

## Article

# An Efficient Dual-Channel Data Storage and Access Method for Spaceborne Synthetic Aperture Radar Real-Time Processing

Guoqing Wang, He Chen and Yizhuang Xie \*

Beijing Key Laboratory of Embedded Real-time Information Processing Technology,  
Beijing Institute of Technology, Beijing 100081, China; bit\_wgq@163.com (G.W.); chenhe@bit.edu.cn (H.C.)

\* Correspondence: xyz551\_bit@bit.edu.cn; Tel.: +86-136-935-19576

**Abstract:** With the development of remote sensing technology and very large-scale integrated circuit (VLSI) technology, the real-time processing of spaceborne Synthetic Aperture Radar (SAR) has greatly improved the ability of Earth observation. However, the characteristics of external memory have led to matrix transposition becoming a technical bottleneck that limits the real-time performance of the SAR imaging system. In order to solve this problem, this paper combines the optimized data mapping method and reasonable hardware architecture to implement a data controller based on the Field-Programmable Gate Array (FPGA). First of all, this paper proposes an optimized dual-channel data storage and access method, so that the two-dimensional data access efficiency can be improved. Then, a hardware architecture is designed with register manager, simplified address generator and dual-channel Double-Data-Rate Three Synchronous Dynamic Random-Access Memory (DDR3 SDRAM) access mode. Finally, the proposed data controller is implemented on the Xilinx XC7VX690T FPGA chip. The experimental results show that the reading efficiency of the data controller proposed is 80% both in the range direction and azimuth direction, and the writing efficiency is 66% both in the range direction and azimuth direction. The results of a comparison with the recent implementations show that the proposed data controller has a higher data bandwidth, is more flexible in its design, and is suitable for use in spaceborne scenarios.

**Keywords:** Synthetic Aperture Radar (SAR); real-time processing; data storage and access; Field Programmable Gate Array (FPGA); dual-channel pipeline processing



**Citation:** Wang, G.; Chen, H.; Xie, Y. An Efficient Dual-Channel Data Storage and Access Method for Spaceborne Synthetic Aperture Radar Real-Time Processing. *Electronics* **2021**, *10*, 662. <https://doi.org/10.3390/electronics10060662>

Academic Editors:

Gustavo Marrero-Callico,  
Himar Fabelo and Lucana Santos

Received: 5 February 2021

Accepted: 9 March 2021

Published: 12 March 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Spaceborne synthetic aperture radar (SAR) is a kind of high-resolution microwave imaging technology which has many characteristics, such as all-time, all-weather, high-resolution and a long detection distance. It is a radar that uses the Doppler information generated by the relative motion between the radar platform and the detected target, and uses signal processing method to synthesize a larger antenna aperture [1–3]. As it can penetrate clouds, soil and vegetation, it has become more and more widely used in many important fields [4–6]. Recent publications have reviewed the applications of satellite remote sensing techniques for hazards manifested by solid earth processes, including earthquakes, volcanoes, floods, landslides, and coastal inundation [7–9].

The world's first SAR imaging satellite, SEASAT-1, was launched in 1978, proving the feasibility of microwave imaging radar in Earth observation [10]. Since then, many countries have scrambled to carry out spaceborne SAR research, and it has been used as an important means of Earth observation in recent years. For example, the Gaofen-3 satellite launched by China in 2016 is a C-band multi-polarization SAR satellite that can quickly detect and assess marine disasters, and it could improve China's disaster prevention and mitigation capabilities [11]. The Sentinel-B satellite launched by the European Space Agency (ESA) in 2016 and the previously launched Sentinel-A satellite belong to the Sentinel-1 SAR satellite mission [12]. Its purpose is to provide an independent operational capability for continuous radar mapping of the Earth with enhanced revisit frequency, coverage,

timeliness and reliability for operational services and applications requiring long time series. In 2019, SpaceX successfully launched the Radar Satellite Constellation Mission (RCM) satellite with a Falcon-9 rocket at Vandenberg Air Force Base in the United States [13]. The satellite has a wide range of applications, including ice and iceberg monitoring, marine winds, oil pollution monitoring and response, and ship detection, etc. In recent years, natural disasters have occurred in various places, and many countries have put forward their requirements regarding the increased performance of SAR satellites. The future spaceborne SAR will develop in the direction of multi-band, high-resolution, and ultra-wide amplitude. The National Aeronautics and Space Administration (NASA) plans to launch the NASA—Indian Space Research Organization (ISRO) Synthetic Aperture Radar (NISAR) mission satellite in 2022 [14]. It will be the first satellite to use two different frequencies (L-band and S-band) to measure changes in the Earth's surface by less than 1 cm. The German Aerospace Agency plans to launch TerraSAR-NG satellites in 2025, which can achieve a resolution of 0.25 m [15,16].

Most of the above mentioned missions place high demands on the real-time performance of SAR data processing. The traditional processing method is that the satellite stores the raw data and sends them to the equipment on the ground for processing. With the continuous improvement of high-resolution and low-latency requirements for spaceborne SAR, traditional data processing methods cannot meet these needs. In recent years, the rapid development of large-scale integrated circuit technology has made on-board real-time processing of spaceborne SAR possible. This technology can greatly reduce the pressure of satellite-to-ground data transmission, improve the efficiency of information acquisition, and enable policymakers to plan and respond quickly. Spaceborne SAR real-time processing has become a research highlight in the field of aerospace remote sensing. In 2010, the ESA funded the National Aerospace Laboratory of the Netherlands to develop the next-generation multi-mode SAR On-board Payload Data Processor (OPDP) [17]. The processor consists of a LEON2 FT anti-irradiation processing chip, an FFTC ASIC chip and a Synchronous Dynamic Random-Access Memory (SDRAM) structure designed by EDAC that can complete  $1K \times 1K$  granularity SAR imaging processing, which will be applied to the ESA EASCOH2O mission, the BIOMASS mission and the panelSAR satellite. In 2016, the California Institute of Technology (CIT) completed an Unmanned Aerial Vehicle (UAV) SAR real-time processing system using a single Xilinx Virtex-5 space-grade Field-Programmable Gate Array (FPGA) chip [18]. The system is the first that applies space-grade FPGA to SAR real-time signal processing. It provides a basis for the use of space-grade FPGA in spaceborne SAR on-board real-time processing scenarios. SpaceCube is a series of space processors developed by NASA in the United States. SpaceCube established a hybrid processing approach combining radiation-hardened and commercial components while emphasizing a novel architecture harmonizing the best capabilities of Central Processing Unit (CPU), Digital Signal Processor (DSP) and FPGA [19]. The latest SpaceCube v3.0 processing board launched in 2019 is equipped with Xilinx Kintex UltraScale, Xilinx Zynq MPSoC, Double-Data-Rate Three Synchronous Dynamic Random-Access Memory (DDR3 SDRAM) and NAND Flash chips, which can complete on-board real-time data processing missions, and provide a processing solution for next-generation needs in both science and defense missions [20]. In 2020, NASA's Goddard Space Flight Center developed a prototype design for the NASA SpaceCube intelligent multi-purpose system (IMPS) to serve high-performance processing needs related to artificial intelligence (AI), on-board science data processing, communication and navigation, and cyber security applications [21,22].

As indicated by the development of spaceborne SAR real-time processing technology, a high-performance on-board real-time processing platform needs to be equipped with processors and memories that meet many restrictions regarding power, performance and capacity constraints. CPU, DSP, FPGA, Graphics Processing Unit (GPU) and Application Specific Integrated Circuit (ASIC) are superior, to some extent, when it comes to real-time processing. Although CPU and DSP have strong design flexibility, they cannot provide enough computing resources, and there will be bottlenecks in massive data processing.

GPU has strong processing capabilities, but has high power consumption and is not suitable for on-board scenarios. ASIC is a device that can be customized and has sufficient processing power, but the development time is long. FPGA has irreplaceable advantages in terms of on-chip resource storage, computing capabilities, reconfigurable characteristics, and can meet the requirements of large throughput and real-time processing under spaceborne conditions [23,24]. The on-chip storage resources of the above-mentioned processors are very limited, which cannot meet the requirements for massive data processing in the SAR algorithm. Taking Stripmap SAR imaging of  $16,384 \times 16,384$  granularity raw data (5 m resolution) as an example, the required data storage space is up to 2 GB [25]. Therefore, a high-speed, large-capacity external memory must be selected as the storage medium for SAR data. Hard Disk Drives (HDDs) have a higher storage capacity, but their stability is limited. Solid State Drives (SSDs) have strong stability, but a shorter lifetime. Flash memory is a non-volatile form of memory that can store data for a long time, but it has low data transmission efficiency and high power consumption. DDR3 SDRAM is a new type of memory chip developed on the basis of SDRAM chips [26]. It has double data transmission rate and meets the requirements of on-board processing in terms of capacity, speed, volume, and power consumption [27–29]. In summary, FPGA and DDR3 SDRAM are ideal processor and memory formats for on-board data processing.

Due to the frequent cross-row access in DDR3, the matrix transposition operation has become a bottleneck restricting SAR real-time imaging. In recent years, researchers have proposed a variety of hardware implementation methods for matrix transposition in SAR real-time system. Berkin et al. [30] optimized memory access method for multi-dimensional FFT by using block data layout schemes. However, the method was only optimized for multi-dimensional FFT algorithm, was not suitable for spaceborne SAR algorithm. Mario et al. [31] optimized the matrix transposition in a continuous flow in a hardware system. However, the method was not suitable for non-squared matrices. Yang et al. [32] used the matrix-block linear mapping method to improve the data access bandwidth. However, the data controller in this method requires 8 cache RAMs, which occupies more on-chip storage resources. Li et al. [33] made the range and azimuth data bandwidth reach complete equilibrium by using matrix-block cross mapping method. However, there is only one data channel in this method, which is less flexible. Sun et al. [34] designed a dual-channel memory controller and reduced the use of cache resources. However, the data access efficiency in this method is low.

In order to solve the problem of inefficient matrix transposition due to the data transfer characteristics between FPGA and DDR3 SDRAM, an efficient dual-channel data storage and access method for spaceborne SAR real-time processing is proposed in this paper. First, we analyzed the mapping method of the Chirp Scaling (CS) algorithm and found that the maximum efficiency burden of the system mainly occurs in the matrix transposition operation due to the activation and precharge time in DDR3 SDRAM chip. Then, we proposed an optimized storage scheme that can achieve complete equilibrium of data bandwidth in range and azimuth. On this basis, a dual-channel pipeline access method is adopted to ensure a high data access bandwidth. In addition, we propose a hardware architecture to maintain high hardware efficiency. The main contributions of this paper are summarized as follows:

- The reading and writing efficiency of the conventional matrix transposition method was modeled and calculated using the time parameters of DDR3, and we found that the cross-row read and write efficiency was reduced due to the active and precharge time in DDR3.
- An optimized dual-channel data storage and access method is proposed. The matrix block three-dimensional mapping method is used to achieve the bandwidth equilibrium in range access and azimuth access, the sub-matrix data cross-mapping is used to realize the efficient use of cache Random-Access Memories (RAMs), and the dual-channel pipeline processing is used to improve the processing efficiency.

- An efficient hardware architecture is proposed to implement the above method. In this architecture, a register manager module is used to control the work mode, and an optimized address update method is used to reduce the utilization of on-chip computing resources.
- We verified the proposed hardware architecture in a processing board equipped with Xilinx XC7VX690T FPGA (Xilinx, San Jose, CA, USA) and Micron MT4A1K512M16 DDR3 SDRAM (Micron Technology, Boise, ID, USA), and evaluated the data bandwidth. The experimental results show that the read data bandwidth is 10.24 GB/s and the write data bandwidth is 8.57 GB/s, which is better than the existing implementations.

The rest of this paper is organized as follows. Section 2 introduces the data access efficiency of the SAR algorithm. In Section 3, the optimized dual-channel data storage and access method is introduced in detail. The proposed hardware architecture is illustrated in Section 4. The experiments and results are shown in Section 5. Finally, the conclusions are presented in Section 6.

## 2. SAR Data Access Efficiency Analysis

### 2.1. Chirp Scaling (CS) Algorithm Mapping Strategy

The CS algorithm is one of the most commonly used algorithms in SAR imaging processing algorithms [2]. Compared with other algorithms, the processing accuracy of the CS algorithm is higher, and it is also more suitable for hardware implementation [25]. The main process of the standard CS algorithm is as follows:

Step 1: Estimate the Doppler frequency center (DFC), and calculate the first phase function. The SAR raw data matrix is transformed by fast Fourier transform (FFT) operation in the azimuth direction, and multiplied by the first phase function to complete the chirp scaling operation, which can make the curvature of all range migration curves the same.

Step 2: Transform the data into a two-dimensional frequency domain via FFT operation in the range direction, and multiply the data by the second phase function to complete the three operations of range compression, secondary range compression (SRC), and range cell migration correction (RCMC).

Step 3: Update the equivalent velocity using the Doppler frequency rate (DFR) estimation method based on the SAR raw data to generate the third phase function. Multiply the data with the third phase function to complete phase correction and azimuth compression. Finally, the inverse FFT operation in the azimuth direction is executed to complete the CS algorithm, and the SAR imaging processing result is obtained.

The flowchart of the CS algorithm is illustrated in Figure 1, and Figure 2 shows the visualization of the SAR raw data and SAR image based on the point target model.

Each processing operation in the range or azimuth direction can be abstracted into Figure 3, that is, the three steps of reading the data, processing the data, and writing the data are completed for each piece of vector data in a two-dimensional matrix.

Assuming that the read time and write time for a piece of vector data are  $\tau_r$  and  $\tau_w$ , and the processing time is  $\tau_p$ , when processing sequentially, the processing sequence relationship for each piece of vector data can be expressed as the process shown in Figure 4a. When the hardware system has dual-channel memory, the processor can process data in parallel. Therefore, when processing each piece of vector data, the task of writing the previous piece of vector data and reading the next piece of vector data can be completed at the same time, that is, the data processing can be completed in parallel with the data reading and writing process, as shown in Figure 4b.

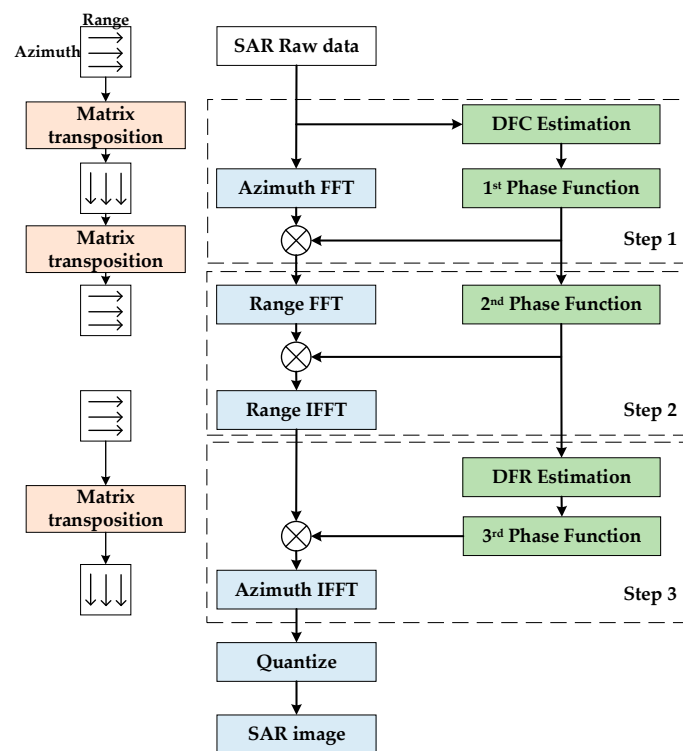


Figure 1. Flowchart of the chirp scaling (CS) algorithm.

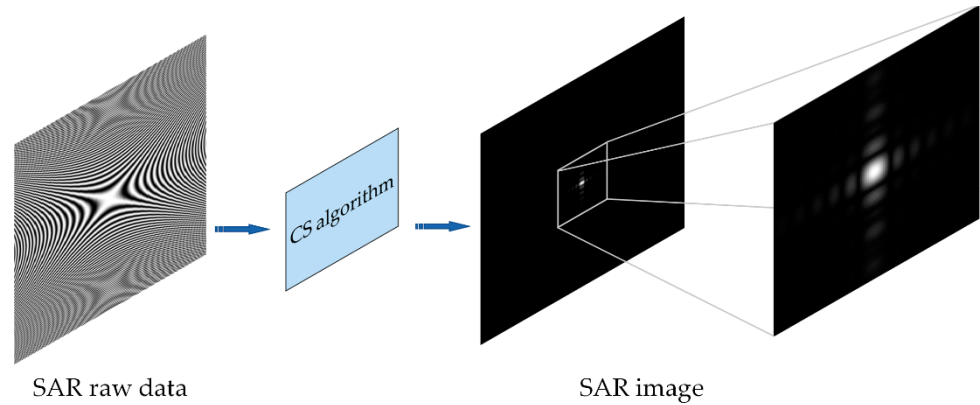


Figure 2. Visualization of the synthetic aperture radar (SAR) raw data and SAR image (point target imaging).

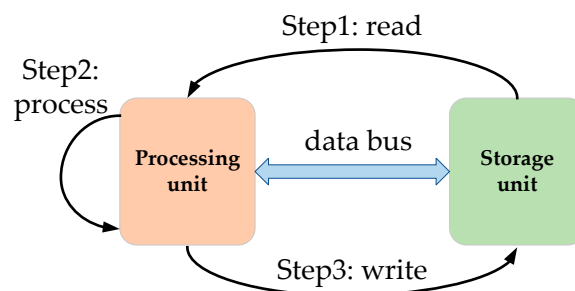
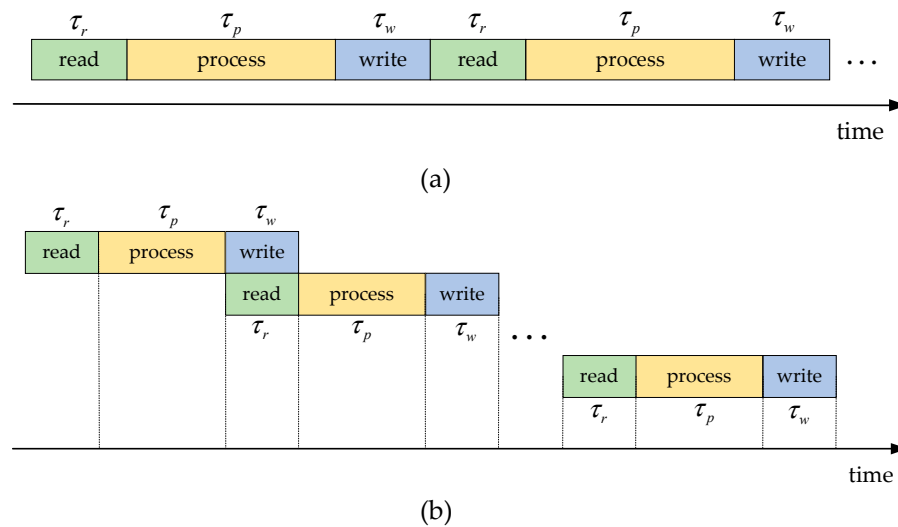


Figure 3. Abstract description of CS algorithm.





**Figure 4.** Sequence diagram of data processing. (a) The system has single-channel memory; (b) the system has dual-channel memory. Write operations and read operations can be completed in parallel.

As can be seen from Figure 4, the ability to process data and the ability to read and write data together determine the total delay time of the hardware system. As there are abundant digital signal processing (DSP) units in FPGA and support for efficient fixed-point computing, the system has strong processing capabilities [35]. On the other hand, because the FFT contains a large number of pixel-by-pixel operations [36,37], the communication between the processor and the external memory must be considered. The CS algorithm includes multiple matrix transposition operations, which increases the communication overheads of the processor and the memory, resulting in longer read and write times. When the processing time for a piece of vector data is less than the read and write time, the processing engine must wait for certain a period of time. Therefore, the key to shortening the processing delay time is to increase the efficiency of access to the memory.

## 2.2. DDR3 Data Access Characteristics Analysis

Section 1 mentioned that DDR3 SDRAM is an ideal memory for spaceborne SAR real-time processing. The SAR raw data and the processing results of each step need to be stored in DDR3. In DDR3, the specific memory cell is determined according to the row address, column address, and bank address. Therefore, DDR3 is a three-dimensional (row, column, bank) storage space. Each DDR3 chip has 8 or 16 banks, and each bank consists of thousands of rows and columns. The reading or writing of data in DDR3 is based on the burst transmission mode, that is, starting from a specified address, the contents of several consecutive memory cells are sequentially read or written. The burst length of DDR3 can be set to four or eight as needed. Generally, the burst length is set to eight in order to obtain a higher data bandwidth. Therefore, DDR3 will continuously read or write the data of 8 memory cells at the rising edges and the falling edges of four clock cycles [26].

In addition, there are two basic concepts in DDR3 read and write operations: active and precharge [26]. Active means that when reading and writing data in a row in DDR3, an active command must be issued to the row. Precharge means that the reading and writing of a certain row of data is completed in DDR3, and when it is necessary to change to the next row, a precharge command must be issued to close the previous row. When reading and writing the entire row of data, the required operation and sequence are as follows: “activate→read/write command→data transmission→precharge”.

For the read and write time of each row of data in DDR3, the active time, read or write command time, precharge time, and the minimum interval between them are all determined values. The only variable is the length of the continuous read and write data. We assume that the burst is read and written  $n$  times and the burst length is set to eight.

Then, due to the fact that the data can be sampled at the rising edges and the falling edges in DDR3, the data transmission will be completed within time  $n \times 4t_{CK}$ .

When writing the data, the time interval from the active command of one row to the active command of the next row is

$$t_{WRITE} = t_{RCD} + t_{CWL} + n \times 4t_{CK} + t_{WR} + t_{RP} \quad (1)$$

When reading the data, the time interval from the active command of one row to the active command of the next row is

$$t_{READ} = t_{RCD} + t_{RTP} + n \times 4t_{CK} + t_{RP} \quad (2)$$

where  $t_{RCD}$  is the delay time from the active command to the read or write command;  $t_{CWL}$  is the delay time from the write command to the first data transmission;  $t_{WR}$  is the minimum interval from the last data transmission to the precharge command;  $t_{RTP}$  is the minimum interval from the read command to the precharge command;  $t_{RP}$  is the minimum interval from the precharge command to the next active command;  $t_{CK}$  is the working clock of DDR3 [26].

SAR data are stored and processed in the form of a matrix. The two dimensions of the matrix are the range direction and the azimuth direction. “Range” means the across-track direction of the radar platform. “Azimuth” means the along-track direction of the radar platform. In the SAR algorithm, these two dimensions are converted to the time domain or frequency domain. For the convenience of presentation, “range” and “azimuth” are used in this paper. They only represent the two dimensions of the matrix, which can be the data in time domain or frequency domain, and the data before or after pulse compression. Each row of the SAR data matrix is called a range line, and each column is called an azimuth line.

The conventional SAR data storage method is used to store the data continuously in DDR3, as shown in Figure 5. When the size of the SAR data matrix is large, it is necessary to use multiple rows of DDR3 to store one row of the SAR data matrix. Thus, the range data are accessed sequentially, and the azimuth data are accessed in frequent cross-rows. Each row has 1024 cells in a typical DDR3. Set the burst length is set to eight, and each row needs to complete  $1024 / 8 = 128$  burst transmissions in range access mode; at this time,  $n = 128$ . By converting the relevant parameters ( $t_{RCD}$ ,  $t_{CWL}$ ,  $t_{WR}$ ,  $t_{RP}$  and  $t_{RTP}$ ) of the MT41K512M16HA-125 chip (Micron Technology, Boise, ID, USA) into a multiple of  $t_{CK}$  into the equation, the reading and writing efficiency of the range access mode can be calculated as follows [38]:

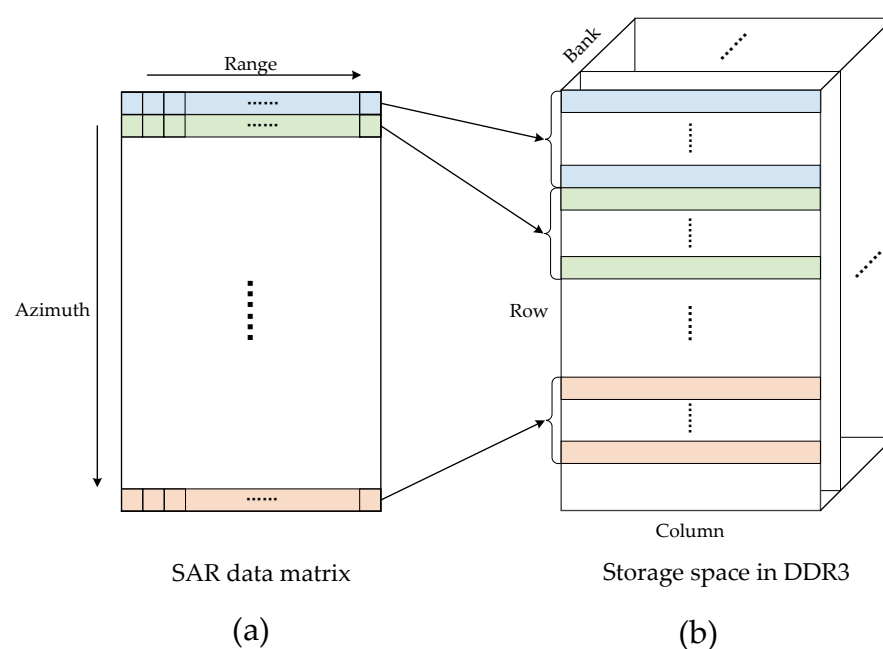
$$\eta_{WRITE\_range} = \frac{128 \times 4t_{ck}}{t_{RCD} + t_{CWL} + 128 \times 4t_{ck} + t_{WR} + t_{RP}} = 92.25\% \quad (3)$$

$$\eta_{READ\_range} = \frac{128 \times 4t_{ck}}{t_{RCD} + t_{RTP} + 128 \times 4t_{ck} + t_{RP}} = 94.64\% \quad (4)$$

In the azimuth access mode, each row only needs to complete one burst data transmission, then proceed to the next row, so  $n = 1$ . The reading and writing efficiency of the azimuth access mode can be calculated as follows:

$$\eta_{WRITE\_azimuth} = \frac{1 \times 4t_{ck}}{t_{RCD} + t_{CWL} + 1 \times 4t_{ck} + t_{WR} + t_{RP}} = 8.51\% \quad (5)$$

$$\eta_{READ\_azimuth} = \frac{1 \times 4t_{ck}}{t_{RCD} + t_{RTP} + 1 \times 4t_{ck} + t_{RP}} = 12.12\% \quad (6)$$



**Figure 5.** Schematic diagram of conventional SAR data matrix storage method. (a) The SAR data matrix; (b) each row in the matrix is stored within multiple rows in Double-Data-Rate Three (DDR3) by using the conventional mapping method.

In the conventional SAR data storage method, the reading and writing efficiency is very high for continuous address access in the range direction, but the reading and writing efficiency will be greatly reduced for frequent cross-row access in the azimuth direction, as shown in Table 1. The azimuth data bandwidth is only about 10% of the peak bandwidth, and the imbalance of the two-dimensional data bandwidths has become a bottleneck restricting the SAR real-time imaging system. In order to improve this situation, it is necessary to avoid or reduce cross-row access as much as possible to balance the data bandwidth of the two-dimensions.

**Table 1.** Performance of the conventional synthetic aperture radar (SAR) data storage method.

Performance	Range	Azimuth
Writing efficiency	92.25%	8.51%
Reading efficiency	94.64%	12.12%

### 3. Optimized Dual-Channel Data Storage and Access Method

The conventional SAR data storage method will make data access in the azimuth direction inefficient extremely, and the processing engine will be in a waiting state when the reading/writing operation is performed, thereby affecting the real-time performance of the entire system. To solve this problem, we propose an optimized dual-channel SAR data storage and access method, which maps the logical addresses in the SAR data matrix to the physical addresses in the DDR3 chip by using a new mapping method instead of the conventional linear mapping method. With this method, both the range direction and azimuth direction can reach high data bandwidths, which are more suitable for the real-time processing of spaceborne SAR.

#### 3.1. The Matrix Block Three-dimensional Mapping

The matrix block three-dimensional mapping method makes full use of the feature of cross-bank priority data access in DDR3 chips [38]. First of all, the SAR raw data matrix is divided into several sub-matrices of equal size. Then, a three-dimensional mapping method is used to map the continuous sub-matrices to different rows in different banks [32–34].



This method can maximize the equilibrium of the two-dimensional access bandwidth of the data matrix and meet the real-time requirements of the system.

The SAR data matrix can be described as  $A(x, y)$ ,  $0 \leq x \leq N_A - 1$ ,  $0 \leq y \leq N_R - 1$ . Where,  $N_A$  is the number of data in the azimuth direction of the matrix, and  $N_R$  is the number of data in the range direction of the matrix. As the SAR imaging algorithm requires multiple FFT operations, both  $N_A$  and  $N_R$  are positive integer powers of two.

The matrix is chunked into  $M \times N$  sub-matrices of the same shape, and the sub-matrix is recorded as  $A_{m,n}$ , where  $M$  and  $N$  are the signs of the sub-matrix in the azimuth direction and range direction, as shown in Figure 6. The size of the sub-matrix is  $N_a \times N_r$ , thus,  $N_a = N_A/M$ ,  $N_r = N_R/N$ . The data of the sub-matrix are mapped to a row in DDR3, so that  $N_a \times N_r = C_n$ , where  $C_n$  is the number of columns in DDR3.

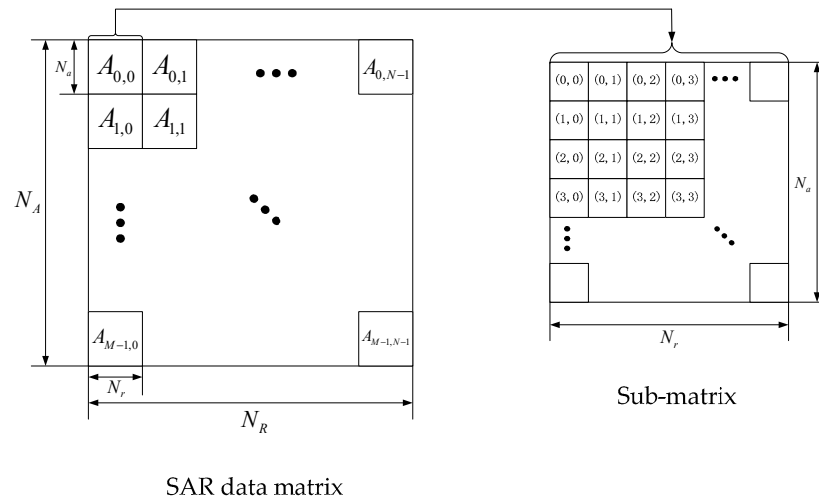


Figure 6. Schematic diagram of matrix block.

After the matrix is divided into various sub-matrices, each sub-matrix is mapped to a row of DDR3 through three-dimensional mapping method, as shown in Figure 7. The three-dimensional mapping method maps consecutive sub-matrices to different rows in different banks by using the priority access feature of cross-bank data in DDR3, effectively improving data access efficiency [38].

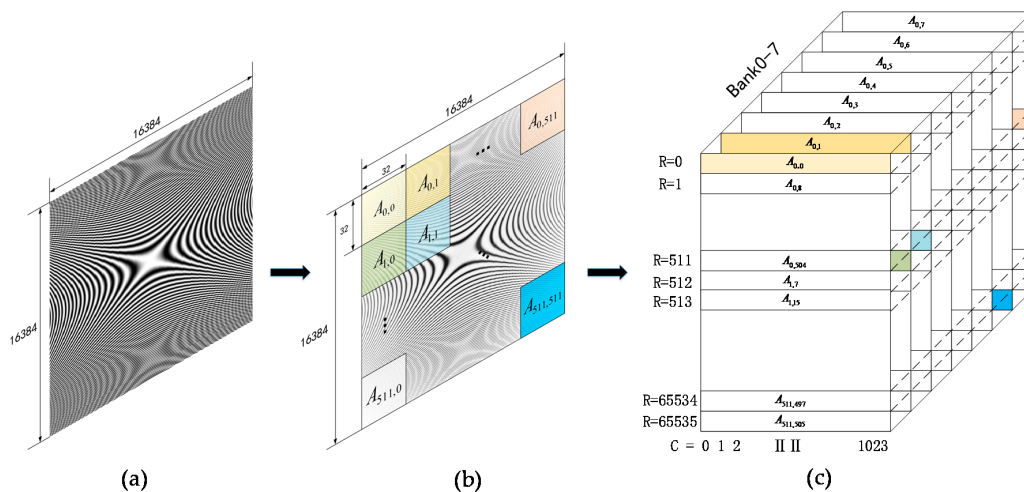
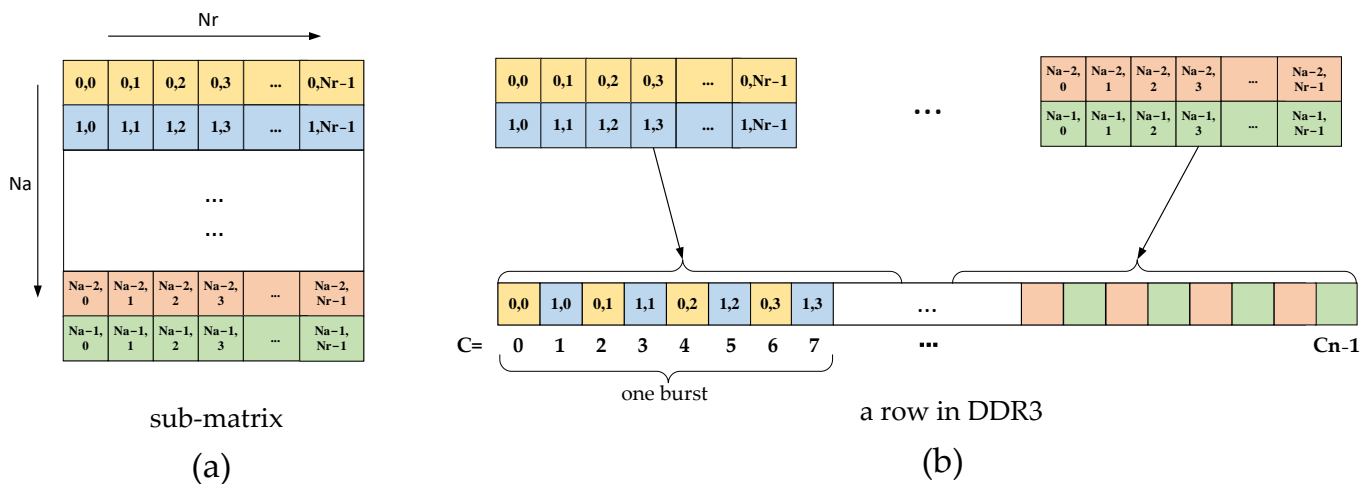


Figure 7. Schematic diagram of three-dimensional mapping for SAR data matrix (take a  $16,384 \times 16,384$  matrix as an example). (a) The SAR raw data matrix; (b) the SAR raw data matrix is chunked into many sub-matrices; (c) each sub-matrix is mapped to a row in DDR3 by using the three-dimensional mapping method.

### 3.2. Sub-Matrix Cross-Storage Method

In the conventional method, the data in the sub-matrix are stored in a row in DDR3 using linear mapping. As DDR3 adopts the burst transmission mode, each read or write operation will complete the access of eight consecutive memory cells in DDR3 when the burst length is eight in a conventional linear mapping strategy. The eight data accessed each time belong to the same range line and eight different azimuth lines. In the SAR imaging system, after the data are read, they need to be cached in the on-chip RAMs first, and then sent to the processing engine such as an FFT processor after the entire row or column of the SAR data matrix is accessed. In range access mode, the eight data transmitted in a single burst belong to one range line, so only one RAM is needed for the cache. However, in azimuth access mode, since the eight data transmitted in a single burst belong to eight azimuth lines, eight RAMs are needed for data caching. In this way, the eight corresponding processing units need to be designed, so that they consume more on-chip resources.

In this paper, we optimize the mapping of single burst transmission data by using the cross-storage method as shown in Figure 8, which reduces the use of cache RAMs and on-chip computing resources. “Cross-storage” means that the data of two adjacent range lines are alternately mapped to one row of DDR3, rather than using the conventional linear mapping method. In the proposed method, the eight data in each burst transmission in DDR3 belong to two adjacent range lines and four adjacent azimuth lines. Therefore, only four RAMs are required for data caching, and the utilization of the RAMs in the range direction is 50% and, in the azimuth direction, 100%, which is significantly improved compared with the linear mapping method. In the SAR real-time imaging system, a reduction in cache resources means that more computing resources can work at the same time, giving the system higher parallel processing capabilities.



**Figure 8.** Schematic diagram of sub-matrix data cross-storage method. (a) The sub-matrix; (b) the data in the sub-matrix are mapped to a row of DDR3 by using the cross-mapping method.

According to the “cross-storage” method, the address mapping rule can be obtained. First, determine the sub-matrix  $A_{m,n}$  where the data logical address  $(x, y)$  is located:

$$\begin{cases} m = \text{floor}(x, N_a) \\ n = \text{floor}(y, N_r) \end{cases} \quad (7)$$

where  $0 \leq m \leq M - 1, 0 \leq n \leq N - 1$ .

Furthermore, the mapping rule between the logical address  $(x, y)$  and the physical storage address  $(i, j, k)$  is obtained:

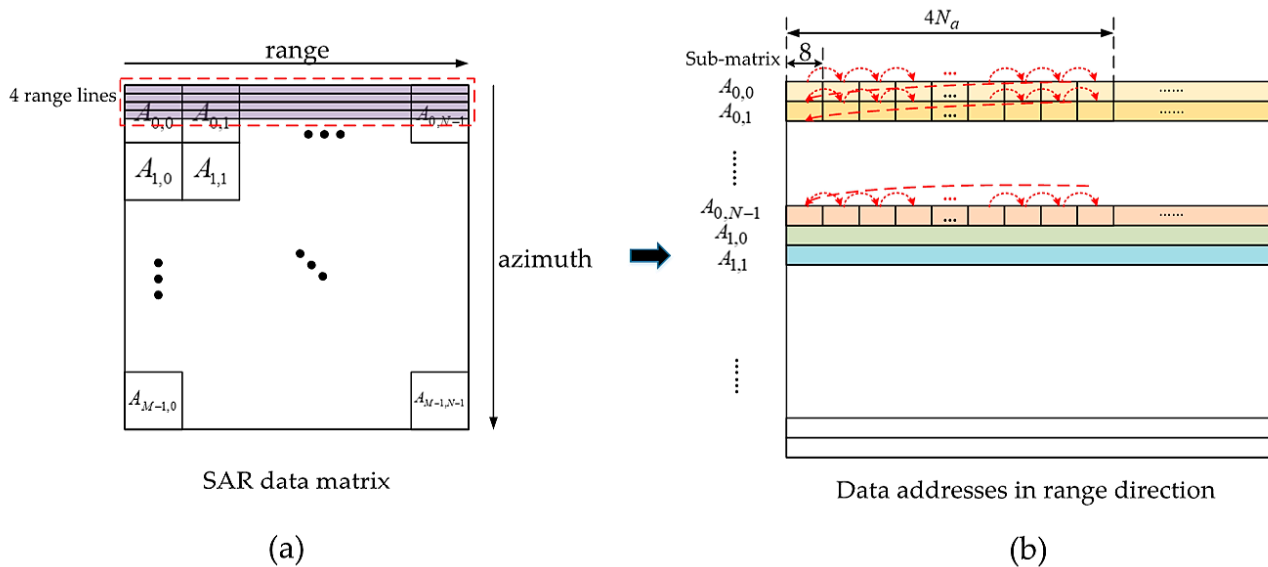
$$\begin{cases} i = \text{floor}((m \cdot N + n) / B_n) \\ j = \text{floor}[(x - m \cdot N_a) / 2] \cdot 2N_r + 2(y - n \cdot N_r) + \text{mod}[(x - m \cdot N_a), 2] \\ k = \text{mod}(m + n, B_n) \end{cases} \quad (8)$$

where  $i$  is the row address,  $j$  is the column address,  $k$  is the bank address, “floor” means round down, and “mod” means take the remainder.

### 3.3. De-Cross Access and Caching Method

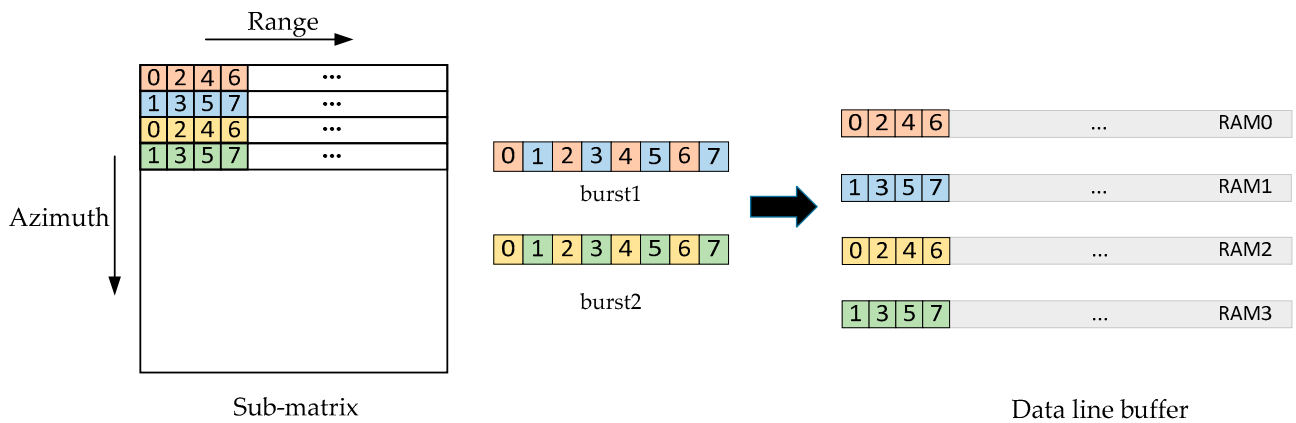
#### 3.3.1. Range Data Access and Caching

In the range access mode, range lines are accessed one by one according to their order in the SAR data matrix. As the burst length is eight, each access can obtain data of two range lines. Since a sub-matrix datum with a size of  $N_a \times N_r$  is mapped to a row of DDR3, there are  $N_a$  data in each row of DDR3 at the same range line, and the data of two adjacent range lines are cross-stored in DDR3. When reading the data, read  $4N_a$  consecutive data in a row of DDR3 (the  $4N_a$  data belong to four consecutive range lines in the SAR data matrix), and then jump to the next row to read the remaining data on the four range lines until all data on these four lines have been read, as shown in Figure 9. When the range data are written back to DDR3 after being processed by the processing engine, the same address change method is used.



**Figure 9.** Data access address change method in range direction. (a) The four range lines to be accessed in the SAR data matrix; (b) the physical addresses of the four range lines to be accessed in DDR3 (each cell represents eight burst transmission data in DDR3; the red dotted line represents the data access sequence in DDR3).

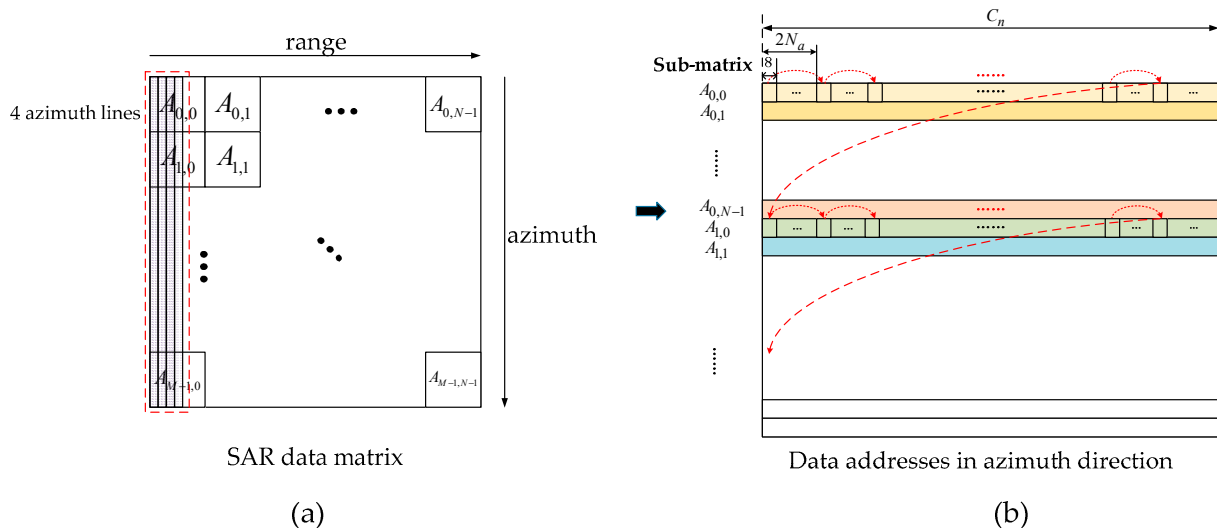
Since the data of two adjacent range lines are cross-stored in DDR3, among the eight data obtained by each burst transmission, two adjacent data belong to different range lines. When performing data caching, the burst data corresponding to the first two range lines needs to be decross-mapped to RAM0 and RAM1, respectively, and the burst data corresponding to the next two range lines need to be decross-mapped to RAM2 and RAM3, respectively, as shown in Figure 10. Therefore, that the data storage order in the cache RAMs is consistent with the SAR data matrix, which is more convenient for the processing engine.



**Figure 10.** De-cross access and caching in range direction.

### 3.3.2. Azimuth Data Access and Caching

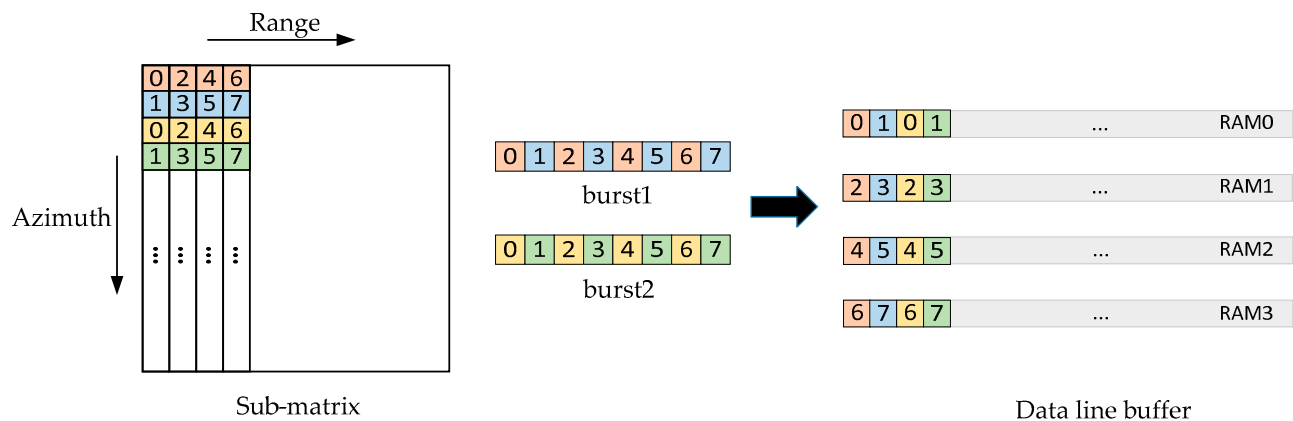
In the azimuth access mode, azimuth lines are accessed one by one according to the order in the SAR data matrix. As the burst length is eight, each burst transmission can obtain the data of four azimuth lines. Since a sub-matrix datum with a size of  $N_a \times N_r$  is mapped to a row of DDR3,  $N_r$  data in each row of DDR3 are on the same azimuth line, and the data of four adjacent azimuth lines are cross-stored in DDR3. When reading the data, read the  $4N_r$  data in one row of DDR3 (the  $4N_r$  data belong to the four consecutive azimuth lines in the SAR data matrix), and then jump to the next row to read the remaining data on these four azimuth lines until all data on these four lines have been read, as shown in Figure 11. The distance between the addresses of the two adjacent burst transmissions of the same azimuth line is  $2N_a$  columns. After reading a row, jump to the corresponding row to read the remaining data in this azimuth direction until the data of these four azimuth lines have been read. When the azimuth data are written back to DDR3 after being processed by the computing unit, the same address jump method is used.



**Figure 11.** Data access address change method in azimuth direction. (a) The four azimuth lines to be accessed in the SAR data matrix; (b) the physical addresses of the four azimuth lines to be accessed in DDR3 (each cell represents eight burst transmission data in DDR3; the red dotted line represents the data access sequence in DDR3).

Since the four adjacent azimuth lines are cross-stored in DDR3, among the eight data obtained by each burst transmission, each pair of adjacent data belongs to a different azimuth line. When performing data caching, data needs to be de-cross-mapped to RAM0,

RAM1, RAM2 and RAM3, as shown in Figure 12. Therefore, that the data storage order in the cache RAMs is consistent with the SAR data matrix, which is convenient for the processing engine.



**Figure 12.** De-cross access and caching in azimuth direction.

Taking a  $16,384 \times 16,384$  SAR matrix as an example, the size of the sub-matrix is  $32 \times 32$ . According to the above method, 128 data must be read in each row for range data access mode or for azimuth data access mode in DDR3. Thus, each row needs to complete 16 burst data transmissions, before proceeding to the next row. As the matrix block three-dimensional mapping method proposed in Section 3.1 uses the priority access feature of cross-bank data in DDR3, there is no interval between the precharge command and the next active command, so there is no  $t_{RP}$  in the efficiency calculation equations. The reading and writing efficiency when using the access method proposed in range access mode or in azimuth access mode can be calculated as follows:

$$\eta_{WRITE\_range\_3D} = \frac{16 \times 4t_{ck}}{t_{RCD} + t_{CWL} + 16 \times 4t_{ck} + t_{WR}} = 67\% \quad (9)$$

$$\eta_{READ\_range\_3D} = \frac{16 \times 4t_{ck}}{t_{RCD} + t_{RTP} + 16 \times 4t_{ck}} = 82\% \quad (10)$$

From the comparison results in Table 2, the method proposed can achieve complete equilibrium in terms of the two-dimensional data bandwidth. Although the range data efficiency is slightly reduced, the azimuth data reading efficiency and writing efficiency are increased to more than 6.7 and 7.8 times, respectively compared with the conventional method. Thus, the azimuthal data access bandwidth is not a bottleneck restricting the real-time performance of the spaceborne SAR system in the proposed method.

**Table 2.** Comparison of theoretical efficiency between conventional method and the method proposed.

Performance	Conventional Method	Ours
Writing efficiency(range)	92.25%	67%
Reading efficiency(range)	94.64%	82%
Writing efficiency(azimuth)	8.51%	67%
Reading efficiency(azimuth)	12.12%	82%

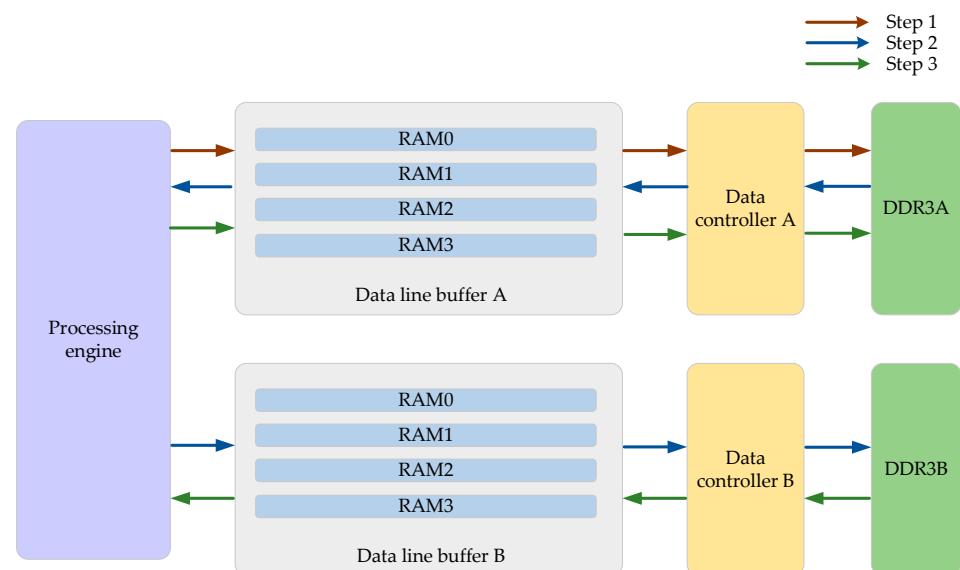
### 3.4. Dual-Channel Pipeline Processing

With the continuous development of integrated circuit technology, the volume, power consumption, and reliability of memory chips have improved significantly. SAR imaging systems often have dual-channel memory units. “Dual-channel” means that two different

data channels can be addressed and accessed separately without affecting each other. This section mainly discusses the dual-channel pipeline processing method.

The dual-channel pipeline processing method requires independent dual-channel DDR3 (hereinafter referred to as DDR3A and DDR3B), and a dual-channel cache RAM that matches the dual-channel DDR3. The schematic diagram of dual-channel pipeline processing is shown in Figure 13; the four steps of this procedure are as follows:

- Step 1: Store the SAR raw data in DDR3A using the mapping method proposed;
- Step 2: Read the required data from DDR3A and save them to RAM for caching. Send data to the processing engine for calculation when the RAM is full; after the calculation is completed, the data are written into the RAM corresponding to DDR3B, and the next batch of data to be processed is read from DDR3A at the same time, so as to create a loop until the entire SAR data matrix is processed. The processing results are all stored in DDR3B;
- Step 3: Read the required data from DDR3B and save them to the RAM for caching. Send data to the processing engine for calculation when the RAM is full; after the calculation is completed, the data are written into the RAM corresponding to DDR3A, and the next batch of data to be processed is read from DDR3B at the same time, so as to create a loop until the entire SAR data matrix is processed. The processing results are all stored in DDR3A;
- Step 4: Repeat Step 2 and Step 3 until all calculations related to the SAR data matrix in the SAR algorithm flow are completed.



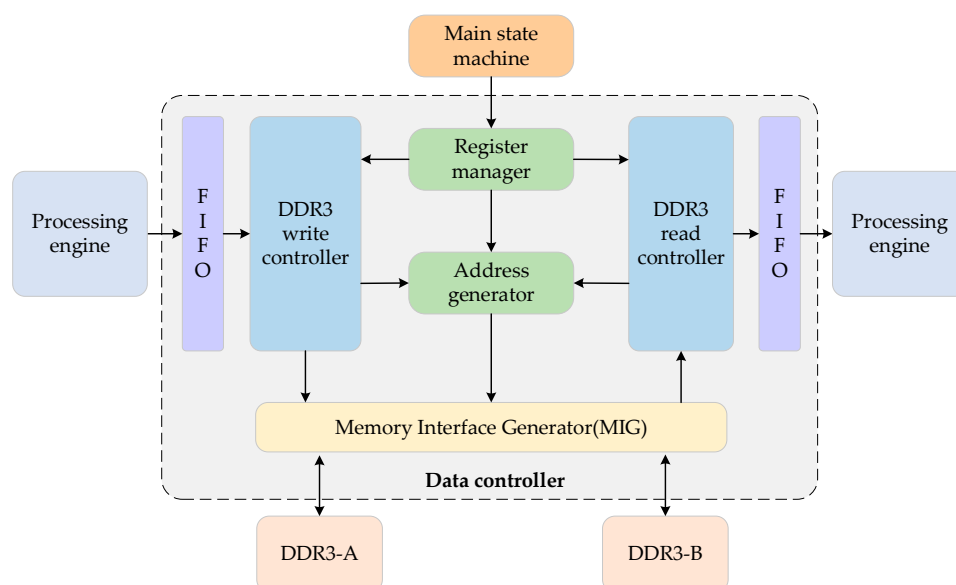
**Figure 13.** Schematic diagram of dual-channel pipeline processing.

The dual-channel pipeline processing method has higher processing efficiency compared with the single-channel processing method. Assuming that the time for DDR3 to read a piece of data is  $t_{rd}$ , the time for writing a piece of data is  $t_{wr}$ , and the processing time for the processing engine to calculate a piece of data is  $t_p$ . In the single-channel processing method, the data need to be written back to DDR3 after the calculation is completed, and then the next batch of data can be read. The total processing time of the entire data matrix is  $N \cdot (t_{rd} + t_p + t_{wr})$ , where  $N$  is the number of batches of processed data. In dual-channel pipeline processing, the reading and writing of the two channels can be performed at the same time. At this time, the total processing time of the entire data matrix is  $t_{rd} + N \cdot (t_p + t_{wr})$ , which is significantly improved compared with the single-channel processing method.



#### 4. Hardware Implementation

Based on the method proposed in Section 3, this paper proposes a dual-channel data reading and writing controller, through which a two-dimensional read / write operation can be carried out from any position in the data matrix, and the bandwidth equilibrium of two-dimensional data access can be guaranteed to meet the efficient matrix transposition needs of SAR imaging. The overall structure of the controller, as shown in Figure 14, consists of a register manager, an address generator, a DDR3 read/write module, and an input/output First-In-First-Out (FIFO) module. The controller invokes the Memory Interface Generator (MIG) IP core to control the dual-channel DDR3.

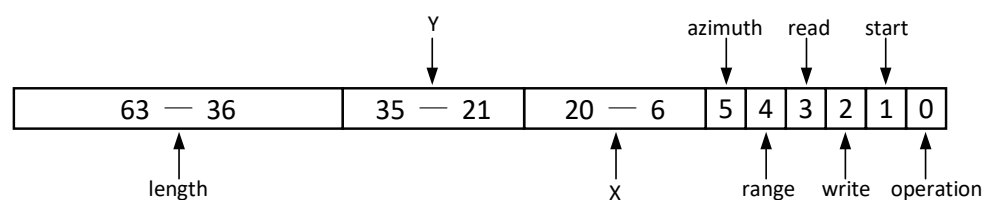


**Figure 14.** The overall architecture of the proposed data controller.

The data read or written from the controller are often input to the buffer RAM, so they work under the system working clock  $\text{sys\_clk}$ . The read or write operation between the controller and DDR3 is performed under the user clock  $\text{ui\_clk}$  provided by the MIG IP core. Under normal circumstances, there is a difference between  $\text{sys\_clk}$  and  $\text{ui\_clk}$ , so it is necessary to use FIFO for cross-clock domain processing.

##### 4.1. Register Manager

When using this controller to complete read or write operation, four parameters need to be provided: the data access starting position, data access length, data access direction (range/azimuth), and data access mode (read/write). In the proposed controller, the design is optimized by using the register to complete read or write operation of any length and direction starting anywhere by hooking up a 64-bit register. The specific definition of each bit of the register is as shown in Figure 15 and Table 3:



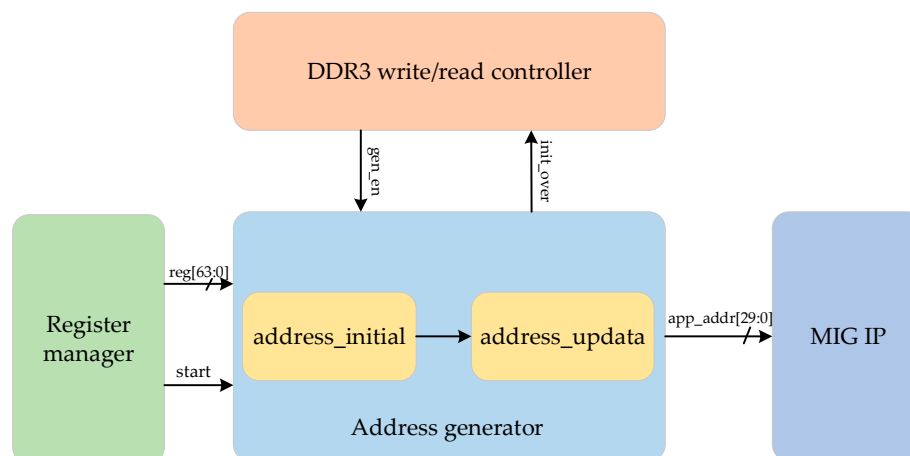
**Figure 15.** The arrangement of the register.

**Table 3.** Bit definition of the register.

Bit	Meaning	Type	Definition
0	work state	read only	0 = idle; 1 = work
1	start enable	read/write	0 = wait; 1 = start
2	writing operation	read/write	0 = invalid, 1 = valid
3	reading operation	read/write	0 = invalid, 1 = valid
4	range direction	read/write	0 = invalid, 1 = valid
5	azimuth direction	read/write	0 = invalid, 1 = valid
20 downto 6	start position (x)	read/write	15-bit binary numbers represent the starting position (x)
35 downto 21	start position (y)	read/write	15-bit binary numbers represent the starting position (y)
63 downto 36	reading/writing length	read/write	28-bit binary numbers represent the read/write length

#### 4.2. Address Generator

The address generator generates the physical address (i.e., row address, column address and bank address) of the data through the address-mapping rules described in Section 3 and outputs it this address to the MIG IP core to complete the data reading / writing operation, as shown in Figure 16. The calculation method for the address-mapping rules is very complicated—if each data address is calculated with a complete formula, this will produce a large delay, which will affect the real-time performance of the hardware system. In order to solve the above problems, in the address generator, the address generation processing can be simplified by using the physical address change rules for the data in the matrix, and the address generation operation can be accomplished efficiently by simple judgment, addition and subtraction. The module consists of two sub-modules: the address initialization module and the address update module.

**Figure 16.** The architecture of the address generator.

The address initialization module calculates the physical address of the starting position by the parameters provided in the register manager, and calculates the intermediate parameters required for subsequent address generation in the range or the azimuth working mode. The address update module includes a range direction address update module and an azimuth direction address update module that continuously updates the physical address to be accessed and feeds it into the MIG IP core based on the starting physical address calculated by the address initialization module and the required intermediate parameters. Due to the difference between the range direction and azimuth direction in the address calculation method, it is necessary to build two corresponding sub-modules. The process of address generation can be represented by pseudo-codes, as shown in Algorithm 1:

**Algorithm 1.** The address generation algorithm.

---

```

1:   Address_initial calculate row_addr, col_addr, bank_addr, subrange_cnt, subazimuth_cnt;
2:   If mod=range, then
3:     Address_initial calculate range_parameters;
4:   Else if mod=azimuth, then
5:     Address_initial calculate azimuth_parameters;
6:   End if
7:   If gen_en=1, then //gen_en is an address request signal
8:     If mod=range, then
9:       Address_updata update row_addr, col_addr, bank_addr;
10:    Else if mod=azimuth, then
11:      Address_updata update row_addr, col_addr, bank_addr;
12:    End if
13:  End if

```

---

**4.3. DDR3 Read/Write Module**

The DDR3 read/write module is connected to the register manager, the FIFO module and the address generator within the controller. First, the module determines the mode of data access (e.g., read/write, range/azimuth) by reading the data of each bit of the register from the register manager, and obtains relevant data parameters (e.g., starting coordinates, read/write length). Secondly, the module reads the data required for a burst transmission from the FIFO and obtains the corresponding address signal from the address generator. Finally, the dual-channel DDR3 data reading and writing is completed by establishing the correct MIG IP user interface timing. The module consists of two parts: the DDR3 write sub-module and the DDR3 read sub-module.

In the DDR3 write sub-module, the values in the register are constantly polled, and the write data process is initiated when the user issues a write data instruction. During the writing process, the write module obtains eight data from the input FIFO as a burst data transmission. After that, an address request signal is sent to the address generator to generate the write address. When the write address is output correctly, it is connected to the address signal of the MIG, and the control signal in the MIG is set up with the correct timing to perform the write operation. When all write operations are completed, a wr\_over signal is sent to the register manager, the value of the register is zeroed, and the write process is completed.

In the DDR3 read sub-module, the values in the register are constantly polled, and the read data process is initiated when the user issues a read data instruction. In the read process, an address request signal is sent to the address generator, requiring a read address to be generated. When the read address is output correctly, it is connected to the address signal of the MIG and the control signal is set with the correct timing. When accepting a valid signal to read data from MIG, eight continuous data from MIG, which can be transmitted at one time, are written to the FIFO. When all read operations are completed, an rd\_over signal is sent to the register manager, the value of the register is zeroed, and the read process is completed.

**5. Experiments and Results**

In this section, extensive experiments were conducted to evaluate the performance of the proposed optimized data storage and access method and hardware data controller. The evaluation experiments were divided into two parts. First, the hardware data controller based on the optimized dual-channel data mapping method proposed in Section 4 is implemented in FPGA. Then, the processing performance of the SAR data matrix with the same granularity as the spaceborne conditions is tested. The experimental settings and detailed experimental results are described in the following subsections.

### 5.1. Experimental Settings

In order to verify the efficiency of the method proposed in this paper, we implemented a data controller based on a Xilinx Virtex-7 FPGA chip (Xilinx, San Jose, CA, USA). The hardware platform is a processing board equipped with a Xilinx XC7VX690T FPGA (Xilinx, San Jose, CA, USA) and two clusters of Micron MT41K512M16HA-125 DDR3 SDRAM (Micron Technology, Boise, ID, USA) particles. Each cluster of DDR3 SDRAM has a capacity of 8 GB and a bitwidth of 64 bit. A SAR data matrix with a granularity of  $16,384 \times 16,384$  is used, and the data is a complex number of single-precision floating-point numbers (the bit width is 64 bit). Two independent data controllers are implemented in FPGA, and two clusters of DDR3 are connected.

The experiment was completed in the Vivado 2018.3 software with the Very-High-Speed Integrated Circuit Hardware Description Language (VHDL), and the Xilinx MIG v4.2 IP core (Xilinx, San Jose, CA, USA) was used. The parameters used in the experiment are as follows: the working clock of DDR3 is 800 MHz, the working clock of the data controller is 200 MHz, the working clock of the processing engine is 100 MHz, and the DDR3 burst transmission length is eight.

The experimental procedure is designed according to the standard CS algorithm. In the first step, the original data are stored in DDR3A by FPGA according to the range direction. The second step is to read data from DDR3A in the azimuth direction, and store them in DDR3B in the azimuth direction after caching. The third step is to read data from DDR3B in the range direction and, after caching, write these data to DDR3A in the range direction. The fourth step is to read data from DDR3A in the azimuth direction. After completing all the above steps, we should compare whether the final read data is the same as the original data, measure the data bandwidth of each step and record it.

### 5.2. Hardware Resource Utilization

Through the synthesis and implementation steps in Vivado 2018.3, the resource utilization of the data controller in the FPGA can be obtained. The results are shown in Table 4:

**Table 4.** Hardware resource utilization of the data controller proposed.

Resource	Utilization	Available	Utilization (%)
Slice LUTs	27,369	433,200	6.32%
Slice Registers	20,514	866,400	2.37%
Block RAM	60	1470	4.08%
DSPs	0	3600	0.00%

The table shows the utilization of the on-chip resources of LUT, the register, RAM and DSP. The total number of available resources is the total number of these resources in Xilinx XC7VX690T FPGA, and the resource utilization (%) in the table is the utilization number for each resource divided by the total number of resources. It can be seen that the controller only occupies a small amount of LUT and storage resources. Since the address generation process is optimized in the controller, the address update can be completed only by judging, shifting, adding and subtracting, without using the DSP resources on the FPGA chip, so the utilization rate of DSP is 0. More FPGA computing resources are used for the system's computing engine, which can further improve the system's computing capabilities.

### 5.3. Performance Evaluation of the Data Controller

We deployed the proposed hardware architecture on the FPGA to demonstrate its efficiency. The peak theoretical bandwidth of the DDR3 chip with a bit width of 64bit and a working clock of 800 MHz in the existing processing board is:

$$B = 800M \times 64 \times 2/8 = 12.8GB/s \quad (11)$$

When considering the efficiency loss caused by row activation and precharging, the actual bandwidth is:

$$B_{\eta} = B \times \eta \quad (12)$$

where  $\eta$  is the efficiency.

We recorded the actual measured access bandwidth and included it in the above equation to calculate the access efficiency, providing the indicators listed in Table 5. Compared with the theoretical calculation results in Section 3.3.2, the experimental data bandwidths are slightly lower than the theoretical results due to the hardware communication overhead.

**Table 5.** Measured bandwidth of the method proposed.

	Range (Read)	Range (Write)	Azimuth (Read)	Azimuth (Write)
Theoretical bandwidth	12.8 GB/s	12.8 GB/s	12.8 GB/s	12.8 GB/s
Measured bandwidth	10.24 GB/s	8.45 GB/s	10.24 GB/s	8.45 GB/s
Efficiency	80%	66%	80%	66%

The FPGA needs to use dual-port RAM as a data buffer between DDR3 and the processing engine. In the experiment, the burst transmission length of DDR3 is eight. According to the sub-matrix cross-storage method, the utilization rate of DDR3 burst transmission data is 100%. For data access in the range direction, the read  $8 \times 64$  bit data belong to two range lines, so two RAMs are needed to cache the data. For data access in the azimuth direction, the read  $8 \times 64$  bit data belong to four azimuth lines, so four RAMs are needed to cache the data. In order to take into account both the range and azimuth access requirements, four dual-port RAMs should be included in the system design.

The recent FPGA-based implementations were also compared in this section. The comparison results of the proposed data controller and several recent FPGA-based implementations are presented in Table 6. The “range” and “azimuth” in the table represent the performance during the range direction access and azimuth direction access, respectively. “Matrix transposition time” refers to the time for reading in the range direction and then writing in the azimuth direction, which can be calculated by the DDR3 access bandwidth. The experimental results show that the data controller proposed achieves a very high data bandwidth in both the range and azimuth directions. As the data controller proposed has the characteristics of independently controlling the dual-channel storage unit, it has strong flexibility and can complete the read and write tasks separately, making the processing engine twice the throughput rate. Compared with the previous FPGA-based implementations, the data controller proposed has a higher data bandwidth and greater flexibility in terms of its design, while ensuring higher access efficiency, leading to a trade-off between resources and speed, which is very suitable for real-time SAR processing systems in spaceborne scenarios.

**Table 6.** Comparisons with previous implementations.

	Ours	[34]	[33]	[32]	[31]	[30]
Data Granularity	$16,384 \times 16,384$	$16,384 \times 16,384$	$16,384 \times 16,384$	$16,384 \times 16,384$	$8192 \times 8192$	$4096 \times 4096$
FPGA	Xilinx XC7VX690T	Xilinx XC7VX690T	Xilinx XC6VLX315T	Xilinx XC6VSX760T	Xilinx XC7VX330T	Altera DE4
DDR3 channel number	2	2	1	1	1	2
Range access bandwidth	10.24 GB/s	8.3 GB/s	4.8 GB/s	6.0 GB/s	-	-
Azimuth access bandwidth	10.24 GB/s	9.62 GB/s	4.8 GB/s	2.37 GB/s	-	-
Range access efficiency	80%	69%	74%	93.75%	-	83%
Azimuth access efficiency	80%	80%	74%	74%	-	83%
Matrix transposition time	0.43 s	0.45 s	0.83 s	1.18 s	0.33 s	-
Cache RAM number	4	4	4	8	-	-
Pipeline processing supported or not	Yes	No	No	No	No	No

## 6. Conclusions

In this paper, a dual-channel data storage and access method is designed and implemented for spaceborne SAR real-time processing. The method proposed ensures that the data bandwidths in the range direction and azimuth direction achieve complete equilibrium, and reduce the use of on-chip cache resources. In addition, based on the above method, a dual-channel data controller is implemented by using Xilinx XC7VX690T FPGA. The experimental results show that the read data bandwidths in the range direction and azimuth direction can both reach 10.24 GB/s, the reading efficiency is 80% which an increase of more than 6.7 times that of the conventional method. Additionally, write data bandwidths in the range direction and azimuth direction can both reach 8.45 GB/s, the writing efficiency is 66% which an increase of more than 7.8 times that of the conventional method. This paper also compares several existing FPGA-based implementations to verify the superiority of the data controller proposed. The method proposed can also be extended to other two-dimensional image real-time processing scenarios, such as FPGA-based Uninhabited Aerial Vehicle Synthetic Aperture Radar (UAVSAR) [18,39] and Convolutional Neural Networks (CNNs) [40–42], etc. In the future, we will conduct research on multi-channel SAR data processing, multi-mode integrated SAR processing, and new efficient algorithms that are suitable for real-time processing in order to meet new remote sensing application requirements.

**Author Contributions:** Conceptualization, G.W. and H.C.; methodology, G.W. and Y.X.; software, G.W., H.C. and Y.X.; validation, G.W.; formal analysis, G.W.; investigation, G.W. and Y.X.; resources, H.C. and Y.X.; writing—original draft preparation, G.W.; writing—review and editing, G.W., H.C. and Y.X.; supervision, Y.X.; project administration, H.C.; funding acquisition, H.C. and Y.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant No. 91738302.

**Acknowledgments:** This work was supported by the Chang Jiang Scholars Program under Grant T2012122 and the Hundred Leading Talent Project of Beijing Science and Technology under Grant Z141101001514005.

**Conflicts of Interest:** The authors declare no conflict of interest.



## References

- Wu, C.; Liu, K.Y.; Jin, M. Modeling and a correlation algorithm for spaceborne sar signals. *IEEE Trans. Aerosp. Electron. Syst.* **1982**, *5*, 563–575. [CrossRef]
- Raney, R.; Runge, H.; Bamler, R.; Cumming, I.; Wong, F. Precision SAR processing using chirp scaling. *IEEE Trans. Geosci. Remote Sens.* **1994**, *32*, 786–799. [CrossRef]
- Long, T.; Zeng, T.; Hu, C.; Dong, X.; Chen, L.; Liu, Q.; Xie, Y.; Ding, Z.; Li, Y.; Wang, Y.; et al. High resolution radar real-time signal and information processing. *China Commun.* **2019**, *16*, 105–133. [CrossRef]
- Moreira, A.; Prats-Iraola, P.; Younis, M.; Krieger, G.; Hajnsek, I.; Papathanassiou, K.P. A tutorial on synthetic aperture radar. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–43. [CrossRef]
- Gierull, C.H.; Vachon, P.W. Foreword to the special issue on multichannel space-based SAR. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4995–4997. [CrossRef]
- Hirose, A.; Rosen, P.A.; Yamada, H.; Zink, M. Foreword to the special issue on advances in SAR and radar technology. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3748–3750. [CrossRef]
- Tralli, D.M.; Blom, R.G.; Zlotnicki, V.; Donnellan, A.; Evans, D.L. Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards. *ISPRS J. Photogramm. Remote Sens.* **2005**, *59*, 185–198. [CrossRef]
- Joyce, K.E.; Belliss, S.E.; Samsonov, S.V.; McNeill, S.J.; Glassey, P.J. A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. *Prog. Phys. Geogr. Earth Environ.* **2009**, *33*, 183–207. [CrossRef]
- Percivall, G.S.; Alameh, N.S.; Caumont, H.; Moe, K.L.; Evans, J.D. Improving disaster management using earth observations—GEOSS and CEOS Activities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 1368–1375. [CrossRef]
- Bernstein, R.; Cardone, V.; Katsaros, K.; Lipes, R.; Riley, A.; Ross, D.; Switft, C. GOASEX workshop results from the seasat-1 scanning multichannel microwave radiometer. In Proceedings of the Symposium on Petroleum Potential in Island Arcs, Small Ocean Basins, Submerged Margins and Related Areas, Suva, Fiji, 18–21 September 1979; p. 657. [CrossRef]
- Gaofen-3 (GF-3) SAR Satellite/CHEOS Series of China. Available online: <https://directory.eoportal.org/web/eoportal/satellite-missions/g/gaofen-3> (accessed on 4 February 2021).
- Copernicus: Sentinel-1—The SAR Imaging Constellation for Land and Ocean Services. Available online: <https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/copernicus-sentinel-1> (accessed on 4 February 2021).
- RCM (RADARSAT Constellation Mission). Available online: <https://directory.eoportal.org/web/eoportal/satellite-missions/r/rcm> (accessed on 4 February 2021).
- Mohr, D.; Doubleday, J. NISAR's unique challenges and approach to Robust JPL/ISRO joint operations. In Proceedings of the 2020 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2020; pp. 1–12.
- TSX-NG (TerraSAR-X Next Generation). Available online: <https://directory.eoportal.org/web/eoportal/satellite-missions/t/tsx-ng> (accessed on 4 February 2021).
- Gantert, S.; Kern, A.; Düring, R.; Janoth, J.; Petersen, L.; Herrmann, J. The future of X-band SAR: TerraSAR-X next generation and WorldSAR constellation. In Proceedings of the 2013 Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), Tsukuba, Japan, 23–27 September 2013; pp. 20–23.
- Bierens, L.; Vollmuller, B. On-board Payload Data Processor (OPDP) and its application in advanced multi-mode, multi-spectral and interferometric satellite SAR instruments. In Proceedings of the EUSAR 2012 9th European Conference on Synthetic Aperture Radar, Nuremberg, Germany, 23–26 April 2012; pp. 340–343.
- Lou, Y.; Clark, D.; Marks, P.; Muellerschoen, R.J.; Wang, C.C. Onboard radar processor development for rapid response to natural hazards. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 2770–2776. [CrossRef]
- Schmidt, A.G.; Weisz, G.; French, M.; Flatley, T.; Villalpando, C.Y. SpaceCubeX: A framework for evaluating hybrid multi-core CPU/FPGA/DSP architectures. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–10.
- Geist, A.; Brewer, C.; Davis, M.; Franconi, N.G.; Heyward, S.; Wise, T.; Crum, G.; Petrick, D.; Ripley, R.; Wilson, C.; et al. SpaceCube v3.0 NASA next-generation high-performance processor for science applications. In Proceedings of the 33rd Annual AIAA/USU Conference on Small Satellites, Logan, UT, USA, 3–9 August 2019.
- Brewer, C.; Franconi, N.G.; Ripley, R.; Geist, A.; Wise, T.; Sabogal, S.; Crum, G.; Heyward, S.; Wilson, C. NASA SpaceCube intelligent multi-purpose system for enabling remote sensing, communication, and navigation in mission architectures. In Proceedings of the 34th Annual Small Satellite Conference, Logan, UT, USA, 1–6 August 2020.
- Prototype Design for NASA SpaceCube Intelligent Multi-Purpose System for Enabling Remote Sensing, Communication, and Navigation (SpaceCube IMPS). Available online: <https://techport.nasa.gov/view/96769> (accessed on 4 February 2021).
- Schmidt, A.G.; French, M.; Flatley, T. Radiation hardening by software techniques on FPGAs: Flight experiment evaluation and results. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–8.
- Yang, Z.Y.Z.; Long, T.L.T. Methods to improve system verification efficiency in FPGA-based spaceborne SAR image processing system. In Proceedings of the IET International Radar Conference 2015, Hangzhou, China, 14–16 October 2015; pp. 1–5.
- Ding, Z.; Xiao, F.; Xie, Y.; Yu, W.; Yang, Z.; Chen, L.; Long, T. A modified fixed-point chirp scaling algorithm based on updating phase factors regionally for spaceborne SAR real-time imaging. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 7436–7451. [CrossRef]
- DDR3 SDRAM. Available online: <https://www.micron.com/products/dram/ddr3-sdram> (accessed on 4 February 2021).

27. Wang, B.; Du, J.; Bi, X.; Tian, X. High bandwidth memory interface design based on DDR3 SDRAM and FPGA. In Proceedings of the 2015 International SoC Design Conference (ISOCC), Gyungju, Korea, 2–5 November 2015; pp. 253–254.
28. Guoteng, P.; Li, L.; Guodong, O.; Qiang, D.; Lunguo, X. Design and Implementation of a DDR3-based Memory Controller. In Proceedings of the 2013 Third International Conference on Intelligent System Design and Engineering Applications, Hong Kong, 16–16 January 2013; pp. 540–543.
29. NTRS-NASA Technical Reports Server. Double Data Rate (DDR) Memory Devices. Available online: <https://ntrs.nasa.gov/citations/20180004227> (accessed on 4 February 2021).
30. Akin, B.; Franchetti, F.; Hoe, J.C. FFTS with near-optimal memory access through block data layouts. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 3898–3902.
31. Garrido, M.; Pirsch, P. Continuous-Flow Matrix Transposition Using Memories. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 3035–3046. [[CrossRef](#)]
32. Yang, C.; Li, B.; Chen, L.; Wei, C.; Xie, Y.; Chen, H.; Yu, W. A Spaceborne synthetic aperture radar partial fixed-point imaging system using a field-programmable gate array—Application-specific integrated circuit hybrid heterogeneous parallel acceleration technique. *Sensors* **2017**, *17*, 1493. [[CrossRef](#)] [[PubMed](#)]
33. Li, B.; Shi, H.; Chen, L.; Yu, W.; Yang, C.; Xie, Y.; Bian, M.; Zhang, Q.; Pang, L. Real-time spaceborne synthetic aperture radar float-point imaging system using optimized mapping methodology and a multi-node parallel accelerating technique. *Sensors* **2018**, *18*, 725. [[CrossRef](#)] [[PubMed](#)]
34. Sun, T.; Xie, Y.; Li, B. Efficiency balanced matrix transpose method for sliding spotlight SAR imaging processing. *J. Eng.* **2019**, *2019*, 7775–7778. [[CrossRef](#)]
35. Virtex-7. Available online: <https://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html> (accessed on 4 February 2021).
36. Yang, C.; Wei, C.; Xie, Y.; Chen, H.; Ma, C. Area-efficient mixed-radix variable-length FFT processor. *IEICE Electron. Express* **2017**, *14*, 20170232. [[CrossRef](#)]
37. Yang, C.; Xie, Y.; Chen, H. A novel word length optimization method for radix-2 k fixed-point FFT. *Sci. China Inf. Sci.* **2017**, *61*, 1–2. [[CrossRef](#)]
38. MT41K512M16HA-125. Available online: <https://www.micron.com/products/dram/ddr3-sdram/part-catalog/mt41k512m16ha-125> (accessed on 4 February 2021).
39. Gong, J.L. Development of integrated electronic system for SAR based on UAV. In Proceedings of the 2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), Xiamen, China, 26–29 November 2019; pp. 1–4.
40. Chang, K.-W.; Chang, T.-S. Efficient accelerator for dilated and transposed convolution with decomposition. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 10–21 October 2020; pp. 1–5.
41. Zhang, X.; Wei, X.; Sang, Q.; Chen, H.; Xie, Y. An efficient fpga-based implementation for quantized remote sensing image scene classification network. *Electronics* **2020**, *9*, 1344. [[CrossRef](#)]
42. Zhang, N.; Wei, X.; Chen, H.; Liu, W. FPGA Implementation for CNN-based optical remote sensing object detection. *Electronics* **2021**, *10*, 282. [[CrossRef](#)]