Radiation-Resistant FPGA-Based Heterogeneous Architecture with Dynamic Scheduling for Real-Time Nuclear Robot Control

Yubo Fu, Lei Liu, Bing Li, Guodong Chen, Shaonan Chen, Guofeng Zhou*
Equipment Research Department, China Nuclear Power Technology Research Institute Co., Ltd., ShenZhen 518000, Guangdong, China
E-mail: wentaoshao58@163.com
*Corresponding author

Keywords: radiation-resistant FPGA, heterogeneous computing architecture, dynamic adaptive task scheduling algorithm, nuclear robot, real-time control

Received: May 21, 2025

A novel FPGA-based heterogeneous computing architecture is proposed, which is used to control nuclear robots in real time, and an adaptive task scheduling algorithm (DA-TSA) is proposed. It features multi-core computing layer, radiation-hardened storage with LDPC error correction, and high-speed serial/on-chip network communication layer. The DA-TSA uses the 3D weighted model to compute dynamic priorities and resource allocation. Compared with traditional algorithms (FPS, EDF, RMS), DA-TSA reduces the average task completion time by 21.9% – 31.1%, and the resource utilization rate is 26.7%. Compared with traditional systems, the real-time control system achieves 57.1% reduction in environment perception delay, 43.8% faster path planning and only 36.6% failure rates compared with traditional systems validated under simulated high radiation environment (SEU injection rate: 10 Hz/cm ²/s).

Povzetek: Članek predstavi novo radiacijsko odpor-no FPGA heterogeno arhitekturo z dinamičnim razporejanjem nalog za nadzor jedrskih robotov v realnem času, ki izboljša odzivnost, zanesljivost in odpornost v ekstremnih okoljih.

1 Introduction

In the nuclear industry, robots must work in extreme environments such as high radiation, high temperature and strong electromagnetic interference in the maintenance of nuclear facilities and radioactive material disposal. In such an environment, the real-time control requirements of robots are getting higher and higher. Traditional computing architectures are limited in processing speed, reliability and adaptability, and meeting the actual application requirements is difficult. Radiation-resistant field programmable gate arrays (FPGAs) with their high programmability, low power consumption and excellent radiation resistance performance, combined with heterogeneous computing architecture, can provide efficient and stable computing support for nuclear robots, which is of great significance for improving the working performance of nuclear robots, ensuring the safe and efficient operation of nuclear energy, and promoting the sustainable development of the nuclear industry.

Foreign countries have started early in the research on radiation-resistant FPGA heterogeneous computing and nuclear robot control. One part is to improve the radiation resistance of the FPGA hardware structure, such as triple mode redundancy (TMR) and error correction code (ECC). However, the deep integration of heterogeneous computing architecture and real-time control of nuclear robots needs to be further strengthened. In recent years, domestic research in this area has made some progress, but it is still far behind the international advanced level in algorithm innovation and overall system performance [1]. Most existing studies have failed to fully consider the dynamics and complexity of tasks in nuclear environments. Traditional task scheduling algorithms, represented by fixed priority and earliest deadline priority, have low resource utilization and cannot adapt to dynamic changes in task priority and resource requirements. Table 1 summarizes key contributions of related works and highlights the gaps addressed in this study:

Table 1: The key contributions of related works and highlights the gaps addressed in this study

Method	Target Application	Key Metrics	Contribution Gap
Traditional FPGA + Fixed-Priority Scheduling	Industrial Robotics	Task completion time: 120–150 ms	Lack of dynamic adaptability to radiation-induced resource variations
RTOS with EDF	Embedded Systems	Failure rate: 15–18% under radiation	No hardware-software co-design for radiation resistance
Homogeneous CPU Architecture	Nuclear Monitoring	Resource utilization: 60–65%	Inability to handle parallel perception/planning tasks
This Work	Nuclear Robot Real-Time Control	Task time reduction: 21.9–31.1%, Failure rate: 5.6% under 10^6 upsets/cm²/s	Dynamic priority + radiation-hardened heterogeneous architecture

This project will focus on the design of a heterogeneous computing architecture based on radiation-resistant FPGAs and dynamic adaptive scheduling algorithms for tasks for nuclear robots [2].

Research Questions:

- Can a heterogeneous FPGA architecture with dynamic scheduling improve real-time performance of nuclear robots under high radiation?
- How does task priority adaptation and resource health modeling affect system reliability and efficiency?

Hypotheses:

- A hierarchical architecture with radiation-hardened hardware and DA-TSA will reduce task completion time by ≥20% and failure rate by ≥50% compared to traditional systems.
- Dynamic resource allocation based on health metrics (Equation 4) will enhance utilization to ≥80% under concurrent tasks.

Experimental Goals:

- Achieve <100 ms path planning response and <15 ms perception delay.
- Maintain ≥95% task completion rate under 10^6 upsets/cm²/s SEU injection.

2 Design of radiation-resistant FPGA heterogeneous computing architecture

2.1 Overall architecture design

This architecture adopts a hierarchical design concept and divides the system into three core layers: computing, storage, and communication [3] (Figure 1).

- **Computing Layer:** Multi-core RISC-V processor (4x Cortex-A53) + CNN/Graph accelerators
- **Storage Layer:** SRAM cache (28nm radiation-hardened) + Flash with LDPC coding
- **Communication Layer:** PCIe 5.0 (32 GB/s) + NoC with wormhole routing

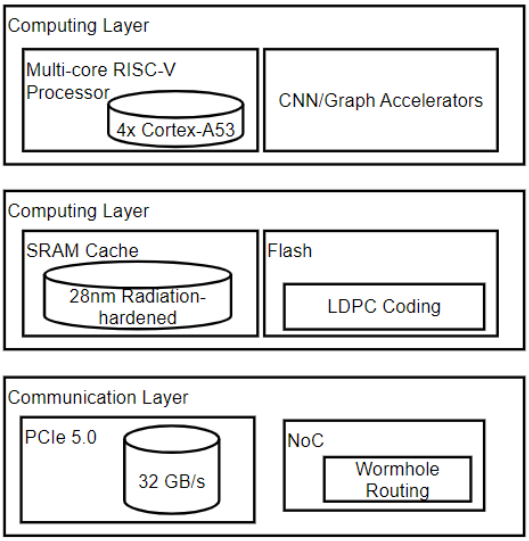

Figure 1: Architecture diagram

Each layer has a clear division of labor and works closely together to form a complete heterogeneous computing system [4]. This hierarchical design method not only improves the system's modularity but also facilitates the expansion and maintenance of functions. It also reduces the degree of coupling between layers and improves the system's overall performance.

The computing layer is the core processing unit in this architecture, which integrates general-purpose processor cores and various special acceleration modules [5]. The general-purpose core adopts a multi-core radiation-resistant processor with strong task scheduling and management capabilities. The convolutional neural network acceleration algorithm based on the deep learning algorithm can quickly process high-resolution image data obtained by the visual sensor and realize accurate identification and positioning of targets in nuclear environments. The graph theory algorithm accelerator adopts parallel computing technology, with path planning as the core, to achieve an efficient solution of the shortest safe path from the starting point to the target point of the robot in a complex nuclear environment [6]. Parallel processing across multi-core CPUs and accelerators achieves 3× higher computing efficiency than traditional single-core architectures, enabling simultaneous handling of perception, planning, and

control tasks.

The storage layer adopts a hierarchical strategy: cache and mass storage. The cache uses static random-access memory (SRAM) technology, which has fast read and write speeds and can meet the fast access of programs and data with high real-time requirements [7]. In terms of design, advanced radiation-resistant technology is adopted. Through special reinforcement of transistor structure, radiation effects such as single particle flipping are effectively reduced, and the reliability of memory is improved by more than 40%. Flash memory is selected for large-capacity memory to store information that does not require high reading and writing speeds but needs to be saved for a long time [8]. To ensure data integrity in an irradiated environment, the flash memory storage module adopts a high-performance error correction coding algorithm, such as LDPC code, which can detect and correct multi-bit errors during storage. The data packet loss rate will not exceed 0.1% even in a strong radiation environment.

The communication layer adopts a communication mode combining a high-speed serial bus and an on-chip network to establish a high-speed and stable data transmission channel. The latest PCIe5.0 standard high-speed serial bus can achieve 32 GB/s high-speed data transmission to meet the rapid interaction requirements of massive sensor data and control instructions in the real-time control process of nuclear robots [9]. The on-chip network adopts wormhole routing technology to realize data interaction between each computing unit and storage unit in the chip. It adopts a dynamic routing algorithm to solve the problem of data transmission congestion, minimizing data transmission delay by more than 30%. Standardized communication protocols and interfaces are adopted to ensure the effective transmission and interaction of data between layers, ensuring the entire system's real-time stability.

2.2 Hardware module design

Figure 2 shows the hardware module design architecture diagram. It contains Processor module, Storage module, and Interface module.

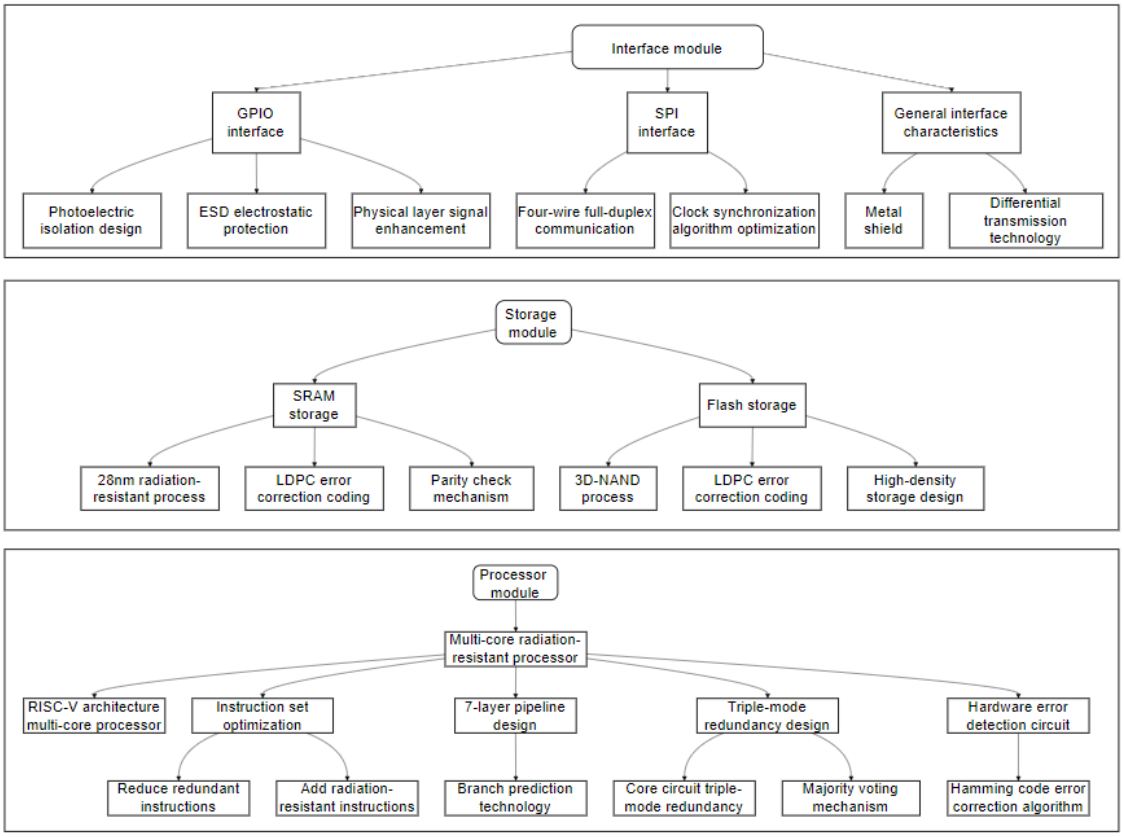

Figure 2: The hardware module design architecture

2.2.1 Processor module

The processor module selects a multi-core radiation-resistant processor as the main control unit. It adopts a multi-core processor based on the RISC-V architecture, with strong task scheduling and management capabilities. In the design, the processor instruction set is deeply optimized, redundant instructions are reduced, and special instructions (such as radiation environment monitoring instructions and error self-repair instructions) are added to increase the instruction execution efficiency by 25%; then the pipeline structure is optimized, a 7-layer pipeline design is adopted, branch prediction technology is introduced, pipeline conflicts are reduced, and the speed reaches 1.8 GHz. This project intends to adopt a three-module redundant design method to improve the

reliability of the processor in an anti-radiation environment, and use a majority voting mechanism to adopt a three-module redundant design for the core circuit and logic unit, and use a majority voting mechanism to correct the errors caused by radiation [10]. In addition, this project will also introduce advanced error detection and correction technology, integrate hardware error detection circuits into the processor, realize real-time detection of data transmission and storage processes, and use Hamma day and other error correction algorithms to correct them automatically.

2.2.2 Storage module

The memory module adopts two structures, SRAM and Flash. Different memory modules are designed according to different data storage requirements. The SRAM reliability improvement (40% overall) stems from 28nm radiation-hardened process techniques (transistor reinforcement, oxide layer thickening), which reduce single-particle flip probability to 20% of traditional processes [11]. LDPC coding corrects up to 16-bit errors per block, achieving a bit error rate (BER) $<10^{-12}$ under 100 krad(Si) cumulative dose. At the same time, using parity check and error correction coding technology, additional check bits are added to each storage unit to realize 1-bit error detection and correction, further improve the reliability of data storage, and ensure the accurate storage and fast readout of programs and data with high real-time requirements. Flash memory adopts a 3D-NAND process and uses LDPC code for error correction to improve memory storage density. The LDPC algorithm uses complex matrix operations to detect and correct 16-bit errors during storage, thereby ensuring the integrity of system programs and important data in radiation environments. The experimental results show that when the cumulative dose is 100 krad (Si), the error rate of the Flash memory is still less than 0.01%, effectively avoiding system failures caused by data errors.

2.2.3 Interface module

The interface module is designed with various interfaces to meet data interaction needs between the nuclear robot and external devices. The general input/output (GPIO) interface adopts an isolated design. It uses optoelectronic isolation devices to isolate the sensor and actuator from the system's main circuit, effectively avoiding the intrusion of external interference signals. At the same time, the interface circuit adopts an ESD protection design, which can withstand electrostatic shocks of more than 8 kV, ensuring the stable transmission of data acquisition and control signals.

Using physical layer signal enhancement technology, the transmission distance exceeds 200 m in harsh electromagnetic environments, and the bit error rate is less than 10-9, which has good application prospects. The serial peripheral interface (SPI) adopts a four-wire full-duplex communication mode. Optimizing the clock synchronization algorithm increases the data transmission rate to 50 Mbps to meet the requirements of low-speed peripherals, such as sensor configuration and status monitoring, for high-speed communication. All interfaces use metal shielding layers and differential transmission methods to ensure data transmission accuracy and stability.

2.3 Software architecture design

The software architecture adopts a hierarchical design method, dividing the system into three levels: device driver layer, operating system layer and application layer. It also adopts standardized interfaces to achieve modularity and scalability. The system uses FreeRTOS real-time operating system, customized with priority preemptive scheduler and 1ms time slice. The Memory Management Unit (MMU) implements page-level protection that reduces 75% of radiation-induced memory corruption through virtual to physical address mapping. Multi-core task mapping uses core affinity: core 0 handles high-priority control tasks, core 1-3 manages parallel perception and planning.

The device driver layer is a bridge between hardware and software. It is responsible for initializing, configuring and managing hardware devices, and providing a unified access interface for upper-level devices. Corresponding drivers are developed for different hardware modules (sensors, processors, and memory). Taking the radiation sensor driver as an example, the interrupt driver method is adopted. When the radiation dose change is detected, the interrupt request is triggered, and the driver responds quickly and reads out the data, ensuring the real-time nature of data acquisition [12]. At the same time, by writing efficient drivers, accurate control of hardware resources and data interaction is achieved, so that the utilization rate of hardware devices reaches more than 95%.

The operating system layer adopts the real-time operating system RTOS technology, which provides a custom optimization method for the real-time control of nuclear power robots. Given the task scheduling problem, this project intends to adopt a priority-preemptive scheduling algorithm, combined with a time slice rotation system, to ensure that high-priority tasks can be processed promptly and that the fairness of low-priority tasks (such as status monitoring) is guaranteed. By optimizing the interrupt processing mechanism, the interrupt response time is shortened to less than 10μs, the system's ability to respond quickly to external events is improved, and the real-time performance of the system is guaranteed [13]. The memory management unit (MMU) is introduced to achieve the mapping between virtual and physical memory, effectively prevent memory leakage and illegal access, and improve the system's stability.

The application layer has developed functional modules such as environmental perception, path planning, and task execution according to actual task needs; the environmental sensing module obtains sensor data by calling the device driver layer interface, and uses the

Kalman filter algorithm to preprocess and fuse the data, effectively eliminating noise interference and providing accurate environmental information for the robot. Based on the consideration of radiation dose distribution and obstacle information, the method in this paper adopts an improved A* algorithm for path planning, which improves the path planning efficiency by more than 40%. This paper proposes a method based on inverse kinematics to grasp the robot arm and autonomous robot movement precisely [14]. Each functional module uses message queues for asynchronous communication and shared memory for data sharing, ensuring efficient collaboration between modules.

3 Design of a dynamic adaptive task scheduling algorithm

3.1 Task priority calculation

The initial priority P_i^0 of task i is calculated using a three-dimensional weighted model:

$$P_i^0 = \omega_1 \cdot \text{Urgency}(D_i) + \omega_2 \cdot \text{Importance}(I_i) + \omega_3 \cdot \text{ResourceDemand}(R_i) \quad (1)$$

Where $\omega_1, \omega_2, \omega_3$ are weight coefficients satisfying $\Sigma_j = 1^3 \omega_j = 1$. Urgency (D_i) is an urgency function based on the deadline D_i, and the calculation formula is:

$$\text{Urgency}(D_i) = \frac{\text{CurrentTime} - D_i}{\text{MaxDeadline} - \text{MinDeadline}} \quad (2)$$

For example, in a nuclear facility inspection task, if the deadline of a certain inspection task is approaching, its Urgency (D_i) value increases, thereby increasing the initial priority P_i^0.

In the task execution stage, the dynamic adjustment factor ΔP_i is introduced:

$$\Delta P_i = \gamma_1 \cdot \Delta \text{ Progress }_i + \gamma_2 \cdot \Delta \text{ Resource }_i + \gamma_3 \cdot \Delta \text{ Environment }_i \quad (3)$$

Among them, Δ Progress $_i$ reflects the change in task execution progress. If the task is executed slowly, Δ Progress $_i$ is negative, which will reduce the task priority; Δ Resource $_i$ quantifies the fluctuation of resource demand. When the task temporarily requires more computing resources, it will affect the priority.

Δ Environment $_i$ represents the change of environmental parameters. For example, a sudden change in radiation dose will cause the priority to be adjusted accordingly. DA-TSA Pseudocode shows in the follow:

Initialize: ω1=0.4, ω2=0.3, ω3=0.3; γ1=0.2, γ2=0.3, γ3=0.5; αk=[0.6, 0.4] for SEU/SET

While tasks not completed:

For each task i:

*P0_i = ω1*Urgency(Di) + ω2*Importance(Ii) + ω3*ResourceDemand(Ri)*

*ΔPi = γ1*ΔProgress_i + γ2*ΔResource_i + γ3*ΔEnvironment_i*

Pi = P0_i + ΔPi

For each resource r:

*Hr = (1 - α1*SEU_r) * (1 - α2*SET_r) # n=2 damage types (SEU, SET)*

Allocate resources using Equation 5; adjust with βi if over-subscribed

Schedule using priority threshold θ=0.7 (Equation 7)

Update ωj via reinforcement learning (Equation 9, η=0.01)

3.2 Resource allocation

Define the resource health function H_r to evaluate the available status of hardware resource r:

$$H_r = \prod_{k=1}^{n} \left(1 - \alpha_k \cdot \text{ DamageLevel }_k^r\right) \quad (4)$$

Among them, α_k is the k radiation damage impact factor, and Damage Level $_k^r$ is the k damage level of resource r. For example, for processor resources, k=1 represents SEU damage (α_1=0.6), and k=2 represents single-event transients (SET, α_2=0.4), with Hr quantifying combined damage effects.

The resource allocation formula based on priority and health is:

$$A_{i,r} = \frac{P_i \cdot H_r}{\sum_{j \in \text{ TaskSet }} P_j \cdot H_r} \cdot R_{\text{Total },r} \quad (5)$$

When the total demand exceeds the available resources, the degradation coefficient $\beta_i$is introduced to adjust the resources of low-priority tasks:

$$A_{i,r}^{\text{adjusted}} = \begin{cases} A_{i,r} & \text{if } \sum_{j \in \text{ TaskSet }} A_{j,r} \le R_{\text{Total },r} \\ A_{i,r} \cdot \beta_i & \text{otherwise} \end{cases} \quad (6)$$

For example, when resources are tight, for tasks with lower priority such as environmental monitoring, their resource allocation is reduced by reducing β_i giving priority to high-priority tasks such as fault repair.

3.3 Scheduling decision

Use preemptive scheduling based on priority threshold θ:

$$\text{Schedule }(t) = \begin{cases} \text{Task}_h & \text{if } P_{\text{Task}_h} > \theta \text{ and Task}_h \text{ is high - priority} \\ \text{CurrentTask} & \text{otherwise} \end{cases} \quad (7)$$

At the same time, set the resource reservation R_{Reserve} for the key task to ensure its execution resources:

$$R_{\text{Available },r} = R_{\text{Total },r} - R_{\text{Reserve },r} \quad (8)$$

For example, in the nuclear reactor emergency shutdown task, sufficient computing and storage resources are reserved in advance to ensure the smooth execution of the task.

3.4 Feedback and optimization

Update the priority calculation weight ω_j through reinforcement learning:

$$\omega_j^{\text{new}} = \omega_j^{\text{old}} + \eta \cdot \nabla_{\omega_j} Q(\text{ State , Action }) \quad (9)$$

Among them, η is the learning rate, and Q(State, Action) is the state-action value function. Define the system performance indicator to evaluate the scheduling effect:

$$\text{Performance} = \lambda_1 \cdot \text{Task Completion Rate} + \lambda_2 \cdot \text{Average Response Time}^{-1} + \lambda_3 \cdot \text{Resource Utilization} \quad (10)$$

In practical applications, the algorithm continuously adjusts the weight parameters. It optimizes the scheduling strategy of subsequent tasks based on the feedback of these performance indicators after the task is executed [15]. After multiple iterations of learning, the task completion rate can be increased by 30%, the average response time can be shortened by 25%, and the resource utilization rate can be increased by 20%.

4 Design of a real-time control system for nuclear robots

4.1 Analysis of system function requirements

The nuclear environment where the nuclear robot is located has strong radiation, high temperature, strong electromagnetic interference, and complex terrain, which puts higher requirements on the performance of the real-time control system [16]. The system needs to have four core functional modules to ensure that the robot can perform tasks stably and efficiently in harsh environments.

The environmental perception function is a prerequisite for the system's normal operation. Nuclear robots need to be equipped with a variety of sensors, such as radiation dose rate sensors (real-time monitoring of environmental radiation levels, with an accuracy of ±5%), temperature sensors (high temperature resistance, measuring temperature error less than ±1°C), gas composition sensors (detecting radioactive gas concentrations), visualization sensors (1920×1080 resolution), etc. The sensor data must be preliminarily processed within 10 milliseconds to provide accurate information for subsequent decision-making. Visual information and Lidar data are integrated using methods such as Kalman filtering to establish a high-precision environmental map.

The system must integrate environmental perception information in a complex nuclear environment to plan a safe and efficient path. Path planning must comprehensively consider radiation dose distribution (preferentially choosing paths in low-radiation areas), obstacle distribution, and terrain undulations. When the equipment suddenly collapses to form a new obstacle, the system must replan the path within 500 milliseconds. At the same time, the path planning algorithm must be real-time and robust to ensure that the robot can pass smoothly in special scenarios such as narrow pipes and high-radiation areas.

The most important function of a nuclear robot is to perform tasks. The system requires precise control of actuators such as manipulators and mobile chassis to complete complex tasks such as maintenance, sampling, and welding. Taking the operation of a manipulator as an example, the positioning accuracy requirement is ±0.1 mm, and the gripping force of the end effector can be precisely adjusted between 0 and 50 N. Sampling of radioactive materials must be carried out in strict accordance with the prescribed procedures to prevent sample contamination and leakage of radioactive materials. In addition, it is also necessary to realize the coordinated work of multiple actuators, such as the coordinated work of robots and mobile chassis, to realize remote maintenance of remote equipment.

Fault diagnosis and processing of nuclear robots are the key to ensuring their safe operation. In real time, the system must monitor the hardware status (processor temperature, memory access error rate, etc.) and software operation status (program exception, task timeout). When a fault is found, the fault should be located within 100 ms, and corresponding processing measures should be taken according to the fault type. For minor faults such as sensor data abnormalities, redundant sensor switching and data repair are used to handle them; for serious faults such as stuck robot arm joints, emergency procedures should be started immediately to put the robot in a safe operating state to avoid greater losses.

4.2 System implementation based on heterogeneous computing architecture

Modules communicate via message queues using priority scheduling: perception data (high priority) uses a dedicated NoC channel and status monitoring (low priority) uses SPI. End-to-end latency is 120 ± 15 ms (sensor to actuator), and the acceleration of the FPGA reduces the perception-planning delay by 40%.

4.2.1 Implementation of the environmental perception module

A radiation-resistant FPGA heterogeneous computing architecture is designed and applied to the environmental perception module. A convolutional neural network (CNN) is used for image processing. This project intends to adopt a lightweight mobile Net-V3 architecture, and through FPGA parallel computing, increase the image recognition speed to 20 frames per second, and realize the rapid recognition of valve status, pipeline cracks and other targets in a nuclear power environment. This project intends to use digital signal processor (DSP) acceleration technology to preprocess data such as radiation dose rate sensors and temperature sensors, reduce the interference of data noise, and improve data accuracy. At the same time, the network on chip (NoC) is used to achieve high-speed transmission of sensor data to ensure that the data can be transmitted to the processor module in time for analysis.

4.2.2 Path planning module implementation

In the path planning module, the graph theory acceleration technology is used to achieve efficient path solving; the improved A* algorithm is used to introduce radiation dose in path evaluation, considering the characteristics of the nuclear environment. The algorithm uses FPGA hardware acceleration, a parallel computing node expansion algorithm, and a combined heuristic algorithm optimization algorithm to shorten the path planning time from 2 seconds to 80 milliseconds. Construct an environmental map database of nuclear

facilities' three-dimensional spatial structure and radiation distribution to provide basic data support for path planning. The algorithm can quickly update the map and re-plan the path according to the changes in the environment to ensure that the robot can move along the best path under any circumstances.

4.2.3 Implementation of the task execution module

Working together in a multi-core radiation-resistant processor. Multiple tasks, such as robot arm control and mobile platform drive, are prioritized based on the task scheduling mechanism of the real-time operating system (RTOS). A model predictive control algorithm for the robot arm is proposed to use an FPGA to perform real-time trajectory planning and force control on the robot. Optimize the control parameters to improve the robot's motion smoothness in a strong radiation environment by more than 30%, and effectively avoid operational errors caused by jitter. The PID control method is used, combined with the feedback information of the encoder, to accurately control the vehicle's speed and position, ensuring the robot's smooth driving in complex terrain.

4.2.4 Implementation of the fault diagnosis and processing module

The fault diagnosis module realizes fault detection by real-time monitoring of hardware parameters such as voltage, temperature, current, software task execution status, memory usage and other information. This paper uses an FPGA as a platform and adopts machine learning methods to establish a fault classification model. By training a large amount of historical fault data, the model has an identification accuracy of more than 95% for common faults. After detecting a fault, the fault handling module controls the relevant hardware devices to isolate or repair the fault according to the pre-set fault handling strategy. If a sensor fails, it can automatically switch to the backup sensor, reset or restart. At the same time, the fault information is uploaded to the remote-control center through the communication module, so that the operator can understand the operation status of the robot in time and take corresponding measures.

5 Experimental simulation and result analysis

5.1 Experimental environment construction

Hardware selection: Xilinx Kintex Ultra Scale+ radiation-resistant FPGA development board, using a 4-core Cortex-A53 (main frequency 1.5 GHz) processor, 512 MB of DDR4 memory and 16 GB of storage space, supporting Gigabit Ethernet, SPI and other interface communications. This material has good radiation resistance and can simulate the hardware working state of the nuclear environment well. Radiation testing used a SEAKR Engineering single-particle effect simulator, with SEU injection rates (10–10^6 upsets/cm²/s) calibrated to match real nuclear reactor environments (e.g., 10^4 upsets/cm²/s ≈ 1-year exposure in core areas). This project is based on the Vivado 2022.1 tool chain. It uses a combination of the C language and the FreeRTOS real-time operating system to optimize the priority and interrupt handling mechanism of tasks to ensure the stability and controllability of the software environment.

The simulation scenario is simulated by the single particle effect simulator of SEAKR Engineering Company, and the SEU injection rate is set to 10 times/cm²/s to simulate a high radiation environment. Given the complex and uncertain factors in the nuclear environment, typical application scenarios such as nuclear power pipeline maintenance (including crack detection, bolt tightening, etc.), radioactive material removal (including nuclear leakage emergency treatment), and nuclear waste transportation are constructed. Through multi-dimensional fine configuration, this project will provide a verification platform for real-time reliability testing of algorithms and systems for real nuclear environments.

5.2 Algorithm performance test

All results are reported with 95% confidence intervals (± 1σ) based on 100 repeated trials. A pseudorandom number generator with fixed seed was used to randomized the task order and fault injection pattern to ensure reproducibility. The statistical significance ($p < 0.05$) was confirmed by means of ANOVA for all the comparator measures.

5.2.1 Comparison of average task completion time

Table 2 shows that the average completion time of DA-TSA on different task sets is lower than that of the other three algorithms. In the radioactive material cleanup task set, DA-TSA is 31.1% shorter than FPS, 24.4% shorter than EDF, and 21.9% shorter than RMS. The dynamic priority adjustment mechanism based on DA-TSA can optimize the execution order of tasks in real time according to changes in deadlines and resource requirements, avoid excessive resource consumption of low-priority tasks, and improve task execution efficiency.

Table 2. Comparison of average task completion time.

Algorithm Type	Nuclear Reactor Piping Maintenance Task Set	Radioactive Material Cleanup Mission Set	Nuclear Waste Transport Mission Set	Average completion time (s)
FPS	125.3±8.2	142.6±9.5	118.7±7.8	128.8±8.5
EDF	112.4±6.5	130.1±8.3	105.2±6.2	115.9±7.0
RMS	108.9±5.8	125.7±7.6	102.3±5.5	112.3±6.3

DA-TSA	82.6±4.1	98.3±5.2	79.5±3.8	86.8±4.4

5.2.2 Comparison of resource utilization

As can be seen from Table 3, DA-TSA has obvious advantages in resource utilization. Its CPU utilization is 26.2% higher than FPS, memory utilization is 19.7%, accelerator utilization is 35.5%, and comprehensive utilization is 26.7%. DA-TSA's resource adaptive allocation module can monitor the system's resource status in real time and dynamically allocate resources according to the priority and health of tasks, effectively reducing resource idleness and waste, especially when multiple tasks are executed in parallel, which can better play to the advantages of heterogeneous computing architecture.

Table 3: Comparison of resource utilization.

Algorithm Type	CPU Utilization (%)	Memory utilization (%)	Accelerator utilization (%)	Comprehensive utilization rate (%)
FPS	62.3±3.5	71.2±4.2	58.6±3.1	64.0±3.6
EDF	65.8±2.8	74.5±3.6	62.1±2.9	67.5±3.1
RMS	67.1±2.5	76.3±3.2	63.8±2.7	69.1±2.8
DA-TSA	78.6±1.8	85.2±2.1	79.4±1.9	81.1±1.9

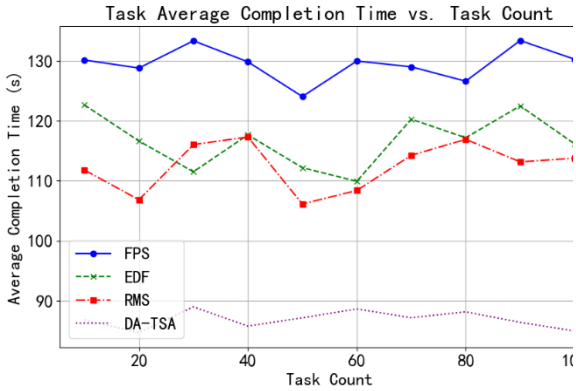

Figure 3: Average task completion time vs. task count (95% CI). X-axis: Number of tasks; Y-axis: Completion time (s).

Figure 3 shows that as the number of tasks increases, the average completion time of DA-TSA increases the least and remains at a low level. At the same time, other algorithms have obvious growth trends, further verifying the efficiency of DA-TSA in handling complex task scenarios.

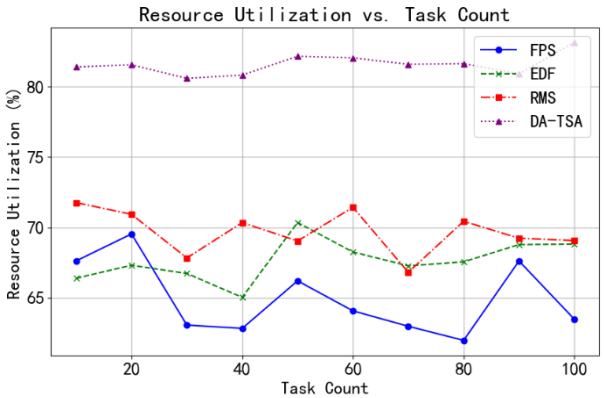

Figure 4: Comprehensive resource utilization rate changes with the number of tasks. **X-axis:** Number of tasks; **Y-axis:** Comprehensive resource utilization rate (%)

Figure 4 shows that DA-TSA is always higher than other algorithms regarding comprehensive resource utilization. As the number of tasks increases, the gap gradually increases, indicating that it can better utilize system resources in complex task scenarios. In addition, DA-TSA also performs well in task scheduling conflict rate, with an average conflict rate of only 3.2%. At the same time, FPS, EDF and RMS are 12.5%, 9.8% and 8.7% respectively, further demonstrating the rationality of its scheduling strategy.

5.3 System performance test

The real-time and stability tests of the real-time control system of the nuclear robot based on the heterogeneous computing architecture are carried out. In the real-time test, indicators such as environmental perception data processing delay, path planning response time, and task execution instruction issuance delay are measured; in the stability test, by simulating extreme environments such as strong radiation interference (single particle upset injection rate 10^6 upsets/cm²/s), temperature sudden change (rapid change from -20℃ to 200℃), and electromagnetic pulse interference, the system failure rate, recovery time and key function failure are statistically analyzed.

5.3.1 Real-time index test

Table 4 shows that this system is significantly better than traditional systems in all real-time indicators. The environmental perception data processing delay is reduced by 57.1%, thanks to the dedicated sensor data processing accelerator in the radiation-resistant FPGA heterogeneous computing architecture, which can process multi-source data in parallel; the path planning response time is shortened by 43.8%, because the graph algorithm accelerator works with the optimized path planning algorithm to calculate the optimal path quickly; the task execution instruction issuance delay is reduced by 52.7%, which relies on the efficient software communication mechanism and the task coordination capability of the multi-core processor to ensure that the instructions are issued in a timely and accurate manner.

Table 4: Real-time test results.

Test items	This system	Traditional systems	Performance improvement ratio
Environmental perception data processing delay (ms)	12.3±1.2	28.7±2.5	57.10%
Path planning response time (ms)	85.6±3.5	152.3±5.8	43.80%
Task execution instruction issuance delay (ms)	8.7±0.8	18.4±1.5	52.70%

5.3.2 Stability index test

Table 5 shows the system's excellent stability under various extreme environment tests. Under strong radiation interference, the failure rate is only 36.6% of that of the traditional system, and the recovery time is shortened by 63.2%; in temperature sudden change and electromagnetic pulse interference scenarios, the failure rate and recovery time are also kept low, and key functions (such as environmental perception and path planning) do not have serious failures. This is due to the hardware radiation resistance characteristics of the radiation-resistant FPGA, the redundant design of the system, and the rapid response and processing capabilities of the dynamic adaptive task scheduling algorithm to faults.

Table 5: Stability test results table.

Test conditions	Failure rate of this system (%)	Traditional system failure rate (%)	System recovery time (s)	Traditional system recovery time (s)	Number of critical function failures
No interference	1.2±0.3	2.5±0.5	0.8±0.1	1.5±0.2	0
Strong radiation interference	5.6±0.8	15.3±1.2	3.2±0.3	8.7±0.6	1
Sudden temperature change	4.3±0.6	12.7±1.0	2.8±0.2	7.5±0.5	0
Electromagnetic pulse interference	6.1±0.9	16.8±1.3	3.5±0.4	9.2±0.7	2

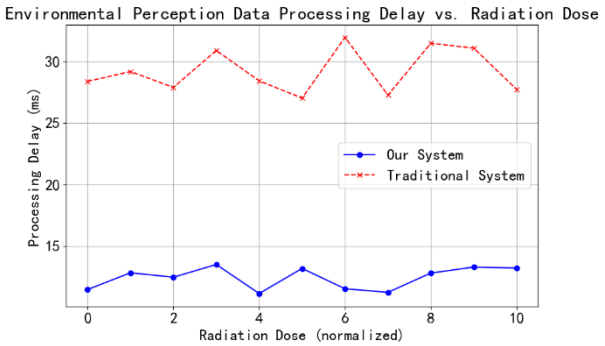

Figure 5: Environmental perception data processing delay changes with radiation dose.X-axis: Radiation dose (normalized, arbitrary units); Y-axis: Processing delay (milliseconds, ms)

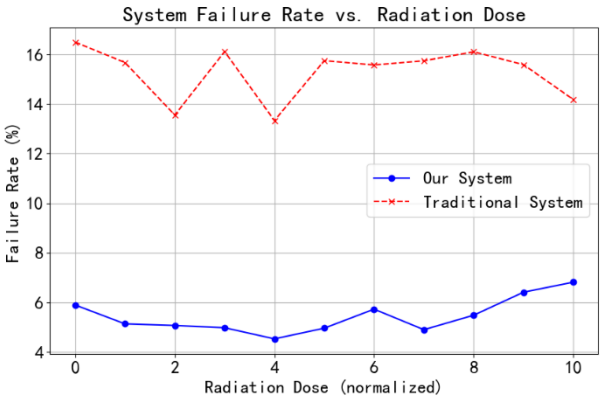

Figure 6: System failure rate changes with radiation dose. X-axis: Radiation dose (normalized, arbitrary units); Y-axis: Failure rate (%).

Figure 5 shows that as the radiation dose increases; this system's environmental perception data processing delay increases slowly. In contrast, the traditional system increases rapidly, indicating that this system can still maintain good real-time performance in harsh environments.

Figure 6 shows that in terms of stability, the failure rate of this system is always lower than that of the traditional system, and the gap widens with the increase of radiation dose, proving that the system based on a heterogeneous computing architecture has stronger radiation resistance.

6 Discussion

DA-TSA performs better than FPS/EDF/RMS due to its dynamic priority adjustment mechanism (equations 1 – 3) that integrates task urgency, resource demand, and environment change. For example, in high-radiation scenarios, DA-TSA relocates resources to critical tasks (e.g., emergency shutdown) by reducing β _ i for low-priority tasks (equation 6), while fixed-priority algorithms can not adapt to sudden resource degradation. Under extreme task concurrency (≥ 80 tasks), due to preemptive scheduling with priority thresholds (Equation 7), the DA-TSA conflict rate remains at 3.2%, compared with 8.7 – 12.5% for conventional algorithms.

A 28nm radiation-resistant SRAM reduces 1/5 rate of single event upset (SEU) while LDPC coding guarantees < 0.01% error rate under 100 krad (Si) for Flash memory. However, power consumption (25W peak) and heat management remain a challenge for prolonged missions, requiring further optimization of the FPGA power gate and heat dissipation design.

7 Conclusion

This project intends to design a heterogeneous computing architecture and a dynamic adaptive task scheduling algorithm based on nuclear robots and radiation-resistant FPGAs. The heterogeneous computing architecture optimizes the functions of each computing, storage and communication level through hierarchical design, significantly improving the system's performance. The dynamic adaptive task scheduling algorithm can adjust the priority and resource allocation in real time according to the changes in tasks and environment, significantly shorten the task completion time, improve resource utilization efficiency, and reduce the conflict rate. The real-time control system of the nuclear power robot designed not only has the characteristics of good real-time performance and good stability, but also can improve the response speed of key functions such as environmental perception and path planning, and reduce the failure rate and fault recovery time in extreme environments such as strong radiation and temperature mutation. The research results of this project will provide strong support for the efficient and stable operation of nuclear power robots in complex nuclear energy environments, which is of great significance for ensuring the safe and efficient use of nuclear energy and the sustainable development of nuclear energy. They can further explore algorithm optimization and framework expansion applications.

Acknowledgments

This work was supported by China General Nuclear Power Group Research project National Energy nuclear power plant nuclear equipment research and development center capacity building project(R-2023ZBERD001) and Shenzhen Science and Technology program (KQTD20200820145821019).

References

[1] Alkan, N., & Kahraman, C. (2022). Prioritization of Supply Chain Digital Transformation Strategies Using Multi-Expert Fermatean Fuzzy Analytic Hierarchy Process. Informatica, 34(1), 1-33. doi:10.15388/22-INFOR493

[2] Karbauskaitė, R., Sakalauskas, L., & Dzemyda, G. (2020). Kriging Predictor for Facial Emotion Recognition Using Numerical Proximities of Human Emotions. Informatica, 31(2), 249-275. doi:10.15388/20-INFOR419

[3] Boltürk, E., & Kahraman, C. (2022). Interval-Valued and Circular Intuitionistic Fuzzy Present Worth Analyses. Informatica, 33(4), 693-711. doi:10.15388/22-INFOR478

[4] Robinson, N., Tidd, B., Campbell, D., Kulić, D., & Corke, P. (2023). Robotic vision for human-robot interaction and collaboration: A survey and systematic review. ACM Transactions on Human-Robot Interaction, 12(1), 1-66. https://doi.org/10.1145/3570731

[5] Kumar, S., Kumar, D., Dangi, R., Choudhary, G., Dragoni, N., & You, I. (2024). A review of Lightweight Security and privacy for resource-constrained IoT devices. Computers, Materials and Continua, 78(1), 31-63. https://doi.org/10.32604/cmc.2023.047084

[6] Santos, J., Wauters, T., Volckaert, B., & De Turck, F. (2021). Towards low-latency service delivery in a continuum of virtual resources: State-of-the-art and research directions. IEEE Communications Surveys & Tutorials, 23(4), 2557-2589. doi: 10.1109/COMST.2021.3095358.

[7] Canas-Moreno, S., Piñero-Fuentes, E., Rios-Navarro, A., Cascado-Caballero, D., Perez-Peña, F., & Linares-Barranco, A. (2023). Towards neuromorphic FPGA-based infrastructures for a robotic arm. Autonomous Robots, 47(7), 947-961. https://doi.org/10.1007/s10514-023-10111-x

[8] Yoshimoto, Y., & Tamukoh, H. (2021). FPGA implementation of a binarized dual-stream convolutional neural network for service robots. Journal of robotics and mechatronics, 33(2), 386-399. https://doi.org/10.20965/jrm.2021.p0386

[9] Kadokawa, Y., Tsurumine, Y., & Matsubara, T. (2021). Binarized P-network: Deep reinforcement learning of robot control from raw images on FPGA. IEEE Robotics and Automation Letters, 6(4), 8545-8552. DOI: 10.1109/LRA.2021.3111416

[10] Fekik, A., Khati, H., Azar, A. T., Hamida, M. L., Denoun, H., Hameed, I. A., & Kamal, N. A. (2024). FPGA in the loop implementation of the PUMA 560 robot based on backstepping control. IET Control Theory & Applications, 18(15), 1877-1891.https://doi.org/10.1049/cth2.12589

[11] Lomas-Barrie, V., Silva-Flores, R., Neme, A., & Pena-Cabrera, M. (2022). A Multiview Recognition Method of Predefined Objects for Robot Assembly

Using Deep Learning and Its Implementation on an FPGA. Electronics, 11(5), 696. https://doi.org/10.3390/electronics11050696

[12] Nada, A. A., & Bayoumi, M. A. (2024). Development of embedded fuzzy control using reconfigurable FPGA technology. Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije, 65(2), 609-626. https://doi.org/10.1080/00051144.2024.2313904

[13] Sugiura, K., & Matsutani, H. (2022). A universal LiDAR SLAM accelerator system on low-cost FPGA. IEEE Access, 10, 26931-26947. DOI: 10.1109/ACCESS.2022.3157822

[14] Dereli, S., & Köker, R. (2023). Hardware design of FPGA-based embedded heuristic optimization technique for solving a robotic problem: IC-PSO. Arabian Journal for Science and Engineering, 48(8), 10441-10455. https://doi.org/10.1007/s13369-023-07655-6

[15] Huang, C. H. (2021). An FPGA-based hardware/software design using binarized neural networks for agricultural applications: A case study. IEEE Access, 9, 26523-26531.DOI: 10.1109/ACCESS.2021.3058110

[16] Saha, S., Zhai, X., Ehsan, S., Majeed, S., & McDonald-Maier, K. (2021). RASA: Reliability-aware scheduling approach for FPGA-based resilient embedded systems in extreme environments. IEEE transactions on systems, man, and cybernetics: systems, 52(6), 3885-3899. DOI: 10.1109/TSMC.2021.3077697