ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Implementation of Artificial Intelligence Algorithm in Embedded System

SHEELA M.G.

Lecturer in Electronics, Women's Polytechnic College, Ernakulam, Kerala, India

ABSTRACT: An embedded system is a computer system that is intended to do a specific set of tasks with the least amount of input from the user. When a system is able to respond quickly and effectively to changing circumstances, it is an intelligent system. Embedded systems' dependability and real-time demands, as well as restrictions on cost, size, and power consumption, provide the greatest hurdles to intelligent solutions. Neural networks and genetic algorithms are two examples of biologically inspired intelligent solutions for embedded systems. The use of multi-agent systems for non-time-critical embedded system services is also a possibility. A third field is soft computing, which is able to model complex (sensory) data in a precise manner. Finally, because embedded systems often provide important services, intelligent validation approaches are required to aid developers in determining whether the system is appropriate for its intended use. They are necessary.

KEYWORDS: AI, EmbeddedSystem , Artificial Intelligence, Real-Time

I. INTRODUCTION

It is clear that the Internet of Things and artificial intelligence are linked and that embedded technology has a massive effect on both. Based on this, the application of artificial intelligence algorithms in embedded systems is researched. As a result of the enhancement of embedded system hardware configuration and embedded system software running, operation steps of the embedded system are simplified, enhancing the embedded system's effectiveness, strengthening the research of the embedded technology, and increasing investment in manpower and material resources in embedded system research. [1] Using an artificial intelligence algorithm to increase the embedded system's performance has been proven by the results of the experiments

Embedded system design differs greatly from desktop programming in many ways. Standard environments with practically infinite (virtual) memory and a robust debugging interface are available to desktop programmers. Instead, an embedded programmer is forced to give up such conveniences. Many various microcontrollers must be dealt with, some of which have only a few bytes of working memory and only a few kB of programme memory. The only debugging tools available to software developers are LEDs, oscilloscope displays, and a serial data stream. [2] It is important that embedded systems make efficient use of the limited resources available to them. This means that they should consume less power, use less programme and working memory, and take up less physical space. With only a few lines of source code, "dumb" programmes have been used to construct many embedded applications in the past. For example, a car's electronic brake system should be able to deliver its service in a variety of extreme scenarios, such as when a braking element on one wheel fails. However, the embedded market requires more wide and intelligent applications. [3]

II. EMBEDDED SYSTEMS

To put it another way, an embedded system (ES) is simply a computing system with limited resources that is integrated into a larger system as a software component, a hardware component, or a combined software/hardware component, and which constantly interacts with the physical world or the external environment via sensors/actuators to accomplish a specific task. When discussing ESs, we sometimes refer to them as a subset of cyber-physical systems (CPS). ES

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

should meet a set of conflictual objectives notably timing deadlines, reduced energy consumption, small memory footprint, small weight, low cost, and reliability.
A micro-controller serves as a basic feedback loop in conventional ES, where sensors and actuators are used to interact with a fully described environment and are assumed to accomplish a set of predetermined tasks. Electronic systems have been evolving in response to the ever-increasing demands for multi-application execution support and high performance computing, large scaling, scalability, autonomy and self-adaptation, security and at the same time to support the next-future technologies such as wireless/optical communication, sustainable energy and IoT (internet of things). [4] This necessitates the development of new design methods or the adjustment and enhancement of existing methods to address these rising issues.

Game Theory
Multiplayer decision theory, or GT, is a multi-player decision theory, and any game must explain who the participants are, what information and actions they have, and what the payoffs are for each event. In a game, the goal is to maximise one's gain by making a rational decision. Game theory is built around the idea of equilibrium. Equilibrium is the goal of both optimization problems and games. As a result, the latter describes a stable state in which either one outcome occurs or a group of outcomes occurs with known probability.
To apply GT to RT scheduling, imagine each task as a self-interested agent with the freedom to choose its own processor. Every processor publishes its policy for scheduling in advance, which causes a game of simultaneous moves amongst the jobs. A task's approach is to decide which processor it will run on. The game approaches an equilibrium where no task can optimise its cost function by moving to a different processor. Scheduling policies for each processor should be designed to minimise the amount of ineffectiveness caused by selfish behaviour, the goal of a system designer. [5]

III. REVIEW OF LITERATURE

An Embedded System (ES) interacts withthe outside world via the sensors/actuators and subjected tostrict spatial, temporal and energy constraints. Indeed, ESare heterogeneous in nature. They typically combinesoftware components (general purpose processors, DigitalSignal processors, etc.) and hardware components(ASIC for application specific integrated circuit, FPGAfor field programmable gate arrays). Unlike a hardwareimplementation, a software implementation has theadvantage of providing flexibility (i.e., the possibility ofreprogramming), but at the prize of satisfying performanceconstraints. An ES is called real-time if it is able to meet itstiming constraints. The principal role of ESo is not thetransformation of data as in conventional software, butrather the interaction with the physical world. It executes onmachines that are not, first and foremost, computers. Theyare cars, airplanes, telephones, audio equipment, robots,appliances, toys, security systems, pacemakers, heartmonitors, weapons, television sets, printers, scanners,climate control systems, manufacturing systems, and so on(Lee, 2002)[6].
Embedded systems (ES) (Gajski et al., 1994) are changingour daily life. They are commonly found in consumerelectronics, games, telecommunication, industrial, control,automotive, aeronautics and military applications. ESdevelopment represents a hot topic in both academicresearch and industry. Contrary to conventional informationsystems, ES design needs software and hardware to bedesigned in a concurrently synergistic fashion so the systemfunctional/non-functional requirements are met. This newstyle of design is called codesign (Schaumont, 2012).Codesign is a collaborative and creative task requiringsome specific skills in hardware, software and systemengineering. Semi-conductor technology evolution (Moorelaw) pushes ES to be implemented as system on chip (SOC)where all system functional elements or components areintegrated in only one chip. Under time-to-market pressure,special customer requirements, rapid technologies changing,increasing applications complexity, and diversity of designstyles, methodologies and associated tools, ES designersmust be assisted along the design process efficiently andinteractively in order to minimise the cost of developmentand increase the productivity: many concepts have beenborrowed from the software community especially higherlevels of abstraction, knowledge and experience reuse,project planning, cost/risks estimation, etc.
[9] explains how ML was used to forecast DRC violations during the global routing stage of the project. Detailed routing DRC breaches might cause or delay tape outs. Automatic place and route tools produce a worldwide routing map to prevent tape out delays. In order to reduce or eliminate DRCs, the placer uses the routine map to place blocks in

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

the most efficient manner possible. As a result, at advanced nodes, the correlation between global and detailed routing decreases, resulting in either aggressive optimization by placers at the expense of area and/or power or excessive DRCs in the final routing stage.
in order to expedite R&D, optimise product portfolios and assist businesses in reducing costs [10], the authors argue that sophisticated analytics for semiconductor design is necessary. While traditional analytics relies on data that has already been gathered and analysed in order to have a better knowledge of it, advanced analytics is a process that begins with a clear definition of what data is needed in order to make a decision. The use of advanced analytics by project teams resulted in a 10% reduction in project duration, according to a study spanning 200 projects.
In another work, Cheng and Druzdzel (2000) develop an algorithm for evidentialreasoning in large Bayesian networks. An adaptive importance sampling algorithm, AIS-BN that shows promising convergence rates even under extreme conditions is developed.It seems to outperform the existing sampling algorithm consistently. This provides abetter substitute to stochastic sampling algorithms that have been observed to performpoorly in evidential reasoning with extremely unlikely evidence.

Objectives

- To investigate the embedded system's multitasking real-time processing platform.
- to learn the fundamentals of embedded system design
- study of AI processor and embedded system block diagram
- To learn about embedded system

IV. RESEARCH METHODOLOGY

Methodology for doing research is a general approach to the gathering of data, the evaluation of the data and the presentation of conclusions based on the findings. A strategy for conducting a study is known as a research technique. To put it another way, research is the systematic collection and analysis of data to enhance knowledge in any field. The study's objective is to develop intellectual and practical solutions to difficulties via the application of systematic methodologies. The information gathered for this research is secondary in nature, having been taken from a variety of previously published sources. The information used to prepare this report was culled from a variety of reliable sources online.

V. RESULT AND DISCUSSION

A block schematic is shown in Fig 1 of the embedded artificial intelligence system. An external data transceiver serves as an interface to other systems, and an AI processor with numerous AI cores performs learning and recognition activities. [12]

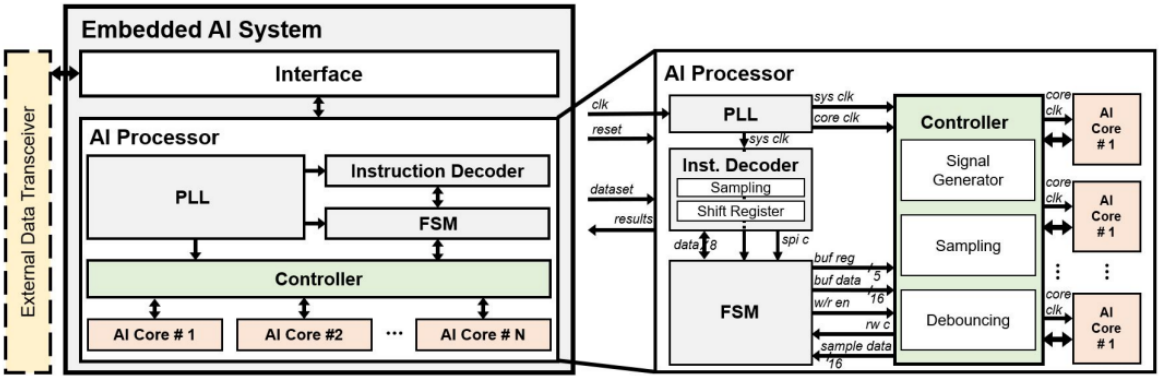


Fig. 1 is a detailed block diagram of the embedded AI system and AI processor.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

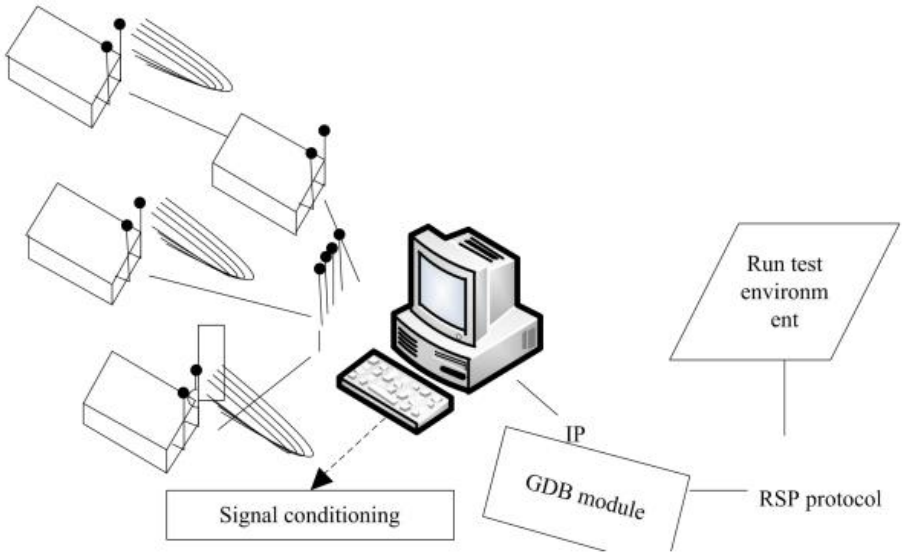
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Using a serial peripheral interface (SPI) protocol, the external data transceiver transmits data to the interface that is used for learning and recognising activities. phase-locked loop (PLL), PLL, instruction decoder, finite state machine (FSM), and controller for efficient management of AI cores are part of the AI processor. In order to keep the controller running at 64MHz and the AI cores at 16MHz, the PLL is used. Decoding a protocol for learning/recognition, the instruction decoder in the AI processor gets the dataset over the interface [13]

Embedded devices are mostly computer systems and other hardware that are integrated within the device. System software, hardware, software layer, and intermediate layer make up the embedded computer, which is the heart of the system. The term "software layer" refers primarily to the software layer that is utilised in the development of embedded systems. Memory, embedded processor, universal device interface, and more are all part of the hardware layer. The embedded system's core module is this component. [14]



Structure diagram of an embedded system (Fig. 2)

File system, network protocol, graphics, and RTOS make up the majority of the software layer. Device drivers are independent of the hardware because of the intermediate layer, also known as driver layer software, which separates hardware from operating system software. An optimised hardware configuration is based on this.[5-6]. Figure 2 depicts the embedded system's basic structure. [15]

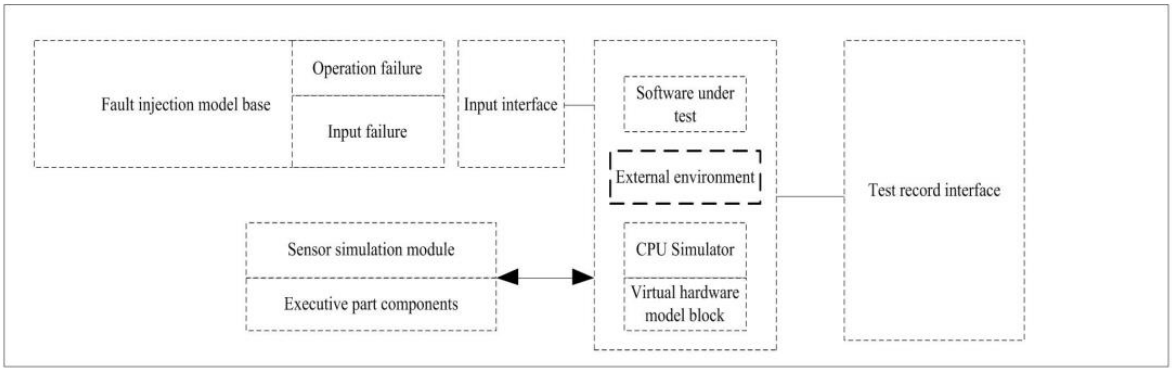
The embedded microcontroller (MCU) coupled with in-house and third-party embedded software to achieve the desired result. The intelligence of embedded devices is unaffected by the lack of a general embedded system software. For example, it may be written in VC + +. [16]

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016



The embedded system's multitasking real-time processing platform (Fig. 3).

The key to an embedded system is its application software, which has high storage capacity needs as well as high requirements for the quality and dependability of its software code. Multi-tasking in real-time raises the bar for real-time performance [11-12]. The embedded system's multi-task real-time processing platform is built on this foundation (Fig. 3).

V. CONCLUSION

People's lives are made infinitely easier by advances in artificial intelligence and Internet of Things (IoT) embedded technologies alike. Thus, an investigation into artificial intelligence algorithm implementation in embedded systems can be conducted. As a result of a better understanding of the embedded system hardware configuration and software running algorithm, as well as an increase in investment in human and material resources in research into embedded technology, the embedded technology is applied to the Internet of Things, to improve the embedded system's effectiveness.

REFERENCES

1. Abdeyazdan M. Parsa S. Rahmani A. M. (2013). Task graph pre-scheduling, using Nash equilibrium in game theory.The Journal of Supercomputing, 64(1), 177–203. 10.1007/s11227-012-0845-z
2. Agrawal, P., & Rao, S. (2012). Energy-Aware Scheduling of Distributed Systems Using Cellular automata. 6th Annual IEEE International Systems Conference (IEEE SysCon 2012).
3. Das Sumit, Dey Aritra, Pal Akash, Roy Nabamita. (2015) “Applications of Artificial Intelligence in Machine Learning: Review and Prospect”
International Journal of Computer Applications 115: 31-41.
4. Bolaji A. L. Khader A. T. Al-Betar M. A. Awadallah M. A. (2013). Artificial bee colony algorithm, its variants and applications: A survey.Journal of Theoretical and Applied Information Technology, 47(2), 434–459.
5. Di Nardo Mario; Gallo Mosè; Madonna Marianna; SantilloLiberatina C. (2015) “A conceptual model of human behaviour in socio-technical
systems”. Communication in Computer and Information Science 532: 598-609.
6. Lee, E.A. (2002) Embedded Software, Advances in Computers,
Zelkowitz, M. (Ed.), Vol. 56, Academic Press, London.
7. Gajski, D.D., Vahid, F., Narayan, S. and Gong, J. (1994) Specification and Design of Embedded Systems, Prentice Hall, Englewood Cliffs, NJ.
8. Schaumont, P. (2012) A Practical Introduction to Hardware/Software Codesign, 2nd ed., Springer Publisher, New York, ISBN: 978-1-4614-3736
9. B. Yu, D. Z. Pan, T. Matsunawa and X. Zeng, "Machine learning and pattern matching in physical design," The 20th Asia and South Pacific Design Automation Conference, Chiba, 2015, pp. 286-293

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

10. Nergiz, M.E.; Clifton, C. -presence without complete world knowledge. IEEE Trans. Knowl. Data Eng. 2010, 22, 868–883.
11. Cheng, J. and Druzdzel, M. J., AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks, Journal of Artificial Intelligence Research, Vol. 13, pp.155-188, 2000.
12. Huang, J., Olaf Blech, J., & Raabe, A. (2011). Analysis and Optimization of Fault-Tolerant Task Scheduling on Multiprocessor Embedded Systems. Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Taipei. 10.1145/2039370.2039409
13. B. Yu, D. Z. Pan, T. Matsunawa and X. Zeng, "Machine learning and pattern matching in physical design," The 20th Asia and South Pacific Design Automation Conference, Chiba, 2015, pp. 286-293
14. G. Batra, Z. Jacobsen, N. Santhanam, “Improving the semiconductor industry through advanced analytics”, McKinsey Article, March 2016.
15. M. T. Jensen. Improve robustness and flexibility of tardiness and total flow-time job shops using robustness measures. Applied Soft Computing, 1:35–52, 2001.
16. A. Abraham and B. Nath. Designing optimal neuro-fuzzy systems for intelligent control. In Proceedings of the 6th International Conference on Control, Automation, Robotics, and Vision, December 2000.