An Efficient On-Chip Data Storage and Exchange Engine for Spaceborne SAR System

Hushan Lv, Yongrui Li, Yizhuang Xie * and Tingting Qiao

Beijing Key Laboratory of Embedded Real-Time Information Processing Technology, Beijing Institute of Technology, Beijing 100081, China; hushanlv@bit.edu.cn (H.L.); 3120215415@bit.edu.cn (Y.L.); ttq@bit.edu.cn (T.Q.)

* Correspondence: xyz551_bit@bit.edu.cn; Tel.: +86-136-935-19576

Abstract: Advancements in remote sensing technology and very-large-scale integrated circuit (VLSI) have significantly augmented the real-time processing capabilities of spaceborne synthetic aperture radar (SAR), thereby enhancing terrestrial observational capacities. However, the inefficiency of voluminous data storage and transfer inherent in conventional methods has emerged as a technical hindrance, curtailing real-time processing within SAR imaging systems. To address the constraints of a limited storage bandwidth and inefficient data transfer, this study introduces a three-dimensional cross-mapping approach premised on the equal subdivision of sub-matrices utilizing dual-channel DDR3. This method considerably augments storage access bandwidth and achieves equilibrium in two-dimensional data access. Concurrently, an on-chip data transfer approach predicated on a superscalar pipeline buffer is proposed, mitigating pipeline resource wastage, augmenting spatial parallelism, and enhancing data transfer efficiency. Building upon these concepts, a hardware architecture is designed for the efficient storage and transfer of SAR imaging system data, based on the superscalar pipeline. Ultimately, a data storage and transfer engine featuring register addressing access, configurable granularity, and state monitoring functionalities is realized. A comprehensive imaging processing experiment is conducted via a “CPU + FPGA” heterogeneous SAR imaging system. The empirical results reveal that the storage access bandwidth of the proposed superscalar pipeline-based SAR imaging system’s data efficient storage and transfer engine can attain up to 16.6 GB/s in the range direction and 20.0 GB/s in the azimuth direction. These findings underscore that the storage exchange engine boasts superior storage access bandwidth and heightened data storage transfer efficiency. This considerable enhancement in the processing performance of the entire “CPU + FPGA” heterogeneous SAR imaging system renders it suitable for application within spaceborne SAR real-time processing systems.

Keywords: synthetic aperture radar (SAR); real-time processing; data storage and exchange; field-programmable gate array (FPGA); superscalar pipeline processing

1. Introduction

Based on the synthetic aperture principle and making use of the Doppler information generated by the relative motion of the radar platform and the target being detected, spaceborne synthetic aperture radar (SAR) is a high-resolution microwave imaging technology [1–4]. It has a variety of characteristics, such as all-time, all-weather, high-resolution large width, and has a certain degree of surface penetration, offering it a unique advantage in applications such as disaster monitoring, marine monitoring, resource surveying, mapping and military [5–9]. Recent publications provide an overview of the application of satellite remote sensing technology to natural hazards such as earthquakes, volcanoes, floods, landslides, and coastal flooding [10–12].

The first spaceborne synthetic aperture radar oceanographic satellite, Seasat-1, was launched by the United States in June 1978, successfully ushering in a new age of space...
Earth observation [13]. Since then, spaceborne SAR imaging has attracted the attention of many countries and has become a highly competitive and rapidly developing technology field. In recent years, it has also become a significant method of Earth observation. For example, on 7 April 2022, China successfully launched the Gaofen-3 03 satellite is a C-band SAR satellite, which mainly acquires reliable and stable high-resolution SAR images to provide operational application data support for China’s marine development, land environmental resource monitoring and emergency disaster prevention and mitigation [14]. The European Space Agency (ESA) launched the SAOCOM-1B satellite in 2020 as part of the SAOCOM-1 SAR satellite mission, which also includes the previously launched SAOCOM-1A satellite. The mission’s objective is to acquire data with high-quality radiometric and geometric accuracy while offering frequent revisits to support specific operational needs. Moreover, it aims to deliver timely information to aid natural and human-caused disaster management, such as regional floods, volcanic eruptions, earthquakes, and landslides [15].

In 2018, ICEYE launched two ICEYE satellites, the first two satellites in the company’s planned satellite constellation, and since 2018 the company has launched 16 satellites and plans to launch more than 10 satellites in 2022 and beyond, forming a world-unprecedented SAR constellation networking system that can provide revisits at several hours level and up to 0.25 m ultra-high resolution [16]. Future spaceborne Synthetic Aperture Radar (SAR) technology will likely advance in several areas, including miniaturization, multi-band, multi-mode, high-resolution capabilities, and ultra-wide imaging.

The traditional method for processing SAR data involves the satellite storing the raw data and then downlinking it to ground equipment for further processing. This approach is referred to as the “satellite data downlink and ground processing” method. As the amount of acquired data geometrically increases, this processing method has low timeliness, which makes it difficult to meet the needs of high-timeliness tasks such as maritime target surveillance, key regional situation changes, earthquake and landslide disaster assessment [17].

In recent years, due to the large-scale development of chip manufacturing technology, researchers have begun to explore real-time processing technology for space-level chips with strong processing capabilities. The utilization of on-orbit processing technology can significantly alleviate the pressure of data transmission between satellites and ground stations, enhance the efficiency of information acquisition, and facilitate prompt decision-making. In 2010, the Netherlands National Aerospace Laboratory (NLR) developed the processor for the on-board payload data processing (OPDP) architecture. OPDP enables the high-speed processing of SAR imaging data at 1K × 1K granularity on the RTAX2000S FPGA through a combination of the general-purpose AT697 LEON2 fault-tolerant processor ASIC and the flexible FFT-oriented DSP coprocessor fast fourier transform coprocessor (FFTC), coupled with configured SDRAM memory and data interaction Switch Fabric modules [18,19]. In 2018, Ruhr University in Germany implemented an SAR imaging system based on the back projection (BP) algorithm on a Xilinx Zynq-7000 chip. In 2019, NASA launched the latest SpaceCube v3.0 space-embedded heterogeneous processing board with different kinds of FPGAs, MPSoC chips, which can perform a variety of on-orbit data real-time processing tasks, providing solutions to meet the needs of next-generation science and defense missions [20,21]. Xilinx released the world’s first 20 nm space-grade Kintex Ultra-Scale field-programmable gate array (FPGA) chip in 2020. This new chip offers enhanced radiation tolerance, extremely high data throughput, and bandwidth performance, making it an excellent choice for space and satellite applications. Moreover, this chip supports artificial intelligence applications, making it possible for satellites to process imagery or radar data locally, without ground review. This innovation provides several benefits, such as improved mission responsiveness and real-time processing capabilities, thereby enabling a new system architecture [22].

The development of technology for the real-time processing of spaceborne SAR demonstrates that a high-performance satellite-based real-time processing platform requires processors with high processing power, low power consumption, and high scalability, as well as memory with a large capacity and high bandwidth. In the direction of real-time process-
Remote Sens. 2023, 15, 2885

ing, central processing unit (CPU), digital signal processor (DSP), field-programmable gate array (FPGA), graphics processing unit (GPU) and application-specific integrated circuit (ASIC) have some degree of superiority [23]. Processors such as embedded multi-core CPUs, which can meet a larger number of applications, have a high processing accuracy and can run operating systems, thus realizing complex embedded control processes, and have flexible designs; however, they have a relatively weak computing power when processing massive amounts of data. Flexibility in programming, high processing precision, multi-chip scalability, ease of development, and low power consumption are all advantages of DSP processors; however, the parallelism is subpar. Despite its high power consumption and lack of suitability for satellite-processing scenarios, GPU is now mostly used in ground stations as a data-processing processor. ASIC is a self-designed processor with a lengthy development time and limited adaptability for fixed algorithm applications. Compared with CPU, DSP and ASIC, FPGA has the advantages of high programmable flexibility, a short development cycle, high parallelism, better performance and higher energy efficiency. Therefore, in recent years, FPGAs have become more and more widely used in the field of on-board real-time processing [24–26]. At present, it is the trend for the real-time processing of SAR imaging to compensate for the shortcomings of a single processor in some specific application scenarios through multi-processor heterogeneous integration [27]. The central processing unit (CPU) in a heterogeneous processor such as the “CPU + FPGA” takes care of scheduling and task management while the FPGA acts as a co-processor to speed up computations [28]. Because the above processors’ on-chip storage capacity is extremely constrained and it is unable to accommodate the demands of massive data storage, SAR data must be stored in a high-speed, large-capacity, off-chip memory. Currently, DDR3 SDRAM can be used as an off-chip memory option for space-borne real-time processing. This has been widely used in the real-time processing platform for satellites [29–32].

Enhancing the real-time imaging processing capability of massive data on satellite has become a technical requirement for spaceborne SAR due to the constrained resource environment on satellite. In the current SAR imaging system research, efficient data exchange and on-chip system architecture design are two of the key technologies. Various traditional SAR imaging algorithms regularly call for several exchange operations. Additionally, due to the limited storage bandwidth caused by the low exchange efficiency of massive data in traditional methods, computation and storage are not matched, which wastes computing power and influences the performance of real-time processing. In recent years, researchers have proposed a variety of methods to solve the problem of the limited storage access bandwidth of two-dimensional matrices, and implemented them on FPGA. A matrix block linear mapping was applied by Yang et al. [33] to improve data access bandwidth; however, the method has a low utilization of eight buffer RAMs and an imbalanced range and azimuth access efficiency. Sun et al. [34] balanced the range and azimuth accesses efficiency using a matrix block cross-mapping method. Wang et al. [35] used a dual-channel pipeline design to implement balanced two-dimensional access efficiency, higher buffer RAM utilization, and improved data exchange efficiency.

This paper presents a potent method for the storage and exchange of synthetic aperture radar (SAR) imaging system data predicated on a superscalar pipeline. This approach addresses the inefficiencies in massive data exchange and limitations in storage access bandwidth that exist between the FPGA computing engine and DDR3 SDRAM. An examination of the Chirp scaling (CS) algorithm elucidated that the row activation and pre-charging of the DDR3 SDRAM memory chip posed the most significant challenges to storage access efficiency due to the two-dimensional matrix transposition operations. Consequently, we propose a three-dimensional cross-mapping approach grounded on the equal division of a sub-matrix with dual-channel DDR3. Further, an analysis of system pipeline processing led to the proposal of an efficient on-chip data exchange solution tailored for SAR imaging processors equipped with a superscalar pipeline buffer. Additionally, we suggest a hardware architecture for an on-chip data exchange engine to enhance hardware efficiency for real-time processing. The primary contributions of this study are as follows:
Through modeling and calculating the time parameters impacting DDR3 access efficiency, such as DDR3 row activation and pre-charging, a three-dimensional cross-mapping method is proposed premised on the equal division of sub-matrix with dual-channel DDR3. While maintaining the three-dimensional cross-mapping method, this approach maximizes the parallelism of two off-chip DDR3 to amplify the range and azimuth data storage access bandwidth, thereby achieving balance in range and azimuth data access.

By analyzing and studying traditional pipeline resources, a superscalar pipeline buffer is proposed as a method for efficient on-chip data exchange for SAR imaging processors. This method subdivides pipeline resources further, diminishes the idle ratio, optimizes spatial parallelism, and leverages the advantages of dual-channel DDR3 storage, thereby markedly augmenting the efficiency of data exchange.

Based on the aforementioned solution, we propose a hardware architecture for an on-chip efficient data exchange engine. This architecture employs a modular register addressing control mode and synthesizes a superscalar pipeline buffer module with a dual-channel DDR3 access control module, thereby forming a data exchange engine with configurable granularity and state monitoring capabilities.

We verified the proposed hardware architecture in a “CPU + FPGA” heterogeneous SAR imaging system, and evaluated the data bandwidth. The experimental results show that the efficient storage and exchange engine for SAR imaging data designed in this paper has a storage access bandwidth of up to 16.6 GB/s in the range direction, which is the read bandwidth in the range direction. In addition, the write access bandwidth in the range direction of DDR3 is 16.0 GB/s. In the azimuth direction, the storage access bandwidth can reach up to 20.0 GB/s, which is read bandwidth in the azimuth, and the write bandwidth in the azimuth is 18.3 GB/s. The method proposed in this paper is better than the existing implementations.

The rest of this paper is organized as follows. Section 2 introduces the SAR imaging algorithm and analyzes the DDR3 data access characteristics, and provides an overview of the currently available data exchange methods. Section 3 provides a detailed introduction of the data storage and exchange method based on superscalar pipeline. The proposed hardware architecture and register configuration will be described in Section 4 of this paper. Section 5 shows the experiments and results. Finally, in Section 6, the conclusions are presented.

2. Analysis of Storage and Exchange Efficiency for SAR Data

2.1. Analysis of Chirp Scaling (CS) Algorithm

In the realm of synthetic aperture radar (SAR) imaging processing, the range Doppler (RD) algorithm and the Chirp scaling (CS) algorithm represent the two most prevalent methodologies. The RD algorithm, characterized by its simplicity and straightforward hardware implementation, is infrequently employed due to its lower processing precision. In contrast, the CS algorithm is primarily composed of fast fourier transform (FFT), inverse fast fourier transform (IFFT), and complex multiplication operations. These can be implemented on hardware platforms such as field-programmable gate arrays (FPGAs), conferring a high processing precision that satisfies the requirements of general SAR imaging accuracy. As such, the CS algorithm has ascended to become a popular choice for the implementation of hardware platforms in spaceborne SAR imaging processing [36,37]. The CS algorithm comprises three stages, detailed as follows:

Step 1: Chirp scaling operation. First, the Doppler center frequency (FDC) is estimated. Then, the raw two-dimensional time domain data of SAR are transformed into the range–Doppler domain by fast fourier transform (FFT) method in the azimuth direction, and multiplied with the first phase function in the range–Doppler domain. Due to the spatial shift characteristics of the range migration curve, the range migration curves of point targets in different range directions have different bending degrees. Thus, the chirp scaling
operation will keep the bent degrees of the range migration curves the same for all range directions [38,39].

Step 2: Range cell migration correction (RCMC), range compression, and secondary range compression (SRC). After processing in Step 1, the two-dimensional matrix in the range–Doppler domain is converted to the two-dimensional frequency domain by the fast fourier transform (FFT) method in the range direction, and is multiplied with the second phase function to complete the range cell migration correction (RCMC) and range compression operations. Then, the data matrix in the two-dimensional frequency domain is transformed into the range–Doppler domain by performing an inverse fast fourier transform (IFFT) operation in the range direction.

Step 3: Azimuth compression. First, the Doppler frequency rate (FDR) is estimated. Then, the two-dimensional matrix in the range–Doppler domain processed in the Step 2 stage is multiplied by a complex number with the third phase function in the azimuth direction to complete the azimuth compression. Finally, the data domain is converted from the range–Doppler domain to the two-dimensional time domain, and this conversion is obtained by inverse fast fourier transform (IFFT) operation in the azimuth direction. After the azimuth compression is completed, the two-dimensional time domain data are quantized to obtain the SAR result image. The CS algorithm is executed. Figure 1 shows a schematic flowchart of CS algorithm processing.

![Figure 1. The schematic flowchart of CS algorithm processing.](image)

### 2.2. Analysis of Storage Access Characteristics for DDR3 SDRAM

As mentioned in Section 1, the massive amount of raw data and the result data from each processing step must be stored during spaceborne SAR imaging processing, and DDR3 SDRAM is the ideal off-chip storage medium for the current spaceborne SAR real-time processing. DDR3 can be regarded as a large warehouse. The specific location of
the stored data must be uniquely determined by the row, column, and bank addresses. Therefore, DDR3 is, internally, a three-dimensional (row, column, bank) storage array. Current DDR3 memory chips are essentially designed with eight banks, each of which contains thousands of rows and columns. To determine the final memory cell, the DDR3 addressing process appoints the bank address first, then the row address, and finally the column address. DDR3 read and write accesses are accomplished through a burst transaction of consecutive storage cells in the same row. The burst length is the number of cycles of continuous transmission, and in the case of burst transaction, as long as the starting column address and the burst length are specified, the memory will automatically read or write the corresponding number of memory cells in sequence, without the need for the controller to continuously provide column addresses. The DDR3 burst length can be configured to 4 or 8 on demand. In most cases, to enable higher DDR3 access bandwidth, the burst length is set to 8, indicating that the memory cell data are read and written at 8 consecutive column addresses on the rising and falling edges of 4 clock cycles.

In DDR3 read and write operations, the “row activation” command must first be executed, followed by the logical Bank address and the corresponding row address to activate the need to read and write the row. DDR3 must send the “activation” command to the row in order to read and write the data in a row. Following that, the “read or write command activation” is executed, which contains both column addresses and the specific operation command (read or write). After reading the data, the memory chip will perform a pre-charging operation to turn off the active working row in order to address and transfer data to other rows in the same bank. Therefore, row activation, the read or write command activation, the data reading or writing process, and pre-charging constitute the order of operations for DDR3 memory read or write operations.

Row activation time, read or write command activation time, data reading or writing process time, pre-charging time, and the interval delay between them are all deterministic values for DDR3 SDARM chips, while the factors influencing the DDR3 read/write access time are primarily the number of times that continuous burst access is performed. The burst length is typically eight, and the number of consecutive bursts is assumed to be \( n \). The interval of time between the current row activation command and the next row activation command during a write operation is:

\[
t_{\text{WRITE}} = t_{\text{RCD}} + t_{\text{CWL}} + n \times 4t_{\text{CK}} + t_{\text{WR}} + t_{\text{RP}}
\]  \hspace{1cm} (1)

The interval of time between the current row activation command and the next row activation command during a read operation is:

\[
t_{\text{READ}} = t_{\text{RCD}} + t_{\text{RTP}} + n \times 4t_{\text{CK}} + t_{\text{RP}}
\]  \hspace{1cm} (2)

where \( t_{\text{CK}} \) is the DDR3 chip’s working clock period; \( t_{\text{RCD}} \) is the interval delay time between when the row activation command is valid and when the read or write command activation is valid; the delay from the effective write command to the effective write data is called \( t_{\text{CWL}} \); the time from the completion of the write operation to the pre-charging is called \( t_{\text{WR}} \); \( t_{\text{RP}} \) represents the delay time between pre-charging and the activation of the next row; \( t_{\text{RTP}} \) represents the time interval between the read data command and the pre-charging [35].

The raw echo data from SAR are stored and processed as a matrix. The direction of radar antenna irradiance is frequently known as the range, and the direction of radar satellite flight is referred to as the azimuth. Each range line in the data matrix corresponds to a row, and each azimuth line corresponds to a column. Therefore, the range direction and the azimuth direction can be regarded as two directions of a two-dimensional matrix, respectively. In the following section, we will investigate the cause of the two-dimensional data’s unbalanced access efficiency by calculating the access efficiency of range and azimuth of the traditional SAR data that are stored sequentially.

The data on one range or azimuth line must be stored in multiple rows of DDR3 storage cells due to the massive amount of data in the SAR two-dimensional matrix. The
burst length in a typical DDR3 storage structure is set to 8 and contains 1024 cells per row. For traditional two-dimensional data access, the data in the range direction only need to be accessed in the order of DDR3 rows, but the data in the azimuth direction can only be accessed through multiple row-crossing. Therefore, the burst transmission for the range access is \( n = \frac{1024}{8} = 128 \). The read and write efficiency of access in the range is calculated using the MT8KTF51264HZ-1G9 memory chip datasheet by converting its above parameters \((t_{RCD}, t_{CWL}, t_{WR}, t_{RP}, \text{ and } t_{RTP})\) into multiples of \( t_{CK} \) as follows:

\[
\eta_{\text{WRITE\_range}} = \frac{128 \times 4t_{CK}}{t_{RCD} + t_{CWL} + 128 \times 4t_{CK} + t_{WR} + t_{RP}} = 92.25\% \quad (3)
\]

\[
\eta_{\text{READ\_range}} = \frac{128 \times 4t_{CK}}{t_{RCD} + t_{RTP} + 128 \times 4t_{CK} + t_{RP}} = 94.64\% \quad (4)
\]

For the azimuth access, only one burst transmission per range line is required for azimuth data access on the next range line; therefore, \( n = 1 \). The following formula calculates the read and write efficiency of access in the azimuth.

\[
\eta_{\text{WRITE\_azimuth}} = \frac{1 \times 4t_{CK}}{t_{RCD} + t_{CWL} + 1 \times 4t_{CK} + t_{WR} + t_{RP}} = 8.51\% \quad (5)
\]

\[
\eta_{\text{READ\_azimuth}} = \frac{1 \times 4t_{CK}}{t_{RCD} + t_{RTP} + 1 \times 4t_{CK} + t_{RP}} = 12.12\% \quad (6)
\]

As shown in Figure 2, the CS imaging algorithm must execute various range and azimuth transposition operations. The traditional SAR data sequential storage method was calculated as shown in the above, and the range access efficiency showed a small efficiency loss compared to the ideal efficiency. However, the efficiency of azimuth access showed a large loss due to the need for frequent row-crossing access. The maximum bandwidth of azimuth access is only about 12% of the peak bandwidth of DDR3 read and write. The theoretical peak read or write bandwidth is:

\[
B = 800 \text{ M} \times 64 \text{ bit} \times 2 = 12.5 \text{ GB/s} \quad (7)
\]

After considering the loss of efficiency caused by row activation, pre-charging, etc., the actual DDR3 read and write bandwidth is:

\[
B\eta = B \times \eta \quad (8)
\]
When the calculated range and azimuth read and write efficiency are substituted into the above formula, the two read and write bandwidths listed in Table 1 can be obtained.

<table>
<thead>
<tr>
<th>Performance</th>
<th>Range</th>
<th>Azimuth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Writing efficiency</td>
<td>92.25%</td>
<td>8.51%</td>
</tr>
<tr>
<td>Reading efficiency</td>
<td>94.64%</td>
<td>12.12%</td>
</tr>
<tr>
<td>Writing bandwidth</td>
<td>11.53 GB/s</td>
<td>1.06 GB/s</td>
</tr>
<tr>
<td>Reading bandwidth</td>
<td>11.83 GB/s</td>
<td>1.52 GB/s</td>
</tr>
</tbody>
</table>

2.3. Analysis of Existing Data Exchange Solutions

The newer data exchange methods in spaceborne SAR imaging systems are currently studied as matrix block three-dimensional cross-mapping. This method allows for an efficient balance between the range and azimuth data access and is pipelined with two off-chip DDR3s to improve efficiency. This method has a data write bandwidth of 8.45 GB/s and a data read bandwidth of 10.24 GB/s. Therefore, the data access bandwidth of the method is still limited. A brief analysis of the key technologies in this method is presented below.

2.3.1. Matrix Block Three-Dimensional Mapping

The matrix block three-dimensional mapping method fully utilizes the DDR3 chip’s cross-bank priority data access feature. The two-dimensional data matrix is divided into many equal-sized sub-matrices, then sub-matrices are consecutively mapped to different DDR3 banks. The so-called “three-dimensional mapping” means that the sub-matrices with a consecutive range and azimuth direction will be mapped to different rows of different banks in DDR3, instead of mapping to consecutive rows of the same bank as in the conventional sequential mapping method. This means that the data points in the two-dimensional matrix will be mapped to different banks, different rows and different columns, i.e., to the three-dimensional space.

2.3.2. Sub-Matrix Cross-Mapping

Submatrix cross-mapping is a more efficient mapping strategy than two-dimensional matrix sequential mapping. The mapping process is shown in Figure 3. The “cross-mapping” alternately maps two adjacent rows of data to a row with a three-dimensional storage array of DDR3, instead of using sequential mapping method, i.e., mapping one row of original data to one row of DDR3 sequentially, and then mapping the second row sequentially. The sub-matrix cross-mapping has a burst length of 8. The data transmitted in one burst belong to two rows of data in range and four columns of data in azimuth, while the data transmitted in one burst of linear mapping belong to one row of data in range and eight columns of data in azimuth. Compared with linear mapping, cross-mapping has a better balance performance when accessing a data matrix in a two-dimensional condition.

Figure 3. Schematic diagram of sub-matrix cross-mapping scheme.
2.3.3. Scalar Pipeline

Pipeline technology is one of the most commonly used techniques for instruction processing in general-purpose processors, which can significantly increase the CPU’s main frequency. Currently, pipelining technology is also increasingly applied to specific processors, such as SAR real-time processing. The use of pipelining technology can fully utilize the built hardware units, reduce idle, and improve the processing efficiency of the system.

In the paper [35], the system pipeline is a scalar pipeline, that is, when data are accessed, the data are read from DDR3A and transmitted to the processing engine for processing. When the processed data are written into DDR3B, DDR3A reads the next data in the range direction or azimuth direction for processing. This operation is repeated many times, until all the data are processed. The time–space diagram of the scalar pipeline process during data access is shown in Figure 4. The above data-access process is transferred from DDR3A to DDR3B while DDR3B is read, and after processing, writing to DDR3A is similar. The scalar pipeline is more efficient than the no-pipeline design, but the long data processing time in the SAR imaging system causes the pipeline to generate a waiting or blocking state, which has an effect on real-time system processing.

![Figure 4](image)

**Figure 4.** Schematic diagram of comparison between non-pipelined design and scalar pipeline. (a) The process of non-pipelined design; (b) The process of scalar pipeline design.

The range and azimuth read and write efficiency and bandwidth in the paper [35] are summarized in Table 2.

<table>
<thead>
<tr>
<th>Performance</th>
<th>Range</th>
<th>Azimuth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Writing efficiency</td>
<td>66%</td>
<td>66%</td>
</tr>
<tr>
<td>Reading efficiency</td>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>Writing bandwidth</td>
<td>8.45 GB/s</td>
<td>8.45 GB/s</td>
</tr>
<tr>
<td>Reading bandwidth</td>
<td>10.24 GB/s</td>
<td>10.24 GB/s</td>
</tr>
</tbody>
</table>

3. The Data Storage and Exchange Method Based on Superscalar Pipeline

The data storage method of linear mapping in the traditional spaceborne SAR imaging system makes the azimuth data access efficiency extremely low. When accessing the data matrix under two-dimensional conditions, the balance between the range direction and the azimuth direction is very low. The method of partitioning a matrix and mapping it to a three-dimensional storage array establishes a great balance during the two-dimensional
access of the matrix, which significantly improves efficiency in azimuth. Furthermore, the scalar pipeline design effectively improves the system’s processing time. However, the method’s bandwidth in both directions still creates limitations for high real-time SAR imaging systems, and the scalar pipeline design continues to have more idle time, wasting pipeline resources. As a result, we propose a superscalar pipeline-based method and architecture for efficient data storage and exchange in SAR imaging systems in order to improve real-time processing. A data storage and exchange architecture suitable for an SAR imaging system is designed based on the efficient storage mapping and superscalar pipeline data exchange method. The coordinate address of the two-dimensional data matrix are mapped to physical addresses in two DDR3s in equal amounts, a new mapping method is adopted to replace the traditional linear mapping method, and the spatial parallelism is increased by the design of a superscalar pipeline. Since the method and architecture have improved data access bandwidths in both the azimuth and range directions, the real-time processing of spaceborne SAR imaging is now more feasible.

3.1. The Three-Dimensional Cross-Mapping Method Based on Equal Division of Sub-Matrix with Dual-Channel

The three-dimensional cross-mapping scheme based on the equal division of a sub-matrix with a dual-channel takes full advantage of the parallel advantages of FPGA. Unlike the method used to block a matrix and map to a three-dimensional memory array, the data of each sub-matrix are equally divided according to their sub-matrix size and mapped to a row in dual-channel DDR3A and DDR3B, respectively. When DDR3A and DDR3B are read or written synchronously, the read operation reads data from DDR3A and DDR3B in the azimuth or range direction, enters the compute core for processing, and writes data to DDR3A and DDR3B after processing, and performs circular read and write access according to the operation flow. Figure 5 shows the method’s complete schematic diagram.

![Figure 5. Schematic diagram of data storage and exchange method based on superscalar pipeline.](image)

3.1.1. The Matrix Block Three-Dimensional Mapping with Dual-Channel

The amount of both the raw SAR imaging data and the intermediate processing result data is \( N_A \times N_R \times 64 \) bit, i.e., the number of range data points is \( N_R \), the number of azimuth data points is \( N_A \), and the data width of each point is 64 bit. Since the SAR imaging processing requires multiple FFTs, both and are positive integer power of 2.

Perform matrix block of the raw data as shown Figure 5. The matrix is divided into \( M \times N \) sub-matrices. The size of the sub-matrix is \( N_a \times N_r \), i.e., \( N_a = N_A / M \), \( N_r = N_R / N \). The sub-matrices need to be mapped to one row in both DDR3A and DDR3B respectively, so that \( N_a \times N_r = 2C_n \), where \( C_n \) is the number of columns of the storage array in DDR3.

The SAR data volume of \( 16 \times 16 \times 64 \) bit is the more common data volume in the current SAR real-time processing system, so this paper targets the \( 16 \times 16 \times 64 \) bit data volumes for the study. The sub-matrix size is \( N_a \times N_r = 64 \times 32 = 2048 \). The sub-matrix has 2048 points, which is exactly the sum of the lengths of one row in DDR3A and one row...
in DDR3B respectively, and can be mapped to DDR3A and DDR3B with an equal division in three dimensions.

After the matrix is blocked, the matrix are mapped in a dual-channel three-dimensional manner, and the data in each blocked sub-matrix are mapped to a row in DDR3A and DDR3B in equal division. Each row of the SAR data matrix is assumed to be the range and each column is the azimuth. The data in the range direction are called the range line and the data in the azimuth direction are called the azimuth line, so there are \( M \) range lines and \( N \) azimuth lines. Using the three-dimensional mapping method according to Figure 6, firstly, 512 sub-matrices on the 0th range line are mapped to the 0th row of bank0 to bank7 in DDR3A and DDR3B, respectively, in a cross-bank mode, and since the number of sub-matrices on the 0th range line is larger than the number of banks, after mapping to bank7, then return to bank0 row 1. When starting the mapping; the next is bank1 row 1, bank2 row 1 . . . bank7 row 1. At present, all the rows 0 to 1 of bank0 to bank7 are mapped, and then the mapping continues from bank0 row 2, bank1 row 2, bank2 row 2 . . . bank7 row 2, etc., until the 512 sub-matrices on the range line 0 are mapped.

![Figure 6. Schematic diagram of the matrix block three-dimensional mapping scheme based on parallel dual channel.](image)

By calculation, the 512 sub-matrices on the 0th range line are mapped, which requires 64 rows in each bank. Therefore, as shown in Figure 6, the 512 sub-matrices on the 0th range line will be mapped from row \( R = 0 \) to row \( R = 63 \) in banks0-7 of DDR3A and DDR3B.

To ensure that the azimuth access also satisfies the cross-bank principle, the sub-matrices on the 1st range line will be mapped from bank1. bank1 row 64, bank2 row 64 . . . bank7 row 64, bank0 row 64, i.e., mapped to bank1-bank7-bank0, respectively. The mapping starts from line 64 within bank. This is followed by bank1 row 65 . . . bank7 row 65, bank0 row 65, and the subsequent range lines are also mapped according to the above rules. Until then, all data can be mapped to DDR3A and DDR3B at the same time.

3.1.2. Sub-Matrix Equal Division Cross-Mapping

After the three-dimensional matrix block mapping method mentioned above is complete, the sub-matrix cross-mapping method is used in this section to equally divide the sub-matrices so that they can be mapped to both DDR3A and DDR3B.

Two pieces of DDR3 can be regarded as a whole with twice the bit width, capacity, and bandwidth of a single piece of DDR3. When the system issues an access command to the matrix transposition module, the two pieces of DDR3 execute the same read or write
command together. Since the burst length of a single DDR3 is 8, the burst length could be 16 when two DDR3s are read and written at the same time; that is, 16 pieces of data are continuously read or written in 4 clock cycles. The 16 pieces of data can be mapped to a set of data arranged according to a square structure in the sub-matrix, and these 16 data belong to four consecutive azimuth directions and four consecutive range directions, respectively. They are divided into two groups by an equivalent amount, which are stored in DDR3A and DDR3B, respectively. In SAR imaging processing, the FFT operation is processed with complete data in the range or azimuth line, and the range and azimuth directions of the data transmitted in a single burst are balanced, making the system computing engine utilization 100%. Figure 7 is a schematic diagram of an equal division cross-mapping of a sub-matrix with a dual channel.

| Column | DDR3A Row | DDR3B Row | 0.0 | 0.1 | 0.2 | 0.3 | 1.0 | 1.1 | 1.2 | 1.3 | ... | 0.Nr-1 | 1.0 | 1.1 | 1.2 | 1.3 | 2.0 | 2.1 | 2.2 | 2.3 | 3.0 | 3.1 | 3.2 | 3.3 | ... | 2.Nr-1 | ... | Na-4,0 | ... | ... | Na-4.Nr-1 | Na-3,0 | ... | ... | Na-3.Nr-1 | Na-2,0 | ... | ... | Na-2.Nr-1 | Na-1,0 | ... | ... | Na-1.Nr-1 | Na-0,0 | ... | ... | Na-0.Nr-1 | 0,0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | ... | 0,Na-1 | 1,0 | 1,1 | 1,2 | 1,3 | 2,0 | 2,1 | 2,2 | 2,3 | 3,0 | 3,1 | 3,2 | 3,3 | ... | 3,Na-1 | ... | ... | ... | 0,1023 |

Figure 7. Schematic diagram of sub-matrix equal division cross-mapping with dual channel.

In this mapping method, two consecutive rows of data in the sub-matrix are mapped to the same DDR3, e.g., rows 0 and 1 in the sub-matrix are mapped to DDR3A, rows 2 and 3 are mapped to DDR3B, etc., in that order. The 16 data in the box are the data of the first burst transmission. After the 16 data are transmitted in burst mode, 16 more data are transmitted in range direction if it is a range access, and if it is an azimuth access, 16 data will be transmitted in the azimuth, and so on until the data access is completed.

3.1.3. Address Mapping Strategy

According to the mapping method proposed above, this section will describe the address mapping strategy that is used. Address mapping refers to the process of obtaining the physical storage address \((i, j, k)\) in DDR3 SDRAM for any given coordinate address \((x, y)\) within the SAR data matrix. The data points whose coordinates are \((x, y)\) in the two-dimensional data matrix can be explicitly distributed to the sub-matrix \(A_{mn}\). Assume that the coordinates in this submatrix are \((a, b)\). The relationship between \((a, b)\) and \((x, y)\) is as follows.

\[
\begin{align*}
    a &= x - m \times Na \\
    b &= y - n \times Nr
\end{align*}
\]

(9)

Among them, \(0 \leq m \leq M - 1, 0 \leq n \leq N - 1, 0 \leq a \leq Na - 1, 0 \leq b \leq Nr - 1,\) satisfying \(M \times Na = N_A, N \times Nr = N_R\). According to the three-dimensional cross-mapping method based on an equal division of the sub-matrix with dual-channel DDR3, the mapping
relationship between the two-dimensional matrix coordinate address \((x, y)\) and the DDR3 physical address \((i, j, k, z)\) is shown in the following formula.

\[
\begin{align*}
  i &= \text{floor}((m \times n + n) / B_n) \\
  j &= \text{floor}(\text{floor}(a/2)/2) \times 2N_r + 2b + \text{mod}(a, 2) \\
  k &= \text{mod}(m + n, B_n)
\end{align*}
\] (10)

\(\text{floor}(p)\) means to return the largest integer less than or equal to \(p\); \(\text{mod}(p, q)\) denotes the remainder of \(p/q\). \(i, j, k\) are the rows, columns, and banks in the DDR3.

Through the above formula, the storage physical address corresponding to all data coordinate addresses of the two-dimensional matrix can be calculated, but the calculation is large and complicated. We found that, for the SAR two-dimensional data matrix, the jump of the coordinate address \((x, y)\) has a continuous and simple law when performing continuous data access operations in the range or azimuth direction. There is also a certain relationship between the physical addresses of the DDR3 SDRAM of the two adjacent data accesses. Therefore, the coordinate address of the currently accessed data can be obtained by adding a regular coordinate offset to the coordinate address of the previous data. Due to the burst transmission characteristics of DDR3 SDRAM, we found that many data points in the two-dimensional matrix do not need to calculate their corresponding physical storage addresses. Eight pieces of data in a burst transmission are accessed using the same unique address. Taking the divided sub-matrix in Figure 8 as an example, the eight adjacent data points with the same color are the data of the same burst transmission. However, the physical storage addresses of these eight data points do not all need to be calculated. Only the physical storage addresses of the data points marked by the circle or triangle in the following figure are needed to store the eight data points into DDR3 at one time. Since the sub-matrix must be mapped equally to two DDR3 SDRAM, the circular representation uses the DDR3A physical mapping address of the data point as a burst transmission address, and the triangular representation uses the DDR3B physical mapping address of the data point as a burst transmission address.

Figure 8. Schematic diagram of sub-matrix address mapping.

Because the addresses mapped to DDR3A and DDR3B are completely independent, the above-mentioned sub-matrix of \(N_a = 64\) and \(N_r = 32\) can be regarded as a combination of two sub-matrices corresponding to DDR3A and DDR3B respectively. The size of two sub-matrices is \(N_a/2 = 32\) and \(N_r = 32\).

Next, the address mapping conversion will be performed on the \(32 \times 32\) sub-matrix mapped to DDR3A. Assuming that the coordinates of the data points in the sub-matrix
are \( P(a, b) \), accessing the data of point \( P \) is actually accessing the address of point \( p'(a', b') \). The calculation formula of point \( p'(a', b') \) is as follows.

\[
\begin{align*}
  a' &= \text{floor}(a/2) \times 2 = \text{floor}(\text{mod}(x, Na/2)/2) \times 2 \\
  b' &= \text{floor}(b/4) \times 4 = \text{floor}(\text{mod}(y, Nr)/4) \times 4
\end{align*}
\] (11)

By substituting the above formula into the following formula, the actual physical storage address of DDR3 SDRAM corresponding to point \( p'(a', b') \) can be obtained.

\[
\begin{align*}
  i &= \text{floor}((m \times N + n)/B_n) \\
  j &= \text{floor}(a'/2) \times 2Nr + 2b' + \text{mod}(a', 2) = a' \times Nr + 2b' \\
  k &= \text{mod}(m + n, B_n)
\end{align*}
\] (12)

According to the above discussion, the current data position coordinates are obtained by adding a certain offset to the coordinates of the previous data. For the range direction, in the continuous access process starting from the data point \( A(x, y) \), it is necessary to sequentially obtain the physical address of \( A(x, y) \), \( A(x, y + 4) \), \( A(x, y + 8) \), etc. To make the address calculation easier, we use the DDR3 physical address \( D \) corresponding to \( A \) to calculate the DDR3 physical address \( D_1 \) corresponding to \( A_1 \). Therefore, during continuous access in the range direction, only the coordinates of the initial data point in the two-dimensional matrix are calculated to the actual physical address of DDR3, and the address of the data point for subsequent access is quickly calculated based on the physical address of the previous data point. The continuous access method in the azimuth direction is similar to that in the range direction.

3.2. Superscalar Pipeline Ping-Pong Buffer

In the SAR imaging system, the read data need to be buffered into the on-chip RAM first, and then sent to the calculation engine such as FFT for processing after an entire row or column of data has been accessed. Therefore, a data buffer is a relatively important data exchange method in SAR imaging systems. Superscalar pipelining can dramatically improve the efficiency of data exchange compared to traditional non-pipelined buffering and most current scalar pipeline processing.

3.2.1. Data Exchange in Range Direction Based on Superscalar Pipeline

When the range access is performed, as shown in Figure 9, the eight data points in green will be accessed from DDR3A, which belongs to two consecutive range directions, and is stored in the buffer RAM0 and RAM1 corresponding to DD3A. The eight data points in yellow will be accessed from DDR3B, which also belongs to two consecutive range directions, and is stored in the buffer RAM0 and RAM1 corresponding to DD3B. After the 16 data points in the four range directions are accessed, other data in the four range directions will be accessed by the operation in the range direction. After the data access of the four range directions in the sub-matrix is completed, the adjacent range direction sub-matrix is accessed by range direction. Repeat the operation until all the data of these four range directions are accessed.

The following four range directions’ data will be accessed when DDR3A and DDR3B keep working simultaneously after the first four range directions’ data are accessed and stored in the Ping RAM buffers corresponding to DDR3A and DDR3B. As shown in Figure 9, the eight data points in blue will be accessed from DDR3A, which belongs to two consecutive range directions, while the eight data points in orange will be accessed from DDR3B, which also belongs to two consecutive range directions. The four types of range data will be stored in the Pong RAMs corresponding to DDR3A and DDR3B, and form a pipeline design mode with the Ping RAMs corresponding to DDR3A and DDR3B. At this time, while the Pong RAM corresponding to DDR3A and DDR3B is buffering the four
3.2.2. Data Exchange in Azimuth Direction Based on Superscalar Pipeline

When performing azimuth access, as shown in Figure 10, one column of data in the figure belongs to one azimuth line; therefore, two data points of the same color in a row belong to two different azimuth lines. The data points of four azimuth lines are designed to be processed at the same time. Therefore, the eight data points in green will be accessed from DDR3A, belonging to four consecutive azimuth lines, and stored in the buffer RAM0–RAM3 corresponding to DD3A. The eight data points in yellow will be accessed from DDR3B, which also belongs to four consecutive azimuth directions, and is stored in the buffer RAM0–RAM3 corresponding to DD3B. After the 16 data points in the four azimuth directions in total are accessed, the other data in the four azimuth directions will be accessed because of the operation in the azimuth direction. After the data in the four azimuth directions in the sub-matrix are all accessed, the adjacent azimuth direction sub-matrix is accessed by azimuth direction. Repeat the operation until all the data points in these four azimuth directions are accessed.

The following four azimuth direction data points will be accessed when DDR3A and DDR3B continue work simultaneously after the first four types azimuth direction data are accessed and stored in the Ping RAM buffers corresponding to DDR3A and DDR3B. As shown in Figure 10, the eight data points in blue will be accessed from DDR3A, which belongs to four consecutive azimuth directions, while the eight data points in orange will be accessed from DDR3B, which also belongs to four consecutive azimuth directions. The four types of azimuth data will be stored in the Pong RAMs corresponding to DDR3A and DDR3B, and form a pipeline design mode with the Ping RAMs corresponding to DDR3A and DDR3B. At this time, while the Pong RAM corresponding to DDR3A and DDR3B is buffering the four types of azimuth data, the Ping RAM corresponding to DDR3A and DDR3B transfers the four types of azimuth data that have been buffered to the compute core for processing.

In the traditional three-dimensional cross-mapping method, the eight data points transmitted in a single burst of DDR3 belong to four azimuth data and two range data, which will cause half of the computing engine to be idle during processing in the range direction. However, the eight data points transmitted in a single burst by the method proposed in this paper belong to four azimuth data and four range data, which will make the data access in the range direction and azimuth direction completely balanced. The computing resource utilization in the range direction can reach 100%.
Figure 10. Schematic diagram of data exchange in azimuth direction based on superscalar pipeline.

The superscalar architecture pipeline design is shown Figure 11; each DDR3 corresponds to a group of ping-pong RAM buffers, which can form a pipeline independently. Dual-channel DDR3 read and write operations can be executed concurrently through the design of the superscalar pipeline, meaning that read or write operations are performed on both DDR3A and DDR3B simultaneously, increasing the parallelism of data access. This method can result in greatly reduced idle latency in conventional pipelined or non-pipelined designs.

Figure 11. Schematic diagram of pipeline processing based on superscalar structure.

4. Hardware Implementation

Based on the method proposed in Section 3, this paper designs a hardware architecture for an efficient data exchange engine on-chip in an SAR imaging processor. The architecture adopts the control mode of modular register addressing. Through the combination of superscalar pipeline buffer module and dual-channel DDR3 access control module, the frequent exchange of massive data in the real-time processing of SAR imaging is optimized and the real-time processing efficiency is improved.

4.1. On-Chip Data Exchange Engine Architecture

Figure 12 below shows the general layout of the SAR imaging processor’s high-efficiency data exchange engine. A DDR3 read and write access control module based on the equal division of two-channel sub-matrix and three-dimensional cross-mapping, regarded as the DDR3 TOP module, and a superscalar pipeline data buffer module are included in the top layer of the whole data exchange engine design.
4.1.1. DDR3_TOP Module

Dual-channel read and write access control of two off-chip DDR3A and DDR3B is achieved by the DDR3_TOP module. All the data that interact with DDR3 must go through this module. The sub-modules are: DDR3 Control Unit Register (DCU Reg) is used to control all operation process interacting with DDR3 and monitor the feedback status of the read and write process. MIG is the official memory interface control IP core of Xilinx, which is used to directly read and write data with off-chip DDR3. The primary modules for mapping address generation, read and write access control, and input and output data buffer management are DDR3A_CTRL and DDR3B_CTRL.

4.1.2. Dline_Buf Module

The superscalar pipeline data exchange control with the DDR3 storage side and the compute core processing side during processing is controlled by the Dline_buf module. The DDR3 output to be processed, the data buffering procedure to be stored after processing, and the feedback status to be monitored during the data exchange process are all controlled by the Data Exchange Unit Register (DEU Reg). The RAMsA_Ctrl and RAMsB_Ctrl modules mainly control the Ping-Pong RAM pipeline and data buffer corresponding to DDR3A and DDR3B. The data exchange paths corresponding to DDR3A and DDR3B are primarily under the command of the Data ManageA Ctrl and Data ManageB Ctrl modules.

4.1.3. Structure of Address Generation

The structure of the address generation module is shown in the figure below (Figure 13). This mainly includes the address initialization part and address update part. The address initialization part is mainly used to initialize the initial coordinates of the two-dimensional...
matrix, and convert them into the DDR3 physical address. Subsequently, the DDR3 physical address will be calculated in the address update part.

![Figure 13](image_url)

**Figure 13.** Schematic diagram of address mapping hardware architecture.

The address initialization part mainly includes the coordinate calculation in the sub-matrix corresponding to the starting point, the calculation of the Bank, Row and Column of the DDR3 physical address, and the calculation of the intermediate parameters required for the update of the range and azimuth addresses. In addition, any coordinate in the two-dimensional data matrix can be used as the starting coordinate of the access.

Since the addresses for continuous data access in the range and azimuth directions are different, the address update part includes an independent range and azimuth address generation module, which can update the DDR3 physical address in the continuous access process according to the access mode.

### 4.2. Register Control for Custom Engines

Our entire system architecture is designed and implemented using register control signals to configure parameters, status feedback, and process control for all engines. All custom engines share a set of register access requests, address lines, and data lines. The interface for the custom engine register access control signals is provided in Table 3. After setting the read or write request and register address of the custom engine that needs to be accessed, as well as the register write data during write access, all custom engines can receive the register access information. Then, within each custom engine, the accurate register that is being accessed is determined through absolute address, and the register access response signal is generated accordingly. The VC709 FPGA register control path and data path are shown in Figure 14. Data download and data upload control receiving raw data and sending processed data, respectively, and the data bit width is 256 bit. The control signal request broadcast shares register control signals to the custom engine in the form of broadcast transmission. The control signal response selection is used to select the output signal of each custom engine register interface, and to accurately select the read or write response signal of the accessed register and the read data during read access.
The register settings of the DDR3_TOP module and the Dline_Buf module in the superscalar pipeline data storage exchange engine are shown in Tables 4 and 5, respectively. Through the control of these registers, the parameterized configuration and imaging process control of the superscalar pipeline data storage and exchange engine can be realized. The main control terminal can also monitor the working status of each customized engine for subsequent operations. In addition, to improve the scalability of the data storage exchange engine, registers can be added or modified according to the design plan.
Table 4. DDR3_TOP module register information table.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Register Address</th>
<th>Access Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR3_op_conf_register</td>
<td>14'h40</td>
<td>W</td>
<td>Operation Flow Control Register</td>
</tr>
<tr>
<td>DDR3A_state_register</td>
<td>14'h44</td>
<td>R</td>
<td>DDR3A Status Register</td>
</tr>
<tr>
<td>DDR3B_state_register</td>
<td>14'h48</td>
<td>R</td>
<td>DDR3B Status Register</td>
</tr>
<tr>
<td>DDR3_main_state_register</td>
<td>14'h4C</td>
<td>R</td>
<td>DDR3 Operation Status Flag Register</td>
</tr>
<tr>
<td>DDR3_start_x_register</td>
<td>14'h50</td>
<td>W</td>
<td>x Coordinate Start Configuration Register</td>
</tr>
<tr>
<td>DDR3_start_y_register</td>
<td>14'h54</td>
<td>W</td>
<td>y Coordinate Start Configuration Register</td>
</tr>
<tr>
<td>Granularity_Conf_Register</td>
<td>14'h58</td>
<td>W</td>
<td>Data Granularity Configuration Register</td>
</tr>
</tbody>
</table>

Table 5. Dline_Buf module register information table.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Register Address</th>
<th>Access Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dline_Buf_op_conf_register</td>
<td>14'h80</td>
<td>W</td>
<td>Operation Flow Control Register</td>
</tr>
<tr>
<td>Dline_BufA_state_register</td>
<td>14'h84</td>
<td>R</td>
<td>Dline BufA Status Register</td>
</tr>
<tr>
<td>Dline_BufB_state_register</td>
<td>14'h88</td>
<td>R</td>
<td>Dline BufB Status Register</td>
</tr>
<tr>
<td>Dline_Buf_main_state_register</td>
<td>14'h8C</td>
<td>R</td>
<td>Dline Buf Operation Status Flag Register</td>
</tr>
<tr>
<td>Buffer_points_Conf_Register</td>
<td>14'h90</td>
<td>W</td>
<td>Data Buffer Points Configuration Register</td>
</tr>
</tbody>
</table>

5. Experiments and Results

Extensive experiments were conducted to evaluate the performance of the proposed superscalar pipelined data storage exchange method and hardware architecture. To ensure the completeness and comprehensiveness of the experimentation and evaluation, the superscalar pipeline data storage and exchange engine was embedded as an intellectual property (IP) into the “CPU + FPGA” heterogeneous SAR imaging processing system platform for testing. The experimental result evaluation is divided into two parts. The first part involves a performance evaluation of the superscalar pipeline data storage and exchange engine, from which the bandwidth of storage access in both the range and azimuth directions, as well as the matrix conversion time, are obtained. Furthermore, a comparison of the performance metrics is conducted with reference to multiple relevant papers. In addition, the throughput rate, efficiency, and speedup ratio of the data storage exchange engine with a super-scalar pipeline are presented and compared with non-pipeline and scalar pipeline approaches in order to analyze the performance enhancement achieved through the super-scalar pipeline design. The second part entails the performance analysis of the imaging results obtained from the “CPU + FPGA” heterogeneous SAR imaging processing system platform. In this part, according to the complete imaging process of SAR, the SAR raw echo data are processed using the “CPU + FPGA” heterogeneous SAR imaging processing system platform to obtain SAR imaging results, which are then compared to the imaging results obtained from MATLAB. In addition, a description of the hardware resource utilization of the entire heterogeneous system is provided. Finally, the imaging performance of SAR is evaluated and compared with other SAR imaging processing systems. In the following subsections, the experimental environment configuration and detailed experimental results will be described.

5.1. Experiment Setting

For the integrity of the experiment, the “CPU + FPGA” heterogeneous SAR imaging processing MPSOC system platform consisting of a host computer and two hardware development boards, namely Zynq UltraScale+ MPSoC ZCU106 and Virtex-7 VC709, is utilized. The host computer is responsible for the downlink transmission of SAR raw echo data. AZCU106 includes Zynq UltraScale+ XCZU7EV-2FFVC1156 MPSoC, equipped with quad-core ARM Cortex-A53 application processor, dual-core Cortex-R5 real-time processor and a large number of programmable logic resources [40]. The development board is
shown in Figure 15, equipped with two SFP Cages for data streaming from heterogeneous platforms and four PL-side Micron MT40A256M16GE-075E DDR4 SDRAM memories with a total capacity of up to 2 GB.

![Physical picture of Xilinx Zynq UltraScale+ MPSoC ZCU106 development board.](image)

Figure 15. Physical picture of Xilinx Zynq UltraScale+ MPSoC ZCU106 development board.

We deployed the superscalar pipeline data storage and exchange engine hardware architecture on the Xilinx Virtex-7 FPGA VC709 development board, which is equipped with Xilinx XC7VX690T FPGA chip and two Micron MT8KTF51264HZ-1G9 DDR3 SDRAM memory modules [41], as shown in the Figure 16. Each DDR3 SDRAM has a storage capacity of 4 GB with storage unit data width of 64 bits. The SAR two-dimensional matrix data granularity is $16,384 \times 16,384$, with a data width of 64 bits, composed of the concatenation of the real and imaginary parts in single-precision floating-point format.

![Physical picture of Xilinx Virtex-7 FPGA VC709 development board.](image)

Figure 16. Physical picture of Xilinx Virtex-7 FPGA VC709 development board.

The experimental software platform used was Vivado 2020.2 and Vitis HLS 2020.2 version. Vivado 2020.2 is mainly used for the hardware design of ZCU106 and VC709 development boards to build an SoC hardware processing platform. Vitis HLS 2020.2 is
mainly used for software design in addition to the hardware platform, and Vitis can be used to control the development application of ARM application processors in ZCU106. This experimental project is developed and deployed using integrated circuit hardware description language (VHDL). The Zynq UltraScale+ MPSoC core version v3.3 is employed in the ZCU106 development board, along with AXI Chip2Chip Bridge IP core version v5.0 and Aurora 64B66B IP core version v12.0. Additionally, the MIG IP core version v4.2 and Block RAM IP core version v8.4 are employed in the VC709 development board.

According to the standard CS imaging algorithm, the experimental procedure is designed as follows. Step 1: raw data entry. The SAR raw data are transmitted from the upper machine through the Gigabit Ethernet port on the ZCU106 side to the DDR4 on the Zynq PS side for storage. After the superscalar pipeline storage exchange engine and the calculation engine on the VC709 side are initialized, they are transferred from the ZCU106 development board to the VC709 development board through SFP Cages. After the VC709 receives the data, they are converted into the AXIS protocol data stream, and the two-dimensional matrix is written into DDR3 through the storage exchange engine in the form of range. Step 2: Azimuth processing. The data are read from DDR3 in azimuth, buffered in superscalar pipeline, and output from the storage exchange engine to the calculation engine for FFT and complex multiplication calculation, which is still written to DDR3 in azimuth after the processing is completed. Step 3: range processing. The data are read from DDR3 in the range direction, buffered in the superscalar pipeline, output from the storage exchange engine to the calculation engine for FFT and complex multiplication calculation, and then written to DDR3 in the range direction after the processing is completed. Step 4: range processing. The data are read from DDR3 according to the range direction, and output from the storage exchange engine to the calculation engine for FFT and complex multiplication calculation, and then written to DDR3 according to the range direction. Step 5: azimuth processing. The data are read from DDR3 according to the azimuth direction, passed through the superscalar pipeline buffer, and output from the storage exchange engine to the calculation engine for complex multiplication and IFFT processing. After the processing is completed, they are still written to DDR3 in the azimuth direction. Then, all operations are completed. Finally, data are read from the VC709-side DDR3, transmitted to the ZCU106 development board through the VC709-side SFP Cages, and subsequently uploaded to the host computer to display imaging results. The experimental test process is shown in Figure 17.

Figure 17. Schematic diagram of experimental testing platform and process.

5.2. Experimental Results and Performance Evaluation

5.2.1. Superscalar Pipeline Data Storage and Exchange Engine

We verified the bandwidth and access efficiency of the proposed hardware architecture through the use of an FPGA. The theoretical bandwidth of the two off-chip DDR3 SDRAMs of the processing board with 64 bit storage bit width and 800MHz operating clock frequency is:

\[
B = 800 \text{ M} \times 64 \text{ bit} \times 2 \times 2 = 25 \text{ GB/s}
\] (13)
After considering loss of efficiency caused by row activation, pre-charging, etc., the actual DDR3 read and write bandwidth is:

$$B_\eta = B \times \eta$$

(14)

where $\eta$ is the access efficiency.

When we use the above formula to calculate the access efficiency, we can obtain Table 6 indexes by recording the actual measured range and azimuth access bandwidth.

**Table 6.** The read and write bandwidth performance of the data storage exchange engine based on superscalar pipeline proposed in this paper.

<table>
<thead>
<tr>
<th></th>
<th>Range (Read)</th>
<th>Range (Write)</th>
<th>Azimuth (Read)</th>
<th>Azimuth (Write)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical Bandwidth</td>
<td>25.0 GB/s</td>
<td>25.0 GB/s</td>
<td>25.0 GB/s</td>
<td>25.0 GB/s</td>
</tr>
<tr>
<td>Measured Bandwidth</td>
<td>16.6 GB/s</td>
<td>16.0 GB/s</td>
<td>20.0 GB/s</td>
<td>18.3 GB/s</td>
</tr>
<tr>
<td>Access Efficiency</td>
<td>66%</td>
<td>64%</td>
<td>80%</td>
<td>73%</td>
</tr>
</tbody>
</table>

Recent SAR imaging processing implementations on FPGA platforms are also compared in this section. Table 7 shows how the hardware implementations of the data storage and exchange method proposed in this paper and a few other recent hardware implementation schemes compare in terms of how well they perform. The range and azimuth access bandwidths in this table show the maximum bandwidth of data storage access in this direction. “Matrix Transposition Time” refers to the time interval from the start of writing DDR3 in the range direction to the completion of reading DDR3 data in the azimuth direction, which can be calculated from the access bandwidth of range and azimuth. “Number of RAM buffer” refers to the data buffer RAM number for a single Chip DDR3.

**Table 7.** Table of the performance comparison between the hardware implementation of the data storage and exchange method proposed in this paper and the recent scheme.

<table>
<thead>
<tr>
<th></th>
<th>Ours</th>
<th>[35]</th>
<th>[34]</th>
<th>[42]</th>
<th>[33]</th>
<th>[43]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Granularity</td>
<td>16,384 × 16,384</td>
<td>16,384 × 16,384</td>
<td>16,384 × 16,384</td>
<td>16,384 × 16,384</td>
<td>16,384 × 16,384</td>
<td>8192 × 8192</td>
</tr>
<tr>
<td>FPGA</td>
<td>Xilinx XC7VX690T</td>
<td>Xilinx XC7VX690T</td>
<td>Xilinx XC7VX690T</td>
<td>Xilinx XC6VLX315T</td>
<td>Xilinx XC6VSX760T</td>
<td>Xilinx XC7VX330T</td>
</tr>
<tr>
<td>DDR3 channel number</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Range access bandwidth</td>
<td>16.6 GB/s</td>
<td>10.24 GB/s</td>
<td>8.3 GB/s</td>
<td>4.8 GB/s</td>
<td>6.0 GB/s</td>
<td>-</td>
</tr>
<tr>
<td>Azimuth access bandwidth</td>
<td>20.0 GB/s</td>
<td>10.24 GB/s</td>
<td>9.62 GB/s</td>
<td>4.8 GB/s</td>
<td>2.37 GB/s</td>
<td>-</td>
</tr>
<tr>
<td>Range access efficiency</td>
<td>66%</td>
<td>80%</td>
<td>69%</td>
<td>74%</td>
<td>93.75%</td>
<td>-</td>
</tr>
<tr>
<td>Azimuth access efficiency</td>
<td>80%</td>
<td>80%</td>
<td>80%</td>
<td>74%</td>
<td>74%</td>
<td>-</td>
</tr>
<tr>
<td>Matrix transposition time</td>
<td>0.225 s</td>
<td>0.43 s</td>
<td>0.45 s</td>
<td>0.83 s</td>
<td>1.18 s</td>
<td>0.33 s</td>
</tr>
<tr>
<td>Buffer RAM number</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>Superscalar pipeline supported or not</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

The data storage and exchange engine of the SAR imaging system in this paper is implemented based on superscalar pipeline. Compared with non-pipeline and scalar pipeline design, superscalar pipeline design has certain performance improvements in terms of pipeline throughput rate, efficiency, and speedup ratio. Subsequently, a comparison of the pipeline performance of the superscalar pipeline designed in this paper and the tradi-
tional non-pipeline and scalar pipeline will be conducted. The schematic diagram of the comparison results is shown in Figure 18.

The pipeline throughput rate refers to the number of tasks completed by the pipeline per unit time. The most fundamental formula for calculating the pipeline throughput rate is as follows:

\[ TP = \frac{N}{T_{\text{Pipeline}}} \]  

(15)

\( N \) is the number of completed tasks, and \( T \) represents the time it takes to complete these tasks in the pipeline. In this paper, the data processing of four range lines or azimuth lines is regarded as the number of tasks at one time.

Pipeline efficiency refers to the utilization rate of pipeline resources, which is one of the important indicators used to measure the idleness of the pipeline. Its calculation formula is as follows:

\[ E = \frac{Z_{\text{Occupied}}}{Z_{\text{Total}}} \times 100\% \]  

(16)

\( Z_{\text{Occupied}} \) is the space–time area occupied by the pipeline for completing \( N \) tasks, and \( Z_{\text{Total}} \) is the total space–time area.

The speedup ratio of the pipeline refers to the ratio of the time spent without the pipeline to the time spent with the pipeline when completing the same batch of tasks. The higher the speedup ratio of the pipeline the better, which reflects the quality of the effect of using the pipeline. Its calculation can be expressed by the following formula:

\[ SP = \frac{T_{\text{Non-pipeline}}}{T_{\text{Pipeline}}} \]  

(17)

\( T_{\text{Non-pipeline}} \) is the time used to complete this batch of tasks without using pipeline technology, and \( T_{\text{Pipeline}} \) is the time used to complete this batch of tasks under when using pipeline technology.

Figure 18. Schematic diagram of superscalar pipeline performance comparison. (a) Throughput rate; (b) Efficiency; (c) Speedup ratio.

Dual-port RAM must be used as the data buffer exchange region between the DDR3 and the processing engine for the superscalar pipeline data exchange engine to be implemented in the FPGA. The DDR3 burst transfer length is 8 in the actual experiment environment. The burst transmission data read by the two channels at the same time belong to four range lines for the range direction in accordance with the dual-channel, sub-matrix, equally divided, three-dimensional, cross-mapping storage method. For the azimuth direction, the burst transmission data read by the dual-channel at the same time also belong to the four azimuth lines. The utilization rate of the azimuth and range computing engines can thus be 100% after the data have been exchanged and transferred to the computing engine for calculation. In addition, compared with non-pipeline and traditional scalar pipeline, the superscalar pipeline shows greater performance improvements.
in pipeline throughput rate, efficiency, and speedup ratio, greatly reducing system idle
time and improving data exchange efficiency.

The experimental results show that the data storage exchange engine based on su-
perscalar pipeline proposed in this paper has a high storage access bandwidth. Both the
storage access bandwidths—the azimuth storage access bandwidth, which can reach up to
20.0 GB/s, and the range storage access bandwidth, which can reach up to 16.6 GB/s—have
significantly more bandwidth than the previous ones. With the pipeline design of the
superscalar structure, the raw data can be stored in two off-chip DDR3 with equal division,
which makes the storage capacity double, which will mean that the storage exchange
engine designed in this paper can adapt to the larger granularity of SAR raw data. This
design also reduces the matrix transposition time by 48% compared with the latest study,
effectively improving the data transposition efficiency. The single burst transmission data
belong to the four range directions and four azimuth directions. The two-dimensional data
acesses are balanced, so that the computing resource utilization rate reaches 100%, and the
idle resources are fully utilized. Finally, the design of the register addressing access used
to control and monitor the work and operation status of each module, the parameterized
configuration of data granularity, starting coordinates, normal start or Debug mode, etc.,
increases the flexibility and configurability of the data storage and exchange engine, which
can be embedded into the SAR imaging system as a custom IP core to make it adaptable to
more complex real-time processing system platforms for spaceborne SAR imaging while
ensuring high bandwidth and high exchange efficiency.

5.2.2. The SAR Imaging Processing System with Heterogeneous
“CPU + FPGA” Architecture

To complete the experiment and result evaluation, we embed the superscalar pipeline
data storage exchange engine as an IP into the SAR imaging processing system platform
with heterogeneous “CPU + FPGA” architecture for experimental verification and result
evaluation.

After deploying the complete heterogeneous imaging processing system to the ZCU106
platform and VC709 platform, the resource utilization on the VC709 side after the synthesis
and implementation steps in Vivado 2020.2 is shown in Table 8.

Table 8. Hardware resource utilization of Xilinx Virtex-7 FPGA VC709 development platform.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Utilization</th>
<th>Available</th>
<th>Utilization (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice LUTs</td>
<td>153,262</td>
<td>433,200</td>
<td>35.38%</td>
</tr>
<tr>
<td>LUT RAM</td>
<td>21,036</td>
<td>174,200</td>
<td>12.1%</td>
</tr>
<tr>
<td>Flip Flop</td>
<td>212,947</td>
<td>866,400</td>
<td>24.58%</td>
</tr>
<tr>
<td>Block RAM</td>
<td>1283</td>
<td>1470</td>
<td>87.28%</td>
</tr>
<tr>
<td>DSPs</td>
<td>580</td>
<td>3600</td>
<td>16.11%</td>
</tr>
</tbody>
</table>

The resource utilization of the ZCU106 side after the synthesis and implementation
steps in Vivado 2020.2 is shown in Table 9.

The Block RAM resource usage is high on the VC709 side because the superscalar
pipeline data storage exchange engine is deployed on the VC709 side for data buffer ex-
change. The calculation engine is also deployed to the VC709 side, and there are operations
such as FFT and multiplication in the calculation engine, so there is some usage for DSP
resources. The ZCU106 side mainly performs data transfer and algorithmic process control
applications, and does not perform arithmetic and buffer exchange operations. Therefore,
the ZCU106 side requires less resource consumption.
Table 9. Hardware resource utilization of Zynq UltraScale+ MPSoC ZCU106 development platform.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Utilization</th>
<th>Available</th>
<th>Utilization (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice LUTs</td>
<td>51,311</td>
<td>230,400</td>
<td>22.3%</td>
</tr>
<tr>
<td>LUT RAM</td>
<td>16,082</td>
<td>101,760</td>
<td>15.8%</td>
</tr>
<tr>
<td>Flip Flop</td>
<td>51,290</td>
<td>460,800</td>
<td>11.1%</td>
</tr>
<tr>
<td>Block RAM</td>
<td>45</td>
<td>312</td>
<td>14.4%</td>
</tr>
<tr>
<td>DSPs</td>
<td>0</td>
<td>1728</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

The SAR imaging processing experiment was carried out according to the above-mentioned test procedure. The SAR data with a granularity of 16,384 × 16,384 in the experiment were provided by the Taijing-4 01 satellite. The Taijing-4 01 satellite was successfully launched at Wenchang Space Launch Site on 27 February 2022. This is China’s first X-band commercial SAR satellite and has the ability to provide customers with diversified and high-quality SAR image products [44]. The SAR imaging results are represented by 8-bit grayscale images, and the pixel grayscale value ranges from 0 to 255. Figure 19 shows the SAR imaging results of the South Andaman Island in India, and compares the imaging results obtained from our “CPU + FPGA” heterogeneous SAR imaging system with the imaging results processed by MATLAB and the optical remote sensing image in the global map. In addition, we performed imaging processing on point target using the entire imaging system and compared the results with the point target imaging results obtained by MATLAB. The imaging results of “CPU + FPGA” for South Andaman Island in India, as well as the comparison with global map optical remote sensing image and MATLAB processing results, are shown in Figure 19. The imaging results of point target using “CPU + FPGA” and their comparison with MATLAB processing results are shown in Figure 20.

![Figure 19. Diagram of “CPU + FPGA” imaging result, global map optical remote sensing image and MATLAB processing result for South Andaman Island in India.](image-url)
Figure 20. Diagram of “CPU + FPGA” imaging result and MATLAB processing result for point target. (a) MATLAB imaging result; (b) CPU + FPGA imaging result.

After imaging processing for point target, the peak-to-side lobe ratio (PSLR) and integral side lobe ratio (ISLR) of the imaging results in the range direction and azimuth direction are evaluated [27,45]. The simulation parameters of the raw echo data related to the imaging process of the point target are shown in Table 10. The peak-to-side lobe ratio (PSLR) refers to the height ratio of the first side lobe to the main lobe in the sinc function waveform of the SAR image. The integral side lobe ratio (ISLR) is the ratio of side lobe energy to main lobe energy. The waveforms of the SAR imaging results from the “CPU + FPGA” imaging system and MATLAB in the range and azimuth directions are shown in Figure 21. The quality evaluation results of SAR imaging for point target are shown in Table 11.

Table 10. Parameters for MATLAB simulation of point targets.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward velocity</td>
<td>7390.3 m/s</td>
</tr>
<tr>
<td>Wavelength</td>
<td>0.0312 m</td>
</tr>
<tr>
<td>Bandwidth of LFM signal</td>
<td>120 MHz</td>
</tr>
<tr>
<td>Range sampling rate</td>
<td>144 MHz</td>
</tr>
<tr>
<td>PRF</td>
<td>5403.2 Hz</td>
</tr>
<tr>
<td>Pulse duration</td>
<td>20.3 µs</td>
</tr>
<tr>
<td>Chirp rate of LFM signal</td>
<td>$5.9113 \times 10^{12}$ Hz/s</td>
</tr>
<tr>
<td>SquintAngle</td>
<td>$-7.865 \times 10^{-4}$ rad</td>
</tr>
</tbody>
</table>
Table 11. The quality evaluation results of SAR imaging for point target.

<table>
<thead>
<tr>
<th>Result Source</th>
<th>Range PSLR/dB</th>
<th>Range ISLR/dB</th>
<th>Azimuth PSLR/dB</th>
<th>Azimuth ISLR/dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB Imaging</td>
<td>−13.34</td>
<td>−10.25</td>
<td>−13.02</td>
<td>−10.17</td>
</tr>
<tr>
<td>“CPU + FPGA” Imaging</td>
<td>−13.31</td>
<td>−10.23</td>
<td>−13.01</td>
<td>−10.08</td>
</tr>
</tbody>
</table>

From the above imaging results, the whole SAR imaging system is shown to have a good performance when dealing with area targets and point targets after the superscalar pipeline storage exchange engine is embed. For point target imaging, our “CPU + FPGA” heterogeneous SAR system imaging results are not very different from MATLAB in terms of peak sidelobe ratio and integral sidelobe ratio. The processing performance of the entire SAR imaging system is analyzed below. Imaging processing time is the most significant performance of spaceborne SAR real-time imaging performance. Therefore, it is one of the essential key factors to measure an SAR imaging processing system. Since many operations, such as data storage and exchange and data calculation and processing, are required during the imaging process, these operation times have a crucial impact on the total SAR imaging processing time. Therefore, it is of great significance to improve the performance of each part of the engine in the system. In this paper, we compare the imaging processing times of several recent SAR imaging processing systems. Table 12 shows the results of the comparison between the imaging processing time of the “CPU + FPGA” heterogeneous imaging processing platform embedded with the superscalar pipeline storage exchange engine and several recent SAR imaging systems.

Table 12. Performance comparison with previous SAR imaging processing systems.

<table>
<thead>
<tr>
<th>Year</th>
<th>System Architecture</th>
<th>Data granularity</th>
<th>Processing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022</td>
<td>CPU + FPGA</td>
<td>16,384 × 16,384</td>
<td>5.218 s</td>
</tr>
<tr>
<td>2021</td>
<td>FPGA</td>
<td>16,384 × 16,384</td>
<td>6.957 s</td>
</tr>
<tr>
<td>2018</td>
<td>FPGA</td>
<td>16,384 × 16,384</td>
<td>10.6 s</td>
</tr>
<tr>
<td>2017</td>
<td>FPGA + ASIC</td>
<td>16,384 × 16,384</td>
<td>12.1 s</td>
</tr>
</tbody>
</table>

Figure 22 shows that the SAR imaging processing system using the superscalar pipeline storage and exchange engine proposed in this paper leads to a large improvement in processing time and storage access bandwidth compared with other processing systems developed in recent years. The system architecture is used for algorithm and system vali-
cation on the ground platform, but is also meaningful for the on-orbit real-time processing of spaceborne SAR.

![Graph showing performance comparison](image)

**Figure 22.** Schematic diagram of the performance comparison between our proposed high-efficiency data storage and exchange engine and Yang et al. 2017 [33], Li et al. 2018 [42], Sun et al. 2019 [34] and Wang et al. 2021 [35]. (a) Storage access bandwidth; (b) Imaging processing time.

6. Conclusions

This study designed and implemented an efficient on-chip data storage and exchange engine predicated on a superscalar pipeline for spaceborne SAR real-time processing. The storage exchange engine significantly augments data storage access bandwidth in both the range and azimuth directions while ensuring balanced data access. The utilization rate of the computing engine in both range and azimuth directions can achieve 100%. The superscalar structure’s pipeline design enhances the storage and exchange efficiency of voluminous data. Experimental results reveal that the proposed data storage exchange engine, based on the superscalar pipeline, could reach 16.6 GB/s in the range direction, reflecting a 62.1% increment compared to conventional methods. The storage access bandwidth in the azimuth direction could attain 20.0 GB/s, a 95.3% increase from traditional methods. Moreover, compared to no-pipeline and scalar-pipeline designs, the storage and exchange engine yielded performance enhancements in throughput, pipeline efficiency, and speed. The feasibility of this experiment was corroborated on the “CPU + FPGA” heterogeneous SAR imaging processing platform integrated with this engine. The outcomes highlight that the data storage and exchange engine based on the superscalar pipeline further diminishes the imaging system’s processing time, boosting processing efficiency. In the context of hardware implementation for uninhabited aerial vehicle synthetic aperture radar (UAVSAR) [46,47] and convolutional neural networks (CNNs) [48–52], among other applications, the data storage and exchange method predicated on the superscalar pipeline proposed in this paper holds value. Future research will explore efficient storage and exchange methods adapted to multi-channel, multi-mode, and other real-time processing scenarios to satisfy the demands of emerging remote sensing applications.

**Author Contributions:** Conceptualization, H.L. and Y.L.; methodology, H.L., Y.L. and T.Q.; software, H.L., T.Q. and Y.X.; validation, H.L.; formal analysis, T.Q. and Y.X.; investigation, H.L., Y.L. and Y.X.; resources, T.Q. and Y.X.; writing—original draft preparation, H.L. and Y.L.; writing—review and editing, H.L., Y.L., T.Q. and Y.X.; supervision, Y.X.; project administration, Y.X. and T.Q.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.
Funding: This research was funded by the National Key Research and Development Program of China under Grant 2021YFA0715204, the China ZYQN Program under Grant 2018-JCJQ-ZQ-046 and in part by the Foundation under Grant SAST2021-040.

Data Availability Statement: Not applicable.

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

References


15. SAOCOM—Earth Online. Available online: https://earth.esa.int/eogateway/missions/saocom (accessed on 29 August 2022).


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.