we can progress the next design stage: high level SW and HW mapping of the SARP. In order to produce the proper SW and HW configuration generating SAR image, this paper suggests the SW and HW mapping procedure of SARP as Fig. 4.

The basic unit of SAR image formation processing is a scene. The types of processing tasks involved in SARP can be categorized as follows: scene-wide tasks which apply to the entire scene, block-wide tasks associated with data blocks independent of one another, sample-by-sample tasks, man machine interface (MMI) tasks, and I/O tasks.

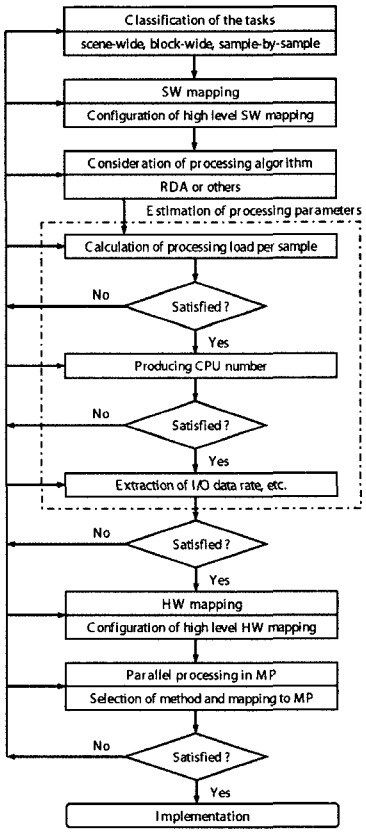

Fig. 4. High level SW and HW mapping procedure.

2.3.1 SW Mapping

Based on the types of processing tasks, SARP can be conveniently subdivided into three functional segments as Fig. 5. The data assembly processor (DAP) takes its input as a stream of echo data records from the synchronized data. The support processor (SP) performs those calculations that need to look at information on a scene-wide basis. Each data block can then be processed without reference to the processing of any other block. Main processor (MP) handles most of the processing load. Thus high level SW mapping can be constructed as Fig. 5.

The design must be first to satisfy the main processing throughput requirement. Fig. 2 illustrates the main processing flow of RDA which is performed in MP. In order to map SW to HW of SARP we need estimates of: the expected number of floating point operations (FLOP) per scene sample, the required scene processing rate, the capability of useful CPU, the expected data I/O rates, and the capability of useful disk controllers and buses[11]. Table 4 illustrates the calculation method for estimating the number of FLOP per sample in main processing sequence (refer Fig. 2), where N_F is the number of FFT points. Total in Table 4 is safe to allow an overhead of approximately 20~30% for proper operations.

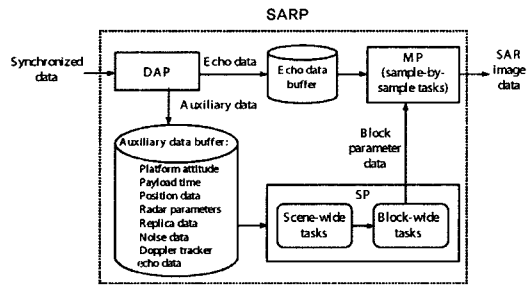

Fig. 5. High level SW mapping configuration

Table 4. Estimated FLOP per sample in main processing

Processing step	Basis of FLOP estimate
Range FFT	$5 \log_2 N_F$
Range IFFT	$5 \log_2 N_F$
Azimuth FFT	$5 \log_2 N_F$
RCMC & resample	2 × interpolation length × 2
Azimuth IFFT	$5 \log_2 N_F$
Azimuth deskew	2 × interpolation length × 2 × 2
Range resample	2 × interpolation length × 2 × 2
Total	Sum of the above results

2.3.2 HW Mapping

MP may require a multi-CPU HW. An array processor is a more efficient system for executing many similar tasks in parallel using partitioned data blocks. However, an array processor requires a host computer system to isolate it from the local area network (LAN) and to co-ordinate its activities. This facilitates operational pipelining[12][13]. Thus, high level HW can be built as Fig. 6.

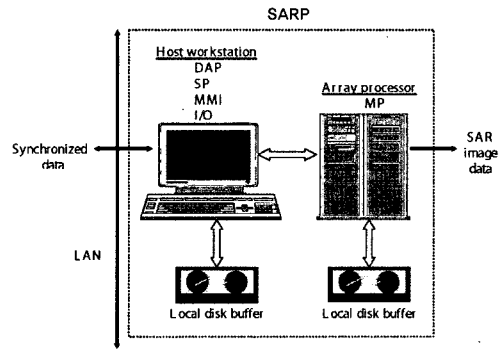

Fig. 6. High level HW mapping configuration

2.3.3 Parallel Processing in MP

A wide range of multiple CPU task configurations can be devised[14]. Fig. 7 depicts the parallel processing in MP. At any time, once the processes are in dynamic equilibrium (with non

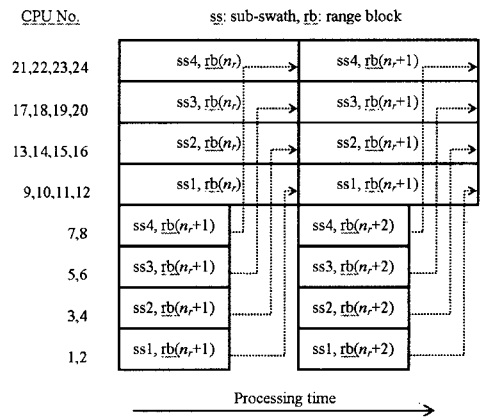

Fig. 7. Range-azimuth pipeline in subdivided sub-swaths

blocking I/O), within a sub-swath the range compression CPUs work simultaneously on data from range block n_r+1 in a pipeline with the azimuth compression CPUs that are working on data from range block n_r. Sub-swaths are processed in parallel[15]. This processing allows us to improve efficiency by adjusting the proportion of CPUs dedicated to range processing in relation to those doing azimuth processing so that pipelining leads to shorter idle periods[16].

Ⅲ. Simulation and Discussion

The parameters and requirements of the EX-SAR and ESARP used for simulation are shown at below Table 5 and 6. EX-SAR uses on-board global positioning system (GPS) receivers for orbital estimation. Orbit is known to be Keplerian. It is assumed that the EX-SAR has an imaging mode of stripmap. By steering antenna beam, its swath number (SN) is variable from SN1 to SN19, depending on incident angle. But their swath width is same.

Table 5. Parameters for the EX-SAR and ESARP

Symbol	Values	Unit	Remark
r_E	6370	km	
h_p	618	km	
f_c	9.65	GHz	
V	7553	m/s	
V_g	6883	m/s	SN1
	6854	m/s	SN19
λ	0.031	m	
i	14.9~49.9	°	
θ	13.52~44.18	°	
R_s	637.4	km	SN1(near)
R_m	642	km	SN1(mid)
R_m	893	km	SN19(mid)
R_s	905.4	km	SN19(far)
f_{DC}	1214	Hz	SN1
	3578	Hz	SN19
B_p	65	MHz	SN1
	25	MHz	SN19
c_s	3×10^8	m/s	
A_R	60	MHz	
τ_s	0.682~0.946	s	
A_e	± 20	m	
N_o	5		
B_{ap}	2551	Hz	
A_b	115	samples	
N_F	8192		2^{13}

Table 6. Parameters and requirements for the EX-SAR and ESARP

Contents	Values	Unit
f_R change over azimuth block	≤ 22.5	°
Swath width	33	km
Integrated side lobe contribution	0.4	dB
Yaw steering error	± 0.6	°
Impulse Response Function (IRF) broadening criteria	< 1	%
Number of interpolation point	10	
Pulse Repetition Frequency (PRF)	2500	Hz
A CPU processing rate	3×10^8	FLOP/s
Read/write rate of a disk controller	20	Mbytes/s
Mean data input rate	2.5	Mbytes/s
Ethernet data rate	1	Gbit/s
The margin of processing load	25	%
The implementation efficiency of CPU	33	%
The margin of over sampling	20	%
The size of a scene	$10^4 \times 10^4$	
Scene processing rate	1/10th real time	

ESARP uses RDA as the processing algorithm, and then the functional flow for processing algorithm of the ESARP is defined as Fig. 2, which is a baseline of this simulation.

3.1 Fractional Error Estimation

By using error equations in Table 2 and estimation sequence in Fig. 3, we can calculate all error values and summarize the worst case error values to the ESARP as Table 7. The noise type error, 0.129 dB (if 5 points interpolation is used), is much less than the integrated side lobe contribution (0.4 dB). The most significant fractional error sources are platform ephemeris errors (in particular, along and across track observation errors) and terrain height errors. These geometric error sources cause slant range errors that result in image distortions.

f_R change over azimuth block is measured as a fractional phase error, and its associated impact is the broadening of the IRF. It is implied that the f_R errors cause broadening of the IRF. Also, we can find out that the fractional phase error at the edge of the aperture is a function of the platform velocity uncertainty and some other parameters [10].

Table 7. Fractional error values for the worst case

Fractional error sources	Worst case values	Unit	Symbol	Remark
• Orbit and observation errors -Observation error: along track -Observation error: across track -Orbit altitude error: across track	 8.944 8.944 33.62	 m m m	 δR_E δR_E δP_A	 SN1
• Orbital estimation error	0.21	°	$\delta\phi_o$	SN19
• Terrain height error	12.484	°	$\delta\phi_T$	SN19
• Polynomial f_R model error	0.421	°	$\delta\phi_P$	SN19
• f_R change over azimuth block	24.845	°	$\delta\phi_A$	SN1
• Position error due to f_R errors	1.188	m	δy	SN19
• RCMC error due to f_R mismatch -f_R mismatch (with perfect yaw steering) -f_R mismatch (without perfect yaw steering: 0.6 ° yaw steering error)	 5.41×10^{-4} 3.57×10^{-3}	 m m	 δRCM δRCM	 SN1 SN19
• Range Doppler dispersion : with 0.6 ° yaw steering error	6.655	°	$\delta\phi_D$	SN1
• Computer truncation errors	2×10^{-6}	dB	δT_C	negligible
• Interpolation errors: 5 points	0.129	dB	δI_E	

3.2 High Level SW & HW Mapping

ESARP uses RDA as the processing algorithm, and then high level SW configuration is defined as Fig. 5. In order to map SW to HW for ESARP, it is required to consider the following contents.

3.2.1 Estimated Processing Load per Scene Sample

By using Table 4, we can calculate the FLOP per sample as Table 8. The minimum number of FLOP per sample is expected to be 460. It is allowed a margin of approximately 25% for these operations giving an estimated 600 FLOP per sample.

Table 8. Estimated FLOP per sample

Processing step	FLOP per sample
Range FFT	65
Range IFFT	65
Azimuth FFT	65
RCMC & resample	40
Azimuth IFFT	65
Azimuth deskew	80
Range resample	80
Total	460

3.2.2 Required Scene Processing Rate and Capability of CPU

From Table 6, we require 1/10th real-time processing rate. At a PRF of 2500 Hz, the data for a scene is gathered in about 4 s. This corresponds to a scene processing time of about 40 s. One scene requires a total of: $600\times10^4\times10^4=6\times10^{10}$ FLOP. Allowing for 20% over sampling in range and azimuth this becomes: 9×10^{10} FLOP. We require 1/10th real time processing rate corresponding to a scene processing time of about 40 s. Therefore we require an effective processing rate of: (9×10^{10}) / $40=2.3\times10^9$ FLOP/s. A CPU is capable of 3×10^8 FLOP/s, its efficiency is 33%, and then we require: (2.3×10^9) / $(0.33\times3\times10^8)\approx$ 23 CPUs to achieve the processing rate requirement. Then, the sample-by-sample operations of ESARP require the resources of 23 CPUs processing simultaneously.

3.2.3 I/O Data Rates & Capability of Disk Controller

One scene consists of $10^8\times4\times2$ bytes. Mean data input rate, at 1/10th real time rate, is of the order of 2.5 Mbytes/s. In 40 s the processor mean output rate$\cong(8\times10^8)/40\cong20$ Mbytes/s. One disk controller can deliver a read / write rate of 20 Mbytes/s sequential access. Therefore one disk controller each for I/O may be sufficient for the ESARP. The aggregate I/O data rate is about 200 Mbits/s and must be handled by a LAN. This can be achieved using a 1Gbit/s Ethernet. A SUN workstation, for example, can support total disk I/O rate of 100 Mbytes/s, so the aggregate mean I/O rate of 23 Mbytes/s is feasible with plenty of margin. It has been determined that we must achieve an effective FLOP rate of about 2.3×10^9 FLOP/s in the sample-by-sample tasks and that in practice this would require about 23 CPUs working simultaneously.

With the above results, we can produce the HW mapping configuration of ESARP as Fig. 6. To perform the parallel processing in MP of ESARP, we define data into 4 sub-swaths. Each sub-swath dedicates: 2 CPUs for range processing

(partitioned data), 4 CPUs for azimuth processing (partitioned data). It is arranged as a range-azimuth pipeline (the range processing load≅1/2 that of azimuth), then total is 24 CPUs in parallel (see Fig. 7). This configuration meets the performance levels derived in this paper. I/O should be supported by non-blocking methods. One disk controller is required for input and at least one controller for output, served either by the array processor or host. Having first defined the MP throughput capability the host workstation resources can be defined so that neither of them constitutes a data flow bottleneck[6] [14].

Ⅳ. Conclusion

This paper starts from the design sequence of SARP and presents the main processing algorithm of RDA and its parameters. According to the design process, it finds out a suitable sequence for fractional error estimation. Fractional errors are quantified in terms of residual phases. Finally, high level SW and HW mapping is performed. Also, simulation has been done to the ESARP to examine the usefulness and practice of the method. In the case of ESARP, the dominant location errors come from orbit altitude error. The problem is the phase error (24.845°) due to f_R change over azimuth block at SN1. The computation errors are much less at a few tenths of a pixel. They are small enough for almost applications. The results of the SW and HW mapping are: host workstation with 2 CPU and several disk controllers, 1 Gbit/s Ethernet LAN, and 24 CPU parallel processing array. SW can be developed without dependence upon HW selection, because this method does not require HW dependent protocols.

Therefore, this paper has proposed the overall work related to critical design method of the space-based SARP using RDA. Our approach can provide a practical methodology for developing the space-based SARP using RDA.

REFERENCES

[1] E.A. Herland, "A SAR processor for a GIS environment," Proc. IGARSS '91, vol.2, pp. 623-626, 1991.

[2] C. Elachi, T. Bicknell, R.L. Jordan, and C. Wu, "Spaceborne synthetic aperture imaging radars: applications, techniques, and technology," Proc. IEEE, vol.70. no.10, pp.1174-1209, Oct. 1982.

[3] C. Elachi, Spaceborne Radar Remote Sensing: Applications and Techniques, pp.85-162, IEEE Press, New York, 1988.

[4] I.G. Cumming and J.R. Bennett, "Digital processing of SEASAT synthetic aperture radar data," IEEE International Conference on ICASS'79, vol.4, pp.710-718, Washington, D.C., April, 1979.

[5] A.A. Thompson, E.S.H. Cheung, and C.Y. Chang, "Precision SAR processing for Radarsat," Proc. IGARSS '95, vol.3, pp. 2307-2309, 1995.

[6] In Pyo Hong, A detailed design technique of the space-based SAR processor using RDA, Ph.D. Thesis, Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Feb. 2004.

[7] L.J. Cantafio, Space-based Radar Handbook, pp.1-80, Artech House, Norwood, MA, 1989.

[8] K. Eldhuset, "A new fourth-order processing algorithm for spaceborne SAR," IEEE Trans. Aerosp. Electron. Syst., vol.34, no.3, pp.824-835, July 1998.

[9] I. Koren, Computer Arithmetic Algorithms, pp. 45-68, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[10] In Pyo Hong and Han Kyu Park, "Fractional error estimation technique of the space-based SAR processor using RDA," IEICE Trans. Commun., vol.E87-B, no.4, pp.967-974, April 2004.

[11] J.C. Curlander and R.N. McDonough, Synthetic Aperture Radar: Systems and Signal Processing, pp.154-208 and 427-591, John

Wiley & Sons, New York, 1991.

[12] G. Fabbretti, A. Farina, D. Laforenza, and F. Vinelli, "Mapping the synthetic aperture radar signal processor on a distributed-memory MIMD architecture," Parallel Comput., vol.22, issue 5, pp.761-784, August 1996.

[13] In Pyo Hong, Jae Woo Joo, and Han Kyu Park, "Top level software and hardware mapping method of the SAR processor," the Journal of the Korean Institute of Communication Sciences, vol. 26, no. 9B, pp. 1308-1313, Sep. 2001.

[14] D.I. Moldovan, Parallel Processing: from Applications to Systems, pp.181-384, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[15] In Pyo Hong and Han Kyu Park, "High level SW and HW mapping method of the space-based SAR processor using RDA," Signal Processing, vol.84, no.5, pp. 943-949, May 2004.

[16] In Pyo Hong, Jae Woo Joo, and Han Kyu Park, "The mapping method for parallel processing of SAR data," the Journal of the Korean Institute of Communication Sciences, vol. 26, no. 11A, pp. 1308-1313, Nov. 2001.

홍 인 표(In Pyo Hong) 정회원

1982년 2월 연세대학교 전자공학과 졸업

1997년 2월 충북대학교 정보통신공학과 공학석사

2004년 2월 연세대학교 전기전자공학과 공학박사

1984년 3월~현재 국방과학연구소 책임연구원

<관심 분야> 위성통신, SAR/IFSAR, Data/Image Fusion