Journal of Electronic Research and Application, 2026, Volume 10, Issue 3
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Research on Optimization of FPGA Streaming Processing System for High-Bandwidth Radar Echo Data

Zhonghao Jiang

Institute of Intelligent Manufacturing,Chongqing Technology and Business Institute, Chongqing 400052, China

Abstract: For the real-time processing scenario of high-bandwidth radar echo data, an optimization scheme for a streaming processing system based on FPGA is proposed. Focusing on the engineering implementation requirements under conditions of high throughput, low latency, and resource constraints, the overall system architecture is designed, and optimizations are carried out in three aspects: pipeline parallel computing, cache organization and memory access scheduling, and timing convergence under resource constraints. The system adopts a modular streaming data path, builds a collaborative mechanism between on-chip cache and computing units, reduces data transfer overhead, and improves the continuity and stability of the processing link. Experimental results show that the optimized system can operate stably under higher input bandwidth conditions, with improved throughput capacity and real-time processing performance, compressed critical path, and resource utilization remaining within a reasonable range. This research can provide a reference for the engineering implementation of high-bandwidth radar signal processing platforms.

Keywords: High-bandwidth radar echo; FPGA; Stream processing; System optimization

Online publication: April 22, 2026

1. Introduction

The improvement of radar system resolution and detection accuracy has led to continuous growth in echo sampling rate, data bit width, and channel scale. This has placed greater demands on the data throughput and real-time constraints of the signal processing platform. The traditional serial processing architecture has struggled to meet the engineering application requirements in terms of bandwidth carrying capacity, memory access efficiency, and power consumption control. FPGA, with its high parallelism, reconfigurability, and low latency characteristics, has significant advantages in the construction of high-speed data paths. Conducting research on the optimization of streaming processing systems for high-bandwidth echo scenarios is of practical significance for enhancing the engineering implementation capabilities of radar signal processing platforms [1].

2. FPGA streaming processing system architecture design for high-bandwidth radar echo data

2.1. Analysis of processing requirements for high-bandwidth radar echo data

High-bandwidth radar echo data has the characteristics of high sampling rate, continuous data arrival, and concentrated processing pressure. In the engineering implementation, the system not only needs to meet the stable data reception capability, but also needs to take into account the on-chip cache depth, computing throughput capacity, and processing delay constraints. If the data transfer rate is lower than the input rate, it is prone to cause cache accumulation and link congestion; if the parallelism of the computing unit is insufficient, the real-time processing requirements are difficult to meet [2]. For this type of application, a demand model needs to be established from three aspects: input data rate, processing delay, and computing load.

$$R_{\text{in}}=N_{\text{ch}}\cdot f_s\cdot Q\cdot\eta \tag{1}$$

Where R_{in} represents the data rate of radar echo input, measured in bit/s; N_{ch} denotes the number of parallel receiving channels; f_s indicates the sampling rate of a single channel, measured in Hz; Q represents the quantization bit width of a single sampling point, measured in bits; η is the coefficient of data encapsulation and interface overhead.

$$T_{\text{proc}}\le T_{\text{PRI}}-T_{\text{buf}} \tag{2}$$

Where T_{proc} represents the total processing delay for single-pulse echo data, measured in seconds; T_{PRI} is the pulse repetition interval, measured in seconds; T_{buf} is the time occupied by buffer scheduling and data transfer, also measured in seconds.

$$C_{\text{tot}}=\frac{N_{\text{ch}}N_s}{T_{\text{PRI}}}\left(\alpha N_{\text{FFT}}\log_2 N_{\text{FFT}}+\beta N_{\text{mf}}+\gamma N_{\text{acc}}\right) \tag{3}$$

Where C_{tot} represents the total computing load required per unit time; N_s is the number of sampling points for a single channel and single pulse; N_{FFT} is the number of Fourier transform points; N_{mf} is the scale of matching filter calculation; N_{acc} is the scale of accumulation operation; α, β, and γ are the conversion coefficients for corresponding processing stages.

The above analysis indicates that the high-bandwidth radar echo processing requirements are not a single-bandwidth issue, but rather a systematic constraint resulting from the coupling of data throughput, computing load, and real-time response. The subsequent architecture design should be based on this constraint relationship [3].

2.2. Overall architecture design of FPGA streaming processing system

Given the characteristics of continuous arrival of high-bandwidth radar echo data, long processing link, and significant differences in throughput requirements among modules, the system adopts a data-flow-oriented hierarchical architecture. It integrates high-speed input, data caching, parallel computing, and result output into a unified data path. Each processing module is cascaded in a streaming transmission manner, reducing intermediate write-back and repetitive movement, and ensuring the continuity and stability of the processing process. To ensure the overall operational efficiency of the system, the matching relationship between input and processing capabilities needs to be described:

$$R_{\text{sys}}=\min\left(R_{\text{in}},R_{\text{buf}},R_{\text{comp}},R_{\text{out}}\right) \tag{4}$$

Where R_{sys} represents the effective throughput rate of the system; R_{in} denotes the throughput rate of the input interface; R_{buf} indicates the data exchange rate supported by the buffer module; R_{comp} signifies the processing rate

of the computing module; R_{out} represents the transmission rate of the output link.

Based on the above relationships, the overall system architecture is shown in **Figure 1**.

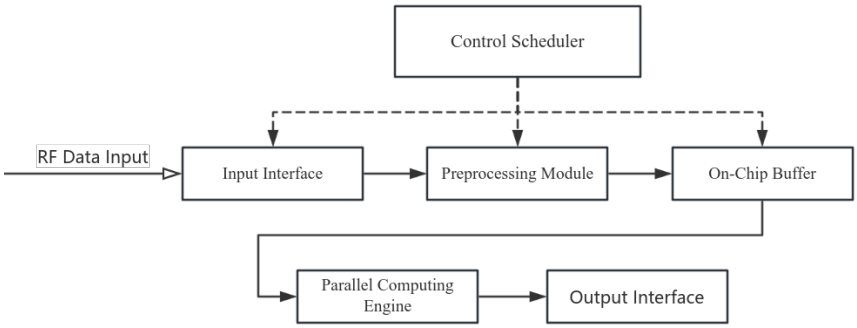

Figure 1. Overall architecture of the FPGA streaming processing system.

3. High-bandwidth radar echo data FPGA streaming processing optimization method

3.1. Parallel computing optimization based on pipeline structure

The processing link of high-bandwidth radar echo data includes multiple computing steps such as data rearrangement, frequency domain transformation, filtering, and accumulation. A single-level serial structure is difficult to meet the real-time processing requirements under continuous input conditions [4]. To address this issue, the system divides the core computing process into several functionally independent sub-levels, advancing them in sequence driven by the clock, enabling different stages to process different batches of data in parallel. This approach can shorten the combinational logic length within a single cycle, increase the hardware operating frequency, and is also beneficial for stabilizing the output rate. To characterize the throughput capability of the pipeline parallel structure, the system processing rate can be expressed as:

$$P_{\text{pipe}}=f_{\text{clk}}\cdot N_{\text{par}} \tag{5}$$

Where P_{pipe} represents the theoretical processing rate of the pipeline computing unit, measured in samples/s; f_{clk} represents the system operating clock frequency, measured in Hz; N_{par} represents the amount of data processed in parallel within a single clock cycle.

After the parallelism is improved, the overall acceleration effect of the system is also influenced by the number of pipeline stages, the degree of inter-stage balancing, and the control overhead. To describe the changes in computing performance before and after optimization, an acceleration ratio model can be used:

$$S=\frac{T_{\text{serial}}}{T_{\text{pipe}}}=\frac{N\cdot t_{\text{op}}}{(k+N-1)t_{\text{clk}}+T_{\text{ctrl}}} \tag{6}$$

Where S represents the parallel acceleration ratio of the pipeline; T_{serial} is the time required for the serial structure to complete all calculations; T_{pipe} is the processing time of the pipeline structure; N is the number of batches of data to be processed; top is the average processing time per batch in the serial mode; k is the number of pipeline levels; t_{clk} is the processing time of a single level pipeline within one clock cycle; T_{ctrl} is the additional overhead caused by control scheduling and inter-level synchronization.

From **Equations (5)** and **(6)**, it can be seen that reasonably increasing the parallelism and maintaining load balance at all levels is the key to improving the efficiency of stream computing.

3.2. Cache and memory access optimization for high-bandwidth data streams

For high-bandwidth radar echo data under continuous input conditions, there are higher requirements for cache organization and memory access efficiency. If the on-chip cache depth is insufficient, data bursts arriving simultaneously are prone to accumulate; if there are waiting or conflicts in external storage access, computing units will become idle, and the system throughput will decrease. To address this issue, the design adopts a hierarchical cache and dual-buffer coordination mechanism, decoupling the data reception, relocation, and computing processes, so that when one set of data enters the computing stage, another set of data can simultaneously complete writing or reading, reducing the impact of memory access blocking on the processing link. To describe the cache capacity requirements, the minimum cache depth can be expressed as:

$$D_{\text{buf}} \geq R_{\text{in}} \cdot T_{\text{lat}} \tag{7}$$

Where D_{buf} represents the minimum required capacity of the cache module, measured in bits; R_{in} indicates the input data rate, measured in bit/s; T_{lat} denotes the maximum tolerable delay for data waiting to be processed or moved, measured in seconds.

At the level of memory access scheduling, the system enhances the utilization of effective bandwidth through burst transmission, contiguous address mapping, and Bank partitioning methods. After considering memory access conflicts and control overheads, the effective bandwidth of the storage system can be expressed as:

$$B_{\text{eff}} = B_{\text{peak}} \cdot \eta_{\text{burst}} \cdot \eta_{\text{bank}} \cdot \eta_{\text{ctrl}} \tag{8}$$

Where B_{eff} represents the effective memory bandwidth, measured in bit/s; B_{peak} represents the theoretical peak bandwidth of the memory; η_{burst} is the burst transmission efficiency coefficient; η_{bank} is the efficiency coefficient for multi-Bank parallel access; η_{ctrl} is the control scheduling efficiency coefficient.

The cache depth and effective bandwidth jointly determine the continuous supply capacity of data flow, and are also an important condition for ensuring the stable operation of the stream processing link.

3.3. FPGA resource and timing optimization design

During the optimization process of the high-bandwidth radar echo streaming processing system, not only the improvement of throughput rate needs to be considered, but also the occupation of logic resources, on-chip memory allocation, and the difficulty of timing convergence should be taken into account. Increasing the parallelism can enhance the processing capability, but it will also lead to an increase in the consumption of LUTs, registers, DSPs and BRAMs. The length of the critical path expands with the increase in module complexity, and the maximum operating frequency of the system is limited. In the design, methods such as reusing arithmetic units, inserting critical path registers, localizing control logic and constraining data bit width are adopted to jointly optimize the resource consumption and timing performance. To describe the relationship of resource consumption under parallel configuration, the overall resource requirement can be expressed as:

$$U_{\text{res}}=N_{\text{par}}(\lambda_1 U_{\text{LUT}}+\lambda_2 U_{\text{FF}}+\lambda_3 U_{\text{DSP}}+\lambda_4 U_{\text{BRAM}})+U_{\text{ctrl}} \tag{9}$$

Where U_{res} represents the overall resource consumption of the system; N_{par} indicates the number of parallel processing units; U_{LUT}, U_{FFU}, U_{DSP} and U_{BRAM} correspond to the lookup table, register, DSP and on-chip storage resource consumption of a single parallel unit respectively; λ_1, λ_2, λ_3 and λ_4 are the different resource conversion weights; U_{ctrl} represents the additional resource consumption brought by the control and interface logic.

The core of timing optimization lies in reducing the delay of the critical path, so that the system can meet the

target clock constraints. After pipelining the registers and splitting the logic, the clock cycle should satisfy:

$$T_{\text{clk}} \geq T_{\text{comb}}+T_{\text{reg}}+T_{\text{skew}}+T_{\text{margin}} \tag{10}$$

Where T_{clk} represents the system clock cycle; T_{comb} denotes the combinational logic delay on the critical path; T_{reg} indicates the delay related to register setup and hold; T_{skew} represents the delay introduced by clock skew; T_{margin} is the timing design margin.

Resource constraints and clock constraints are coupled with each other. Only when a balance is achieved among the parallel scale, module division, and timing convergence, can the system maintain stable operation under high bandwidth conditions.

4. System implementation and performance verification

4.1. FPGA system implementation and module design

The system implementation aims at the continuous reception and real-time processing of high-bandwidth radar echo data. The hardware platform uses FPGA as the core processing component. A complete processing link is constructed around the input interface, data cache, computing unit, control scheduling, and result output. The input end completes the parallel reception and byte alignment of high-speed echo sampling data. The pre-processing module is responsible for data format conversion, effective data extraction, and timing arrangement, providing stable input for subsequent calculations. The on-chip cache module adopts a partitioned management method, isolating data writing, reading, and computing processes to reduce the impact of sudden traffic on the processing link. The core computing part is divided into several independent functional units according to the echo processing flow. Each unit operates in a unified clock domain and is connected according to the stream interface, with data advancing along a fixed path after entering, avoiding frequent backwriting of intermediate results to external storage. The control scheduling module is responsible for state switching, cache address management, and module handshake control, ensuring stable coordination of each processing stage under load changes [5].

Module design emphasizes clear structure and engineering feasibility. The internal computing unit adopts cascaded data paths, mapping multiplication, addition, transformation, and accumulation operations to processing structures suitable for hardware expansion, facilitating subsequent parallel expansion and timing optimization [6]. For common data blocking problems in high-bandwidth scenarios, the system sets a flow control mechanism between the cache and the computing unit. It dynamically adjusts the reading and writing rhythm based on the buffer status to maintain a matching relationship between the input rate and the computing rate. The interface design adopts a standardized packaging method to reduce the coupling between modules, facilitating subsequent independent debugging and function reuse. During the implementation process, local buffering processing was carried out on key control paths and high-fanout signals to shorten the combinational logic depth and improve overall timing stability. After modular division, the system has good scalability. When the input bandwidth or processing scale increases, the structure can be expanded by increasing the number of parallel units or adjusting the cache configuration. The system implementation structure is shown in **Figure 2**.

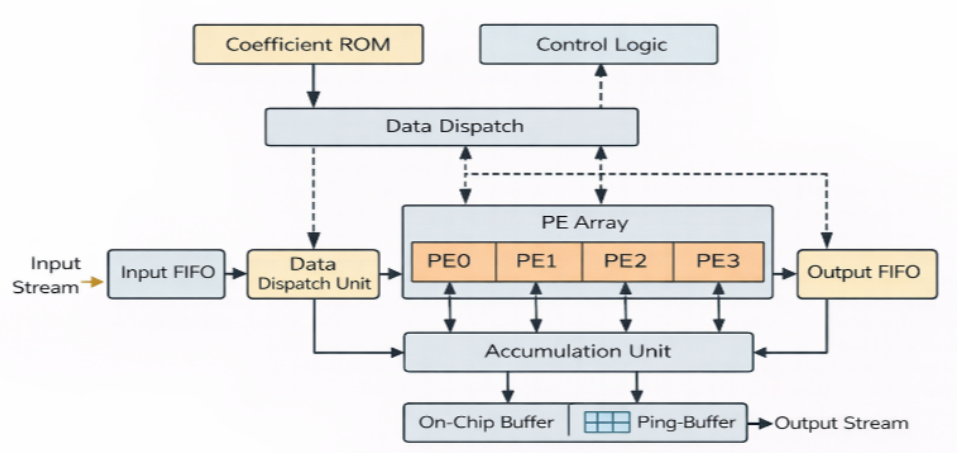

Figure 2. FPGA system implementation and module design.

4.2. System resource utilization and timing performance analysis

After the system completed the synthesis, layout and timing analysis, the usage of major hardware resources and key timing indicators were statistically analyzed. The results showed that the resource allocation of each functional module was relatively balanced. The computing units and on-chip cache accounted for the majority of the system's hardware costs, which was consistent with the parallel computing requirements and data buffering requirements during the high-bandwidth echo data processing process. The proportion of control scheduling and interface logic was relatively low, and it did not cause significant pressure on the overall resource distribution. The overall implementation results showed that the utilization rates of LUTs, registers, BRAMs and DSPs remained within the acceptable range of the device, and a certain expansion space was reserved to support subsequent adjustments for higher input bandwidth or larger processing scales [7].

From the perspective of timing performance, after key path optimization and register insertion processing, the system could meet the target clock constraints, and no significant negative slack issues were found in the main data paths. The key paths were concentrated between the parallel computing module and the cache control module, indicating that the coordination of computing access and memory access under high-speed data flow remains an important factor affecting timing convergence [8]. The analysis of system resource utilization and clock performance revealed that the current implementation structure achieved a reasonable balance among throughput capacity, resource consumption and timing stability. The specific resource utilization and timing results are shown in **Table 1**.

Table 1. System resource utilization and timing analysis results

Module	LUT	FF	BRAM	DSP	Maximum operating frequency / MHz	Timing slack / ns
Input and preprocessing module	4210	3896	12	8	268	1.84
On-chip buffer module	3568	3014	28	0	251	1.27
Parallel computing module	9824	8742	16	64	243	0.96
Control scheduling module	2146	2378	4	0	286	2.15
Output interface module	1875	1692	6	2	272	1.73
System total	21623	19722	66	74	243	0.96

4.3. Experimental results and system performance evaluation

After the system was fully implemented, functional verification and performance testing were conducted for the scenario of high-bandwidth continuous echo input, focusing on throughput rate, processing delay, bandwidth utilization, and operational stability. The test results indicated that the optimized streaming processing system could maintain stable operation under continuous data input conditions, without any obvious data accumulation or link congestion. Compared with the unoptimized implementation, the system performed better in terms of throughput capacity and delay control, demonstrating the effectiveness of the pipeline parallel structure, cache scheduling method, and resource timing optimization. The improvement in bandwidth utilization indicated an enhanced matching between data transfer and computation processes, and the overall real-time processing capability of the system met the design expectations. The specific experimental results are shown in **Table 2**, and the performance change trends before and after optimization are depicted in **Figure 3**.

Table 2. System performance evaluation results

Metric	Before optimization	After optimization	Improvement
Throughput / Gbps	9.86	12.56	27.4%
End-to-end latency / μs	34.4	27.9	18.9%
Maximum operating frequency / MHz	214	243	13.6%
Bandwidth utilization / %	78.2	89.5	11.3%
Continuous operation stability time / h	12	12	-

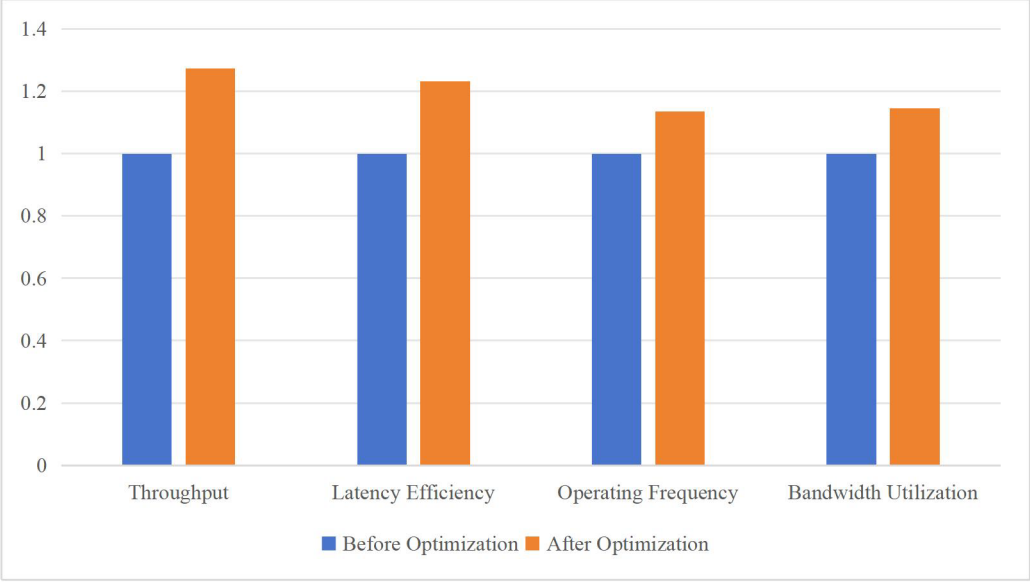

Figure 3. Normalized comparison of system performance before and after optimization.

As shown in **Figure 3**, the optimized system demonstrates significant improvements in terms of throughput capacity, delay control, and bandwidth utilization. This indicates that the proposed streaming processing optimization method can effectively enhance the overall performance of the high-bandwidth radar echo data processing system.

5. Conclusion

This paper focuses on the real-time processing requirements of high-bandwidth radar echo data and conducts research on the design and optimization of a streaming processing system based on FPGA. By addressing key issues such as system throughput, cache access, parallel computing, and timing constraints, the overall architecture design, optimization method analysis, and engineering implementation have been completed. The design concept proposed in this paper can well adapt to the scenario of continuous high-bandwidth data input, achieving a balance between processing efficiency, structural stability, and resource consumption. The related research has certain reference value for the engineering design of high-performance radar signal processing platforms and also provides a foundation for subsequent system expansion targeting larger data scales.

Disclosure statement

The authors declare no conflict of interest.

References

[1] Altalqi F, Fennane S, Kacimi H, et al., 2024, Development of a Multi-Band High Bandwidth Circular Microstrip Patch Antenna for Radar Implementations, 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM), 1–5.

[2] Altalqi F, Fennane S, Mabchour H, et al., 2024, Design Monopole Antenna of Ultra-Wideband High Bandwidth and High Efficiency for Ground Penetrating Radar Application. Telkomnika, 22(4): 154–163.

[3] Hong A, Su L, Wang Y, et al., 2025, A Terahertz Dual-Band Transmitter in 40 nm CMOS for a Wideband Sparse Synthetic Bandwidth Radar. Electronics (2079-9292), 14(22): 24–30.

[4] Liu M, Xie X, Deng Y, et al., 2023, Efficient Implementation of BP Imaging Algorithm on FPGA. IET Conference Proceedings, 2023(47): 1092–1097.

[5] Jonsson R, Ankel M, Tholen M, et al., 2023, Experimental Analysis of a Clutter Suppression Algorithm for High Time-Bandwidth Noise Radar, 2023 IEEE International Radar Conference (RADAR), 1–6.

[6] Xie Y, Zhong Z, Li B, et al., 2024, An ARM-FPGA Hybrid Acceleration and Fault Tolerant Technique for Phase Factor Calculation in Spaceborne Synthetic Aperture Radar Imaging. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2024(17): 5059–5072.

[7] Ankel M, Tholen M, Bryllert T, et al., 2024, Implementation of a Coherent Real-Time Noise Radar System. IET Radar, Sonar & Navigation, 18(7): 1002–1013.

[8] Zhang Z, Xu Y, Li Y, et al., 2023, Spaceborne SAR Imaging System with High-Performance Polynomial Engines. IET Conference Proceedings, 2023(47): 2435–2442.