Real Time Dynamic Control and Optimization of ARM Based Robotic System Using System-Verilog

[1]G. Deepika Bharathi ,[2]K. Bhahusha,[3]K. Sravani,[4]G. Meghana,[5]G. Monika, [6]P. Anil Kumar

[1,2,3,4,5]U. G Student, Dept ELECTRONICS AND COMMUNICATION ENGINEERING,
St.Ann's College of Engineering and Technology (Autonomous), Chirala, Bapatla Dist,
Andhra Pradesh – 523187, India
[6]Assistant Professor, Dept ELECTRONICS AND COMMUNICATION ENGINEERING,
St.Ann's College of Engineering and Technology (Autonomous), Chirala, Bapatla Dist,
Andhra Pradesh – 523187, India

ABSTRACT

The integration of embedded processors, reconfigurable hardware, and wireless communication technologies has led to the development of advanced robotic systems with improved flexibility, reliability, and real-time control. This project presents the design and implementation of an ARM-based robotic system whose control logic is developed in Verilog and functionally verified using SystemVerilog. The system incorporates real-time dynamic optimization algorithms to adapt to environmental changes and task variations, ensuring efficient performance. A Bluetooth communication module is integrated for wireless control, monitoring, and feedback, enabling human–machine interaction through portable devices such as smartphones and tablets. The hardware-centric design ensures low latency, high-speed decision-making, and deterministic responses, while verification through SystemVerilog ensures correctness, robustness, and reliability. The proposed robotic system demonstrates the synergy of hardware optimization, embedded control, and wireless communication for next-generation robotic applications.

Keywords: ***ARM-Based Processing, Hardware Description Language (HDL), QuestaSim Verification, HC-05 Bluetooth Module, Real-Time Dynamic Control, System-Verilog Modeling, Low-Latency Hardware Design***

INTRODUCTION

Robotics has become an essential part of modern automation, contributing to industrial, medical, and domestic applications, with robotic arms widely used for tasks such as assembly, pick-and-place operations, packaging, and material handling. Traditional robotic arm control systems often rely on microcontrollers, which limit flexibility and computational efficiency, leading researchers to adopt advanced design approaches using hardware description languages like System-Verilog. System-Verilog offers high-level abstraction, modularity, and powerful verification methods, making it well-suited for modeling complex robotic systems. The proposed work focuses on an ARM-based robotic platform for real-time dynamic control and optimization, integrating IR sensors, Bluetooth communication, Arduino, and motor drivers.

The robotic arm interacts with its environment in real time, enhancing autonomy and adaptability, while the HC-05 Bluetooth module enables wireless communication with a mobile application that provides intuitive controls and real-time feedback. Equipped with IR sensors, the system can detect objects and perform autonomous decision-making for tasks like sorting, picking, and placing. Unlike conventional microcontroller-based designs, the ARM platform supports faster data handling, improved signal processing, and reduced latency, while motor drivers such as L293D ensure precise control of robotic and arm movements. The control logic is developed and verified in System-Verilog, with testbenches validating accuracy, timing, and performance. This HDL-based design enables higher performance compared to traditional software-driven systems, with simulations and verification carried out in QuestaSim to ensure reliability prior to real-time deployment.

The closed-loop feedback system enhances responsiveness and reduces errors, while optimization techniques improve efficiency by minimizing power consumption and mechanical stress. Dynamic control strategies allow the robotic arm to adapt to environmental changes, ensuring smooth motor operation and preventing irregular movement. Integration with a mobile application provides flexibility for remote monitoring and control, making the system more user-friendly. The design significantly improves accuracy, reliability, and scalability compared to conventional robotic arms, making it suitable for both industrial and domestic applications. Its modular architecture also supports future enhancements such as additional sensors or machine learning integration. By combining System-Verilog-based modeling, ARM processing, mobile application control, and IR sensor-based object detection, the system demonstrates advanced flexibility, speed, and autonomy.

Ultimately, this ARM-based robotic system serves as a next-generation model for efficient, intelligent, and autonomous robotic arm applications, while providing a strong foundation for academic research and future innovation in real-time robotics and HDL-based design.

LITERATURE SURVEY

Over the last decade, robotic arm control systems have evolved to meet the growing demands of industrial and domestic automation. Early microcontroller-based and wired systems lacked flexibility, prompting a shift toward wireless Bluetooth interfaces (Kumar et al.) and sensor-integrated autonomous designs (Singh et al.). However, traditional microcontroller architectures often struggled with computational latency and limited processing power during real-time tasks.

To address these performance bottlenecks, FPGA-based hardware-software co-designs were introduced. These systems typically integrate embedded processors (e.g., Nios) for complex inverse kinematics and dedicated logic for high-frequency tasks, such as multi-axis PWM position control and sine/cosine microstepping for stepper motors. While FPGAs achieve high speed and precise motion control, their hardware complexity often limits scalability. Consequently, ARM-based controllers have emerged as a superior alternative, offering lower latency, highly efficient real-time processing, and robust closed-loop feedback handling.

Concurrently, the design validation process has advanced significantly. While Patel highlighted the necessity of HDL simulation for timing verification, modern methodologies have transitioned to SystemVerilog. Supported by rigorous QuestaSim testbenches, SystemVerilog provides the modularity and advanced verification required to ensure hardware reliability before deployment.

EXISTING SYSTEM

Existing robotic arm systems are largely microcontroller-based, which makes them affordable but limits their computational capabilities and flexibility. These systems typically rely on manual control, either through wired connections or simple Bluetooth interfaces, restricting autonomy. Many lack advanced sensing mechanisms, and when IR sensors are used, the processing is too slow for real-time decision-making. Motor control in existing systems often suffers from latency and limited precision due to basic control logic. Closed-loop feedback mechanisms are rarely implemented, leading to errors in object handling and movement. Verification of such systems is limited to basic simulations, which reduces reliability under real-time conditions. Industrial scalability is restricted because microcontrollers cannot handle high computational loads. Hazard prevention is minimal, creating safety concerns in critical applications. Mechanical stress is often higher due to unoptimized movements, resulting in reduced system life.

Mobile app integration, when present, is generally limited and lacks feedback mechanisms. These designs are not energy-efficient and often consume more power. System adaptability to environmental changes is poor, preventing flexible use across multiple tasks. Existing systems also face difficulties in integrating additional sensors or modules due to lack of modularity. Their performance is insufficient for modern industrial automation demands. Reliability under heavy workloads remains questionable. Latency is higher compared to ARM-based designs. Most existing robotic arms cannot be easily reprogrammed for new applications. These drawbacks highlight the need for optimized, autonomous, and flexible robotic systems. Hence, the present ARM and System-Verilog-based design emerges as a strong alternative to overcome the limitations of current systems.

The Robotic Arm is designed using the switch controller, this process works on the principle of interfacing servos and spur gears. This task is achieved by using perforated plates. The battery emits the power source to the motor it rotates the spur to convert to and flow motion. . This servo will respond with regards to the pulses which results in the moment of the arm [6].The servo motor and battery are the main components for the wired controlled robotic arm. The process of complete section and motion of the project by providing efficient power source. For the pick and place the robotic arm is used, usually a dedicated module designed specifically for use with servomotors. The components in short terms, Servo motors, Spur gears, Preforated plates, Perforated plates, Servo motors, Battery, Spur gears, Connecting wires, Switches, Wheels, Grippers, Holders.

The working principle of the wired controlled robotic arm can be clearly defined in the diagram. If the human needs the help of robotic arm the main role is to operate the Robot carefully, by providing comments or the operator should handle carefully, it should also possess the capability of sending specific orders to the manipulator which as to be carried out in terms of positions or velocities of its final effectors. There are three kinds switches are in the control panel, one is for forward and reverse, and the another one is for upward and downward directions, and similarly for holding the objects. These three unit of system are in one end of connections operated by human. Therefore, the implementation of the task is done by using the Robotic arm, The functions of moving and rotating the arm of the project is may be continuous process or whenever it wants may applied for the work. Now these functions are needed to be an integrated with the applications are then used to solve the particular task. The maintenance of the project is low by providing loads or work the capacity of the material is limited to done the process of the task.

By using Robotic arm, it perform the similar activities like welding, gripping, etc., for Eg. According to the type of Robots the system of working process is also changed like assembled the parts of objects. In hospitals, the hazardous wastes are completely handled by Robots. For pick and placed the Robots are highly used by providing cameras the robots accuracy is constant. In case of handling bombs and or explosive materials are proper. The robot arms can be autonomous or controlled manually and can be used to perform a variety of tasks with high accuracy. Shifting of robotic arms may very useful for the change in need of positions. Conformation of material handled by the arm while gripping the material is brittle . Holding device is capable of differentiate the complete level of approach to transform the objects from the positions.

PROPOSED SYSTEM

The proposed methodology focuses on the design and optimization of a real-time ARM-based robotic system that integrates mobile app control, IR sensing, and System-Verilog-based hardware modeling. A mobile application is developed as the primary user interface to provide commands wirelessly to the robotic system using a Bluetooth HC-05 module. These commands are received and processed by the ARM-based controller, which executes control logic modeled in System-Verilog. By adopting System-Verilog, the system ensures modular design, efficient execution, and reliable verification of robotic control tasks.

The robotic arm is equipped with IR sensors for real-time object detection, enabling it to sort, pick, and place objects without direct human intervention. Sensor data is processed dynamically by the ARM processor to support decision-making and autonomous operations. The robotic arm movement is powered by L293D motor drivers, which provide precise control over motor rotation and grip functions. Closed-loop feedback is implemented to minimize errors and ensure smooth movement, while hazard prevention is managed by monitoring sensor feedback in real time. The entire control logic is tested and validated using System-Verilog testbenches, and functional verification is carried out in QuestaSim under various scenarios to confirm robustness. The methodology ensures low latency and near real-time responsiveness, making it suitable for industrial applications. Optimization algorithms are applied to reduce power consumption and mechanical stress, extending system life and improving efficiency.

The design supports hybrid control, allowing both manual operation through the mobile app and autonomous functioning through IR sensors, thereby enhancing flexibility. Scalability is embedded in the design, allowing future integration of additional sensors or robotic modules. Independent testing of modules before system-level integration ensures reliability. Simulation results validate accuracy, speed, and correctness of the robotic control mechanism, while real-time deployment proves its adaptability to dynamic environments. The modular HDL-based approach allows reprogramming and design modification with ease. Real-time feedback through the mobile app provides users with system updates, ensuring transparency and confidence in operation. The proposed system ultimately emphasizes accuracy, speed, autonomy, and scalability, making it a robust and future-ready robotic solution.

ARCHITECTURE

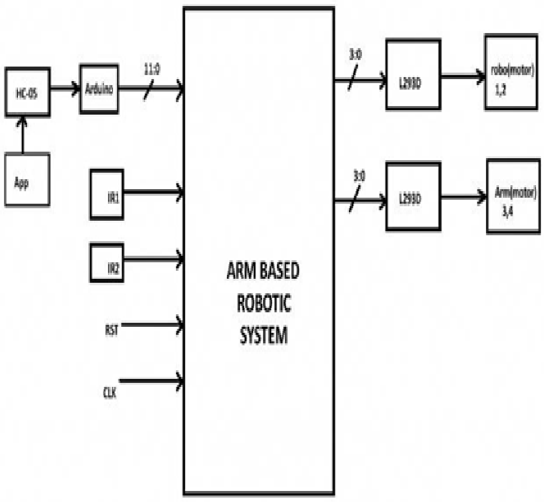

Fig :1 Block Diagram

The architecture of the proposed real-time dynamic ARM-based robotic system is designed to achieve efficient, flexible, and adaptive control of robotic operations. The system is composed of multiple functional blocks including input interface, ARM-based controller, communication module, motor driver unit, robotic arm, and feedback system. At the input level, commands are provided either through a mobile application or sensor inputs. The mobile application communicates wirelessly with the system using a Bluetooth HC-05 module, enabling real-time user interaction and control. The received commands are forwarded to the ARM-based controller, which acts as the central processing unit of the system. The ARM controller processes input data and executes control logic modeled using

System-Verilog. This hardware-oriented design ensures high-speed processing, reduced latency, and deterministic behavior. The controller generates appropriate control signals based on user commands and sensor feedback.

IR sensors are integrated into the system to detect objects and environmental conditions. These sensors continuously provide real-time data to the ARM controller, enabling dynamic decision-making. Based on this feedback, the controller adjusts the movement and operation of the robotic arm. The motor driver unit, implemented using L293D drivers, converts control signals into appropriate power signals to drive the motors. This ensures precise control of motor rotation, direction, and gripping actions of the robotic arm. A closed-loop feedback mechanism is incorporated to improve system accuracy and reliability. Sensor data is continuously monitored, and corrective actions are taken to minimize errors and ensure smooth operation. This feedback system also enhances safety by detecting abnormal conditions. The modular architecture allows easy integration of additional sensors and functional modules, improving scalability. Overall, the architecture supports real-time processing, dynamic adaptability, and efficient robotic control, making it suitable for advanced automation applications.

MEDHOLOGY DISCRIPTION

A. Data Collection

The proposed system continuously collects real-time data from multiple sources, including user inputs and IR sensors. The mobile application provides control commands through a Bluetooth communication module, enabling wireless interaction with the robotic system. IR sensors are deployed to detect objects and environmental conditions in the working area. These sensors provide continuous feedback regarding object presence, position, and movement. The collected data is forwarded to the ARM-based controller for further processing.

B. Data Pre-processing

The acquired data is processed to ensure accuracy and reliability before execution. Sensor signals are filtered to remove noise and unwanted disturbances. This step ensures stable and precise detection of objects. The input commands from the mobile application are also validated and formatted properly before being executed. Pre-processing helps in reducing errors and ensures smooth system operation under varying environmental conditions.

C. Data Processing and Analysis

The processed data is analyzed by the ARM-based controller, which executes control logic developed using System-Verilog. The controller compares real-time sensor data with predefined conditions to make intelligent decisions. Based on the analysis, the system determines the appropriate action such as moving the robotic arm, picking or placing objects, or stopping operation. This enables dynamic adaptation to changing conditions.

D. Output and Actuation

Based on the processed data, control signals are generated and sent to the motor driver unit. The motor driver controls the movement of the robotic arm by driving motors with required speed and direction. The robotic arm performs actions such as object detection, gripping, lifting, and placing based on these signals. Simultaneously, feedback is continuously monitored to ensure accurate execution. The system also provides real-time updates to the user through the mobile application, enabling monitoring and control. In case of abnormal conditions, the system can take corrective actions automatically, ensuring safety and reliability.

SOFTWARE REQUIREMENTS

Software Tools:

Questa Sim is a powerful simulation and functional verification tool used for designing and testing digital circuits using languages like Verilog and VHDL. Developed by Siemens EDA, it helps detect errors early through simulation before hardware implementation. It also provides debugging, waveform analysis, and coverage features to ensure accurate and reliable system performance.

Languages:

Verilog – Hardware design and implementation

System Verilog – Verification, testbench, and simulation

RESULTS AND DISCUSSION

Rand Testcase:-

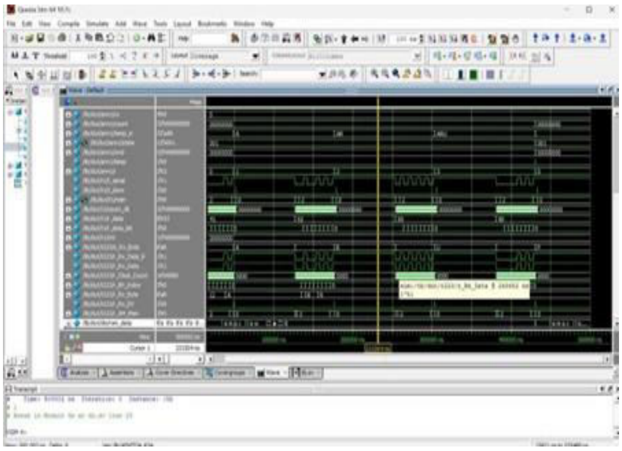

Fig :2 Rand Test case

This rand case is used to obtain any of the movements so that robot gets some certain action. The 6 servo motors we fixed to mechanical system will work.

Forward Testcase:-

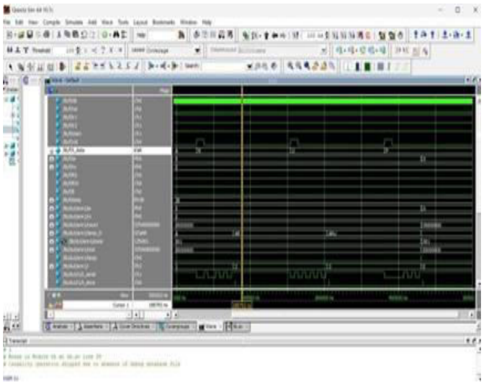

Fig: 3 Forward Testcase

The four motors at the bottom of mechanical system of robot should be rotate in clockwise direction to move in forward direction.

Left Testcase:-

The four motors will be in square format at the bottom of system. The two motors at the left side should rotate in anti-clockwise direction while other two motors will rotate in clockwise direction to obtain the robot

Right Testcase:-

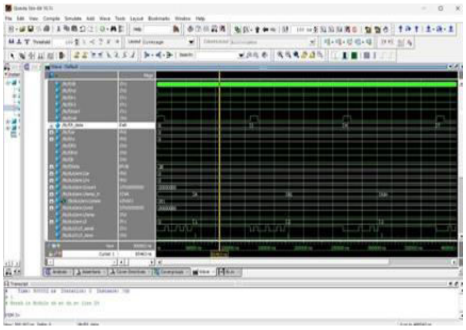

Fig: 4 Right Testcase

The four motors will be in square format at the bottom of system. The two motors at the right side should rotate in anti-clockwise direction while other two motors will rotate in clockwise direction to obtain the robot movement to turn right.

Stop Testcase:-

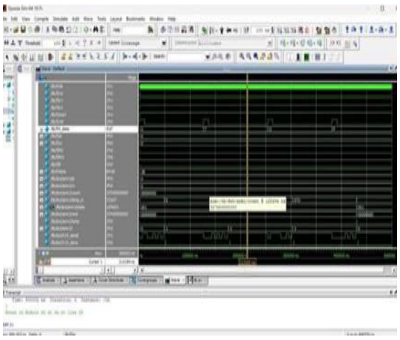

Fig: 5 Stop Testcase

In this case the robot will be in stop position performing no action by turning off all the servo motors. For Example, if robot is in forward direction then, to get it into stop position, the motors in clockwise direction will start rotating in anti-clockwise direction to obtain stop position of the robot.

Pick Testcase:-

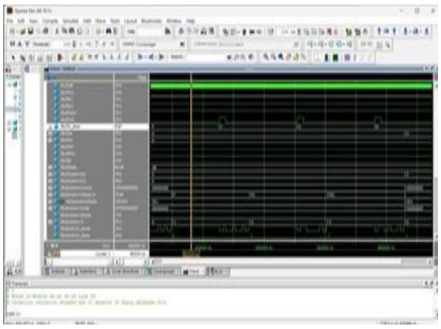

Fig: 6 Pick Testcase

The two motors of hand in which one serves for thumb movement and other for four finger movement. In this case, the thumb motor will rotate in anti- clockwise and other motor in clockwise direction to pick any desired object.

Drop Testcase:-

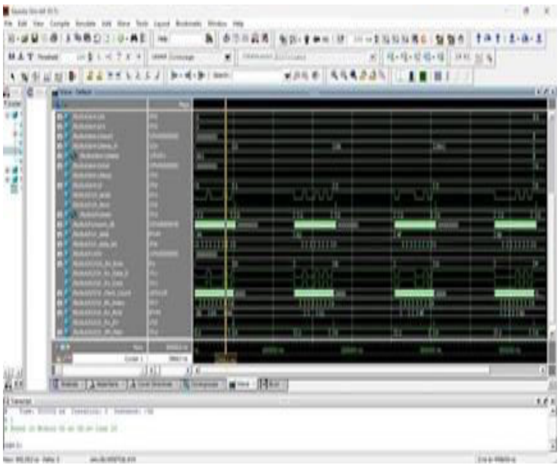

Fig: 7 Drop Testcase

The two motors of hand in which one serves for thumb movement and other for four finger movement. In this case, the thumb motor will rotate in clock wise and other motor in anti-clockwise direction to pick any desired object.

CONCLUSION

For the ARM-based robotic system implemented in Verilog and verified using SystemVerilog, comprehensive code coverage and testcases are essential to validate its functional correctness and robustness. Code coverage is measured across statement, branch, toggle, and FSM coverage to ensure that all control paths, conditional decisions, and hardware states are exercised during simulation. This guarantees that the Verilog control logic for dynamic motion, sensor feedback, and actuator interfacing is fully verified. Functional coverage is defined for key scenarios such as different robotic motion commands, dynamic speed adjustment, obstacle detection response, and Bluetooth communication events, ensuring end-to-end behaviour is captured. The testcases are categorized into directed and constrained-random scenarios: directed tests verify specific functionalities like forward/backward movement, rotation, fault-injection for sensor errors, and Bluetooth command reception, while constrained-random tests explore system robustness under varied environmental conditions, data rates, and fault events. Additionally, regression suites are run to validate power optimization features, parallelized control execution, and recovery from unexpected inputs. Together, these coverage-driven testcases confirm the reliability, performance, and adaptive capabilities of the robotic system.

FUTURE SCOPE

Incorporating machine learning algorithms for adaptive decision-making and autonomous task execution. Extending wireless connectivity beyond Bluetooth to Wi-Fi/5G for cloud based robotic control and monitoring. Integration of fault-tolerant architectures and self-healing circuits for mission-critical robotic systems. Expanding from a single robotic unit to multi-robot collaboration and swarm robotics. Real-time integration of advanced sensors (LIDAR, vision modules, IMU) for intelligent navigation and environmental interaction. Using advanced power management techniques to extend operational life in portable robotic platforms.

REFERENCES

[1] Hennessy, J. L., & Patterson, D. A., *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann, 2019.

[2] Asada, H., & Slotine, J. J. E., *Robot Analysis and Control*, Wiley-Interscience, 2017.

[3] Kumar, S., & Kumar, R., "Bluetooth Based Robotic Control Using Android Application," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 6, pp. 789–793, 2015.

[4] Singh, A., & Verma, R., "Design and Development of an Object Sorting Robotic Arm Using IR Sensors," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 5, no. 8, pp. 6789–6796, 2016.

[5] Patel, D., "FPGA-Based Robotic Arm Control Using Hardware Description Languages," *International Journal of Computer Applications*, vol. 169, no. 7, pp. 12–18, 2017.

[6] Bhattacharjee, T., et al., "Autonomous Robotic Arm Manipulation Using Sensor Feedback," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 923– 936, 2018.

[7] Lee, K., & Cho, H., "Design of Embedded Robotic Systems Using ARM Processors," *IEEE International Conference on Control, Automation and Systems (ICCAS)*, pp. 1152–1157, 2019.

[8] System-Verilog IEEE Standard 1800-2017, *IEEE Standard for SystemVerilog – Unified Hardware Design, Specification, and Verification Language*, IEEE Computer Society, 2017.

[9] Al-Asaad, H., "Functional Verification Using System-Verilog and QuestaSim," *Journal of Electronic Testing: Theory and Applications*, vol. 36, no. 3, pp. 345–358, 2020.

[10] Yaskawa Motoman, "Industrial Robotic Arms: Applications and Future Trends," *Robotics and Automation Reports*, 2021.

[11] Zhang, Y., & Wang, H., “Mobile Application Based Human-Robot Interaction for Autonomous Robots,” *International Journal of Robotics and Automation*, vol. 9, no. 2, pp. 55–63, 2020.

[12] Siciliano, B., & Khatib, O., *Springer Handbook of Robotics*, Springer, 2nd ed., 2016.