JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 07, ISSUE 08, 2020

Development of a Low-Power, High-Performance Processor Architecture for IoT Applications

MANIKA MANWAL,
Department of Comp. Sc. & Info. Tech., Graphic Era Hill University,
Dehradun, Uttarakhand, India 248002
DOI: 10.48047/jcr.07.08.611

Abstract

The Internet of Things (IoT) represents a new era of computing, where an enormous number of small, low-power devices with sensing and actuation capabilities are connected to the internet. These devices, often referred to as "smart" or "connected" devices, are designed to collect data from the environment, communicate with other devices and systems, and perform various tasks based on the collected data. The processors used in IoT devices have to meet various challenges to ensure the effective functioning of IoT applications. Some of the key challenges faced by IoT processors include low power consumption, limited battery power, limited memory space, high performance, low cost, and security. Additionally, IoT applications often require processors to interact with the environment in ways that differ from traditional computing systems, which can impose additional constraints such as environment invisibility, ruggedness, and real-time event timing properties. To address these challenges, the next generation of processors for IoT applications must be highly efficient in terms of power and performance while also providing strong reliability and security. Configurable and extensible processors are emerging as a potential solution to meet these requirements, providing flexibility in terms of both hardware and software design.

This paper discusses the key characteristics required for IoT processors and the possible configurations to meet the growing demands of the processor market. The authors also describe the architecture of two configurable processors that could be used in IoT applications. By addressing the unique requirements of IoT applications, these processors are expected to enable the development of a wide range of new and innovative IoT devices and systems.

Introduction

The widespread adoption of sensor networks in various applications has posed new challenges for connecting these devices to the internet while minimizing resource usage such as battery life, data storage, processing power, and human interaction. Wireless Sensor Networks (WSN) have gained significant attention in recent years and have great potential for integration with the Internet of Things (IoT). Various systems and architectures have been proposed that incorporate WSN with IoT for diverse applications such as smart homes, industrial monitoring, smart cities, etc., using RFID, ZigBee, and other protocols to deploy these applications. One promising application of low-power WSN is in the field of medical monitoring, where it can monitor physical activity, heart rate, muscle activity, and other physiological vitals for intensive care or chronically ill patients. WSN can also be used to monitor physical activity and movement patterns through a combination of sensors. In smart homes, WSN can monitor temperature, humidity, plant moisture, air quality, sound, and any unwarranted home activity to ensure better living conditions.

Despite the broad range of applications for WSN, their connectivity to the internet or cloud is often limited, which restricts the universal accessibility of the collected data. To address this limitation, new solutions are needed to enable WSN to connect to the internet and cloud, allowing for better management and analysis of the collected data.

Overall, the integration of WSN with IoT has enormous potential to transform various industries and improve the quality of life for individuals. However, the limited connectivity of WSN to the internet or cloud remains a significant challenge that needs to be addressed to unlock the full potential of these technologies.

Literature Review

The paper proposes a data-driven event triggering mechanism for Internet of Things (IoT) applications. The proposed mechanism is designed to optimize the energy consumption of IoT devices while maintaining the required level of responsiveness. The mechanism is based on a machine learning algorithm that predicts the occurrence of an event based on the available data. The algorithm learns from the past events and their corresponding data to predict the future events [1].

The paper proposes a body node coordinator placement algorithm for Wireless Body Area Networks (WBANs) that optimizes the network performance in terms of energy consumption and packet delivery ratio. The proposed algorithm considers the heterogeneity of the body nodes and the interferences between them in the placement of the coordinators [2].

The paper presents the design and implementation of a wireless sensor network (WSN) based room monitoring system that monitors and controls the temperature, humidity, and illumination of a room. The proposed system consists of a sensor node, a coordinator node, and a gateway node. The sensor node collects data from the sensors and sends it to the coordinator node, which processes the data and sends it to the gateway node for remote monitoring and control [3].

The paper proposes a sensor fusion methodology for a mobile robot that enables it to avoid dynamic obstacles in real-time. The proposed methodology integrates data from multiple sensors, including a LiDAR sensor, a stereo camera, and an inertial measurement unit (IMU), using a Kalman filter. The resulting output from the Kalman filter is then used to generate control commands for the robot's actuators [4].

The paper presents the challenges involved in building such a testbed and the design decisions made to address those challenges. The authors also provide a detailed evaluation of the Livelabs system, including its performance and scalability. The results show that Livelabs is capable of providing an effective platform for conducting mobile experimentation, and the authors believe that it has the potential to significantly advance the field of mobile computing research [5].

The paper presents a comprehensive survey on static and mobile wireless sensor network (WSN) experimentation facilities. It analyses various factors that should be considered when choosing an experimentation platform for WSNs, such as the type of application, the size of the network, the level of realism required, the cost of deployment, and the ease of use. The survey covers a wide range of experimentation platforms, including emulators, simulators, testbeds, and real-world deployments. It provides an overview of the features, advantages, and limitations of each platform, as well as some examples of how they have been used in research and education. The paper concludes with some recommendations on how to choose the most appropriate experimentation platform for a given WSN application [6].

ISSN- 2394-5125 VOL 07, ISSUE 08, 2020

The paper describes how mobile applications can provide a user-friendly interface for controlling and monitoring IoT devices by using a lightweight IoT framework. The authors designed a framework that can easily integrate with mobile applications and provide the necessary communication and control functions for IoT devices. The framework was implemented and tested on a number of IoT devices, including a smart thermostat and a smart lighting system, showing promising results in terms of usability and performance [7].

The proposed system integrates various sensors and devices such as temperature, humidity, light, gas, and motion sensors, as well as switches and actuators for controlling home appliances, into a WSN to provide real-time monitoring and control of the home environment. The authors describe the system architecture and the communication protocols used for data transmission between nodes and the gateway. The results of experiments conducted to evaluate the system's performance show that the proposed system is effective and reliable for home automation applications [8].

Some general principles that can help in designing IoT devices with low power consumption:

i) Use Low-Power Components: Select components that have low power consumption, such as low-power microcontrollers, sensors, and wireless modules. These components are designed to use as little power as possible, which can extend the battery life of your device.

ii) Optimize Your Code: Write efficient code that minimizes the amount of processing required. This can be achieved by using optimized algorithms, avoiding unnecessary computations, and reducing the amount of data transmitted.

iii) Manage Power Modes: Use power management techniques such as sleep modes to reduce power consumption when the device is idle or not in use. This can help conserve battery life and extend the operating time of the device.

iv) Use Energy Harvesting: Incorporate energy harvesting technologies such as solar panels or kinetic energy harvesters into your design. This can help recharge the battery or power the device directly.

v) Consider the Environment: Consider the operating environment of your device and optimize its design to reduce power consumption. For example, if the device operates in a low light environment, consider using a low-power display or using audio cues instead of a visual display.

By following these general principles, we can design IoT devices with low power consumption and longer battery life.

The architecture of an IoT node typically includes four main components: the MCU, the sensor, the wireless transceiver, and the power management circuit.

The MCU is responsible for processing the signals received from the sensors and sending the processed data to the wireless transceiver for transmission. The sensor part of the node collects data from the environment and provides it to the MCU. The wireless transceiver unit transmits the data from the node to a relay station in the network. Finally, the power management circuit provides power to each component on the board and ensures that the voltage of the power rail is within specifications (Figure 1).

By partitioning the IoT node into these function blocks, designers can optimize the performance and power consumption of each component individually. This helps to reduce the overall power consumption of the device and improve its reliability.

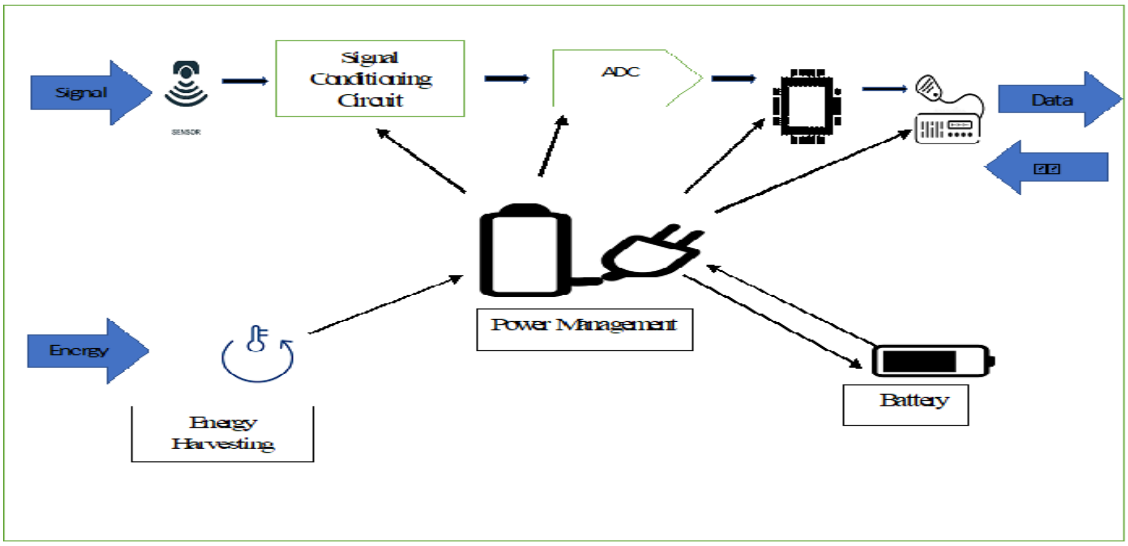

Figure 1:

IoT devices have unique characteristics that differentiate them from other computing systems. Some of these key characteristics include:

i) Size: IoT devices are typically small in size, often referred to as "tiny" or "micro" devices. This small size means that the processors used in IoT devices must also be small in size and highly compact.

ii) Power: Many IoT devices are powered by batteries, which means that they must operate with extremely low power consumption to conserve battery life. This requires processors that are designed to be highly energy-efficient.

iii) Memory: IoT devices often have limited memory, which means that the processors used in these devices must be designed to operate with minimal memory requirements.

iv) Communication: IoT devices rely heavily on communication with other devices and systems. This means that the processors used in IoT devices must be capable of handling communication protocols and networking technologies.

v) Sensing and Actuation: IoT devices are designed to collect data from the environment and perform various tasks based on that data. This means that the processors used in IoT devices must be able to handle sensing and actuation capabilities.

These unique characteristics of IoT devices require new designs and scenarios for processors to meet the specific demands of IoT applications. The development of new and innovative processors for IoT devices is critical to enabling the growth and expansion of the IoT industry.

Classification of IoT Applications and Processor Requirements

IoT applications can be classified into different categories based on their use cases, such as smart homes, smart cities, industrial IoT, healthcare, and agriculture. Each of these categories has its own set of requirements and challenges, which impacts the processor requirements for IoT devices.

i) Smart Homes: IoT devices used in smart homes are designed to provide automation and control of various home appliances and systems. These devices require low-power, cost-effective processors that are capable of handling multiple communication protocols and supporting real-time data processing.

ii) Smart Cities: IoT devices used in smart cities are designed to improve the management and efficiency of urban infrastructure, such as traffic lights, waste management, and public transportation. These devices require processors that can handle large amounts of data processing and communication while also being highly energy-efficient.

iii) Industrial IoT: IoT devices used in industrial settings are designed to monitor and control various industrial processes and equipment. These devices require processors that are rugged and reliable and can operate in harsh environments with high levels of noise and vibration.

iv) Healthcare: IoT devices used in healthcare applications are designed to monitor and track patient health data and provide remote patient monitoring. These devices require processors that are highly secure, energy-efficient, and capable of processing real-time data.

v) Agriculture: IoT devices used in agriculture are designed to provide precision farming and crop management. These devices require processors that can handle data processing for multiple sensors and communication protocols while also being rugged and reliable enough to operate in harsh outdoor environments.

The processor requirements for IoT devices depend on the specific use case and application requirements. However, some common processor requirements for IoT devices include low power consumption, energy efficiency, high performance, reliability, security, and the ability to handle multiple communication protocols and real-time data processing. The development of processors that can meet these requirements is crucial to the growth and success of the IoT industry.

Processor settings for Internet of Things applications

There are different processor configurations that can be used to meet the requirements of IoT applications. Here are a few examples:

i) System on Chip (SoC): SoC is a single chip that integrates all components of a computer or other electronic system. For IoT applications, SoCs are designed to include the processor, memory, communication interfaces, and other necessary components in a compact and energy-efficient package.

ii) Microcontroller Units (MCUs): MCUs are integrated circuits that include a processor, memory, and input/output peripherals in a single package. MCUs are used in low-power and cost-sensitive applications such as smart home devices and wearables.

iii) Field Programmable Gate Arrays (FPGAs): FPGAs are integrated circuits that can be programmed to perform specific functions. They can be reprogrammed to adapt to changing application requirements, making them suitable for applications that require flexibility, such as smart cities and industrial IoT.

iv) Digital Signal Processors (DSPs): DSPs are specialized processors designed to handle digital signal processing tasks, such as audio and video processing. They are used in applications such as smart homes and healthcare, where real-time processing of data is critical.

v) Graphics Processing Units (GPUs): GPUs are specialized processors designed to handle graphics-intensive tasks such as image and video processing. They are used in applications such as surveillance and smart cities.

Different processor configurations can be used for IoT applications based on the specific requirements and constraints of the application. The processor configurations discussed above are just a few examples of the types of processors that can be used for IoT applications. It's important to choose the right processor configuration to ensure that the IoT device can handle the required tasks efficiently, while also meeting power, size, and cost constraints.

Processor configurable:

Synopsys' Design Ware ARC Processors are a versatile family of 32-bit CPUs that can be tailored to meet the needs of different embedded systems and applications. The processors use a 16-/32-bit instruction set architecture (ISA) that provides high performance and code density, making them ideal for System-on-Chip (SoC) and embedded applications. They are supported by a wide range of open-source and commercial software development tools, and can be implemented in any process technology.

One of the key advantages of ARC processors is their ability to optimize power, performance, and area (PPA) efficiency for a wide variety of embedded applications. This customization is possible through the use of configurable processor cores that can be optimized for specific application requirements. This approach allows designers to achieve the optimal balance of power consumption, processing performance, and silicon area for their particular use case.

In addition to their flexibility and configurability, ARC processors offer a broad range of features and capabilities that are well-suited for embedded applications. These include support for real-time operating systems, low-power modes, hardware accelerators, and advanced debugging and profiling tools.

Overall, Synopsys' Design Ware ARC Processors provide a highly flexible and customizable solution for embedded system designers, offering a wide range of features and capabilities that can be tailored to meet the specific requirements of a given application.

ARC EM processors:

ARC EM processors are a subset of the Design Ware ARC Processor family, designed specifically for embedded applications that require ultra-low power consumption and small silicon area. The EM stands for "embedded microcontroller" and these processors are optimized for use in power-constrained applications such as Internet of Things (IoT) devices, wearables, and sensors.

The ARC EM processors use a 32-bit RISC instruction set architecture (ISA) with a configurable pipeline that allows for fine-grained control over power consumption and performance. They also incorporate a range of power-saving features such as dynamic voltage and frequency scaling (DVFS) and hardware accelerators for common signal processing tasks.

ARC EM9D/EM11D Processors

The ARC EM9D and EM11D processors are two members of the ARC EM family of ultra-low power, high-performance embedded microcontroller cores. They are designed for applications that require a combination of low power consumption, high processing performance, and small silicon area.

Table 1: ARC processor Families

ARC Family	Features
ARC HS Family	Cache Size : 64kb
	Architecture : ARC v2 ISA
	CCM: 16MB
	DSP : HS45D & HS47D
	XY Memory: Not present
	Extension : APEX
	FPU: Optional
	MPU: Optional
	Sleep Modes: Yes
ARC 700 Family	Cache Size: 8KB-64KB
	Architecture: ARC compact ISA
	CCM: 8KB to 512 KB
	DSP : Optional
	XY Memory: Not present
	Extension : User defined
	FPU: Optional
	MPU: Not Present
	Sleep Modes:Yes

Conclusion:

It can be difficult to optimise CPUs for Internet of Things applications when electricity is becoming increasingly scarce. This paper investigated ways to permit such optimisations while keeping a generic synthesis and programming flow for a selected 32-bit RISC processor. As anticipated, the acquired findings demonstrate that, depending on the application, optimisations may be gained by enhancing individual hardware blocks. A key finding from the investigation is that a core with less hardware complexity allows for the optimum power efficiency for general applications. Additionally, creating a comprehensive strategy for programming such processors is a difficult endeavour. For investigating their use in IoT applications, maintaining compatibility with a single technique for all versions of a number of application-specific customised cores is crucial. The suggested cores are now being used in multi-processor systems-on-chip, enabling the development of hybrid solutions where several core iterations are combined in a single system. Finally, utilising a modified E32SS core and a set of peripherals and programmable memory, we recently taped out a microcontroller. HF-RISC was verified on silicon after the device underwent testing.

References

[1] P. Kolios, C. Panayiotou, G. Ellinas, M. Polycarpou, "DataDriven Event Triggering for IoT Applications," IEEE Internet of Things Journal, vol.PP, no.99, pp.1-1, May 2016

[2] M. T. I. ul Huque, K. S. Munasinghe and A. Jamalipour, "Body node coordinator placement algorithms for wireless body area networks," IEEE Internet of Things Journal, vol.2, no. 1, pp. 94- 102, Feb 2015.

[3] S. Noh, K. Kim, and Y. Ji, "Design of a room monitoring system for wireless sensor networks," International Journal of Distributed Sensor networks, vol. 2013(2013), no. 189840, pp. 1-7, Jul 2013.

[4] M-S. Aman, M-A. Mahmud, H. Jiang, A. Abdelgawad, K. Yelamarthi, "A Sensor Fusion Methodology for Dynamic Obstacle Avoidance Robot," IEEE International Conference on Electro/Information Technology, May 2016

[5] R. K. Balan, A. Misra and Y. Lee, "Livelabs: Building an in-situ real-time mobile experimentation testbed," Proceedings of the 15th Workshop on Mobile Computing Systems and Applications, pp. 14:1-14:6, Feb 2014.

[6] A-S. Tonneau, N. Mitton, and J. Vandaele, "How to choose an experimentation platform for wireless sensor networks? a survey on static and mobile wireless sensor network experimentation facilities," Ad Hoc Networks, vol. 30, pp. 115–127,July 2015.

[7] S. K. Datta and C. Bonnet, "Connect and Control Things: Integrating Lightweight IoT Framework into a Mobile Application," IEEE International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 66-71, Sep 2014.

[8] J. Zhang, G. Song, H.Wang, and T. Meng, "Design of a wireless sensor network based monitoring system for home automation," Int. Conf. Future Comput. Sci. Appl. (ICFCSA), pp. 57–60, Jun 2011.