International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7600
***Article Received**: 30 May 2023 **Revised**: 28 July 2023 **Accepted**: 12 August 2023*

An Intelligent Fault Alert Mechanism for Dynamic IoT Communication Microarchitecture

Nitesh Gaikwad[1], Dr. Shiyamala. S[2]
[1]Research Scholar, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi-Vel Tech Road Vel Nagar, Avadi, Chennai, Tamil Nadu 600062.
Email: niteshgaikwad78@gmail.com
[2]Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi-Vel Tech Road Vel Nagar, Avadi, Chennai, Tamil Nadu 600062.
Email: drshiyamala@veltech.edu.in

Abstract— The usage Internet of Things (IoT) was maximized throughout the entire world. Hence, the different core processors incorporated microarchitecture makes this IoT communication system. However, the rise of faults due to the malicious event and the data overload might maximize energy and power utilization. So, the current study has proposed a novel Chimp-based Domain adaptation Alert System (CbDAAS) for the dynamic IoT communication microarchitecture. Before initiating the communication sharing process, the present fault in the designed IoT dynamic core microarchitecture was predicted, and those cores were removed for the current data broadcasting process. Henceforth, the designed fault alert microarchitecture is tested in the MATLAB platform. The reliability was valued using different metrics like power usage, energy consumption and detection exactness value. Finally, the validated metrics were compared with the associated studies and scored the finest outcome in fault detection score as 98% and less energy usage at 0.025mj.

Keywords- fault alert system; dynamic microarchitecture; power utilization; fault detection exactness; energy consumption.

I. INTRODUCTION

In the IoT Reconfigurable Architectures, the edge computing prototype was frequently used feed circuit [1]. Furthermore, the pipeline levels differ depending on the performance and scalability of each assessment [2]. Diverse framework pipelines investigated various speed capacitance sensors to identify the best subsystem in edge-level IoT [3]. Instead of frequency modulation bits, strategies based on signaling have transmitted characteristics such as binary and word [4]. As a result, the encoding subsystem is the critical mechanism for carrying out this process [5]. Moreover, the main source of concern for energy usage in IoT systems is data transmission bandwidth and speed of response, which are regarded as cloud computing operating costs [6]. All digital apps have progressed with microprocessors, each with a dedicated processing unit for an intended area [7]. However, causing damage to the microarchitecture element influenced the whole system by ingesting a large amount of power and assets [8]. As a result, the project's schedule is cautiously signed by incorporating all functioning variables [9]. If one variable is missed during the programme obtainable, that function cannot be revised in the multiprocessors [10]. Because of the rising popularity of IoT applications, a wide range of gadgets and objects are aimed at providing services across the developed Internet [11]. This diversity of services and devices has introduced new challenges, necessitating the use of significance in various architecture models [12]. The major challenges in realizing IoT's full potential are interconnectivity, mobility, expandability, performance, protection, and privacy [13]. In addition, Interoperability is a significant challenge to overcome the device heterogeneity platforms [14]. Also, Protocol standardization and pattern explanation are critical for making all gadgets interoperable and accessible [15].

As a result, file transfer between the several gadgets must be controlled so confidential material is not negatively impacted [16]. Effective IoT systems necessitate an underlying network infrastructure capable of flexibly and effectively transmitting massive amounts of information [17]. A messaging system of this type should be capable of adapting to shifts in maintaining the services at a reasonable level [18]. In addition, the network route's traffic has increased tremendously due to a large number of users [19]. So, if the network suffered from data or user traffic, the highest communication delay was recorded, reducing the dynamic IoT process [20]. Considering all these issues, the microarchitecture with multiple processing chips was introduced for performing different communication processing functions [21]. However, those chips required more power energy to execute the specific Process because of the complex and different format data [22]. The key reasons for the dynamic microarchitecture's high energy and power consumption are high data overload and malicious threat events [23].

Considering these issues, the current study was intended to develop an intelligent optimal strategy as the fault alerting system to make awareness about the data overhead and malicious events. In the past, fault prediction processes existed, such as partitioning microarchitecture [27], mechanistic gap microarchitecture [28], etc., but the proper detection is still not reported. This has tended to cause high power and energy consumption. Considering these issues, the key objectives executed in this present study are alerting the high data overload and malicious events. The chimp optimal function and Domain adaptation prediction behaviours were considered for that. Here, optimal chimp functions were processed to enable the data overload and malicious prediction condition. Here, the malicious events in the IoT communication system were detected by measuring the taken to complete the data broadcasting process. The key contributions of the fault prediction dynamic IoT system is defined as follows,

- The IoT microarchitecture was developed by incorporating dynamic communication features in the initial phase.
- Then the novel CbDAAS was implemented as the fault alert system and incorporated into one control chip.
- Now, the IoT communication process was initiated for sharing the data between the IoT users
- During the data broadcasting process, the fault-alerted node was avoided from the data sharing process
- Here, the presented Chimp optimal features recorded the finest fault prediction score.
- Finally, the dynamic IoT communications' functioning parameters were valued and analyzed with other recent associated models.

The contents of this fault alert microarchitecture research study are arranged as recent associated dynamic microarchitecture work is exposed in the second section. The problem in the traditional dynamic microarchitecture is explained in the third section. The appropriate solution for the discussed issues is illustrated in section 4. The outcome of the defined solution is determined in the section.5, then the research study ended in the conclusion section.6.

II. RELATED WORKS

Few recent associated literatures of IoT dynamic microarchitecture is described as follows,

Security is the major concern for digital wireless communication, so Hamza Omar *et al.* [27] have introduced the partitioning-centric microarchitecture to improve the digital data transmission process in a dynamic environment. Here, the partitioning process was executed based on the user's counts and the data load types. Finally, this microarchitecture is validated by launching the state attack in the transaction channel. It has afforded a better malicious recognition outcome but has scored high energy consumption.

1) *For managing the chip reliability of the microarchitecture, the mechanistic gap-closing microarchitecture was introduced by Casey Paquola et al. [28]. Here, the strength of the connected chip in the microarchitecture is analyzed at each Process, and the reliability analysis report was prepared. Then the gained strength analysis report is updated in the microarchitecture controller based stored controller features the chips in the microarchitecture was maintained. But, it has reported high power usage.*

2) *Optimizing the power is one of the major concerns in dynamic microarchitecture. So, Beng-Liong et al. [29] have introduced optimal power chip dynamic microarchitecture for minimizing the power usage of the microcontroller. Here, the energy reduction program analyzer stores the power usage details. Finally, the designed optimal chip microarchitecture is implemented in the FPGA module, and the performance was noted by activating the communication process. But, less throughput ratio was gained.*

3) *A partial reconfiguration module was implemented by Wei kiat et al. [30]. Also, the executed partial reconfiguration microarchitecture was validated by connecting different IoT nodes with multiple microprocessors. This connected multiprocessor significantly allocated the required resources for the multi-user simultaneously. Hence, it is a multiple-job execution system with less duration. However, it is high in energy consumption.*

4) *Eunjin Baek et al. [31] have introduced the intelligent neural-based spatial-temporal model for microarchitecture memory management. Here, the memory status was identified through the Spatiotemporal features. Based on this, the required maintenance was executed. After analyzing the memory status, the performance of the microarchitecture was checked by performing the training and testing process through the neural network features. But it has consumed more power.*

III. DYNAMIC MICROARCHITECTURE WITH PROBLEM

Usually, the communication microarchitecture is rich with several commiseration cores, providing dynamic

DOI: https://doi.org/10.17762/ijritcc.v11i10s.7600
***Article Received**: 30 May 2023 **Revised**: 28 July 2023 **Accepted**: 12 August 2023*

communication process features [24]. However, any disturbance in the function module or the chip characteristics might affect the microarchitecture process [25]. In addition, if the overloaded core and the high duration required processing cores are not identified, it has resulted in chip damage that decreases the microarchitecture performance.

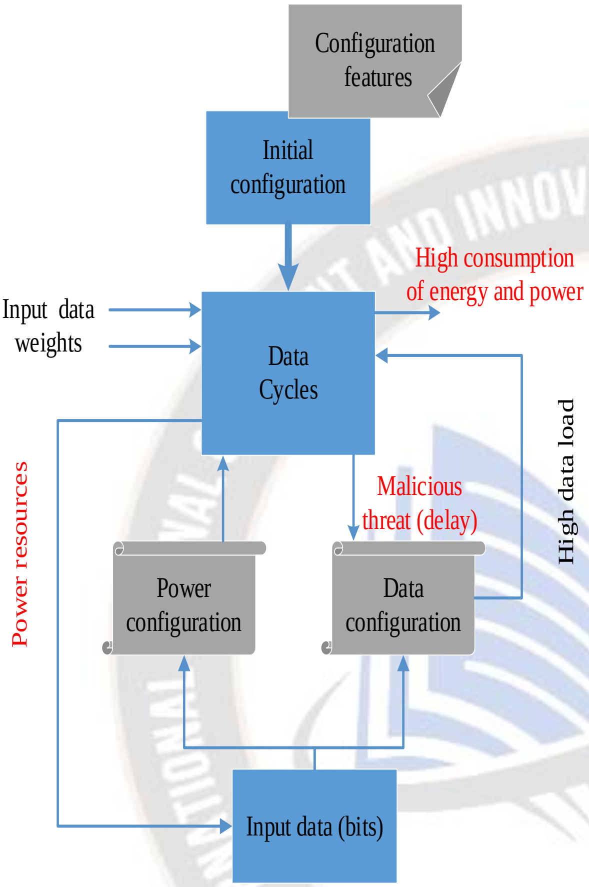

Fig. 1 issues in dynamic IoT microarchitecture

The issues in the traditional microarchitecture are defined in fig.1. Considering these issues, and the current study intends to incorporate the intelligent neural features in the dynamic microarchitecture for predicting faulty functional Processes.

IV. PROPOSED METHODOLOGY

A novel Chimp-based Domain adaptation Alert System (CbDAAS) was executed in this present research study as the fault alert module during the communication exchange process. Primarily, the communication microarchitecture was designed with dynamic communication features. Consequently, a novel CbDAAS is modelled in the control chip of the dynamic IoT microarchitecture. Here, the considered faults are overloaded IoT nodes and Malware events. These events were predicted by the intelligent chip process that connected to the microarchitecture. Once these issues were detected, the data processing function was continued with the help of read/write operation of the microarchitecture.

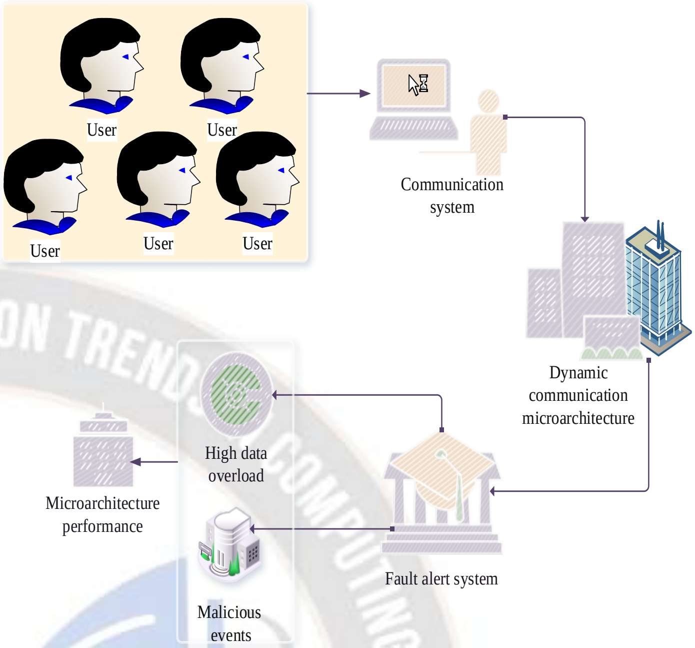

Fig.2 Proposed architecture

The proposed Design is defined in fig.2. Finally, the communication parameter constraints were valued and compared with the associated studies. It has revealed the robustness score of the designed intelligent microarchitecture over the other traditional microarchitecture.

4.1 Process of the proposed CbDAAS

The main objective that was incorporated in the microarchitecture is fault tolerant function. So, the fault-tolerant process condition is upgraded in the single chip based on the principle of optimal chimp function [33] and domain adaptation process [32]. The Design of the presented microarchitecture is exposed in fig.3. It is designed with several functional modules and cores that are Thread processor (TP), floating point (f), integer (int) and some lx cores. The internal structure of the novel CbDAAS microarchitecture is exposed in fig.3.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7600
***Article Received**: 30 May 2023 **Revised**: 28 July 2023 **Accepted**: 12 August 2023*

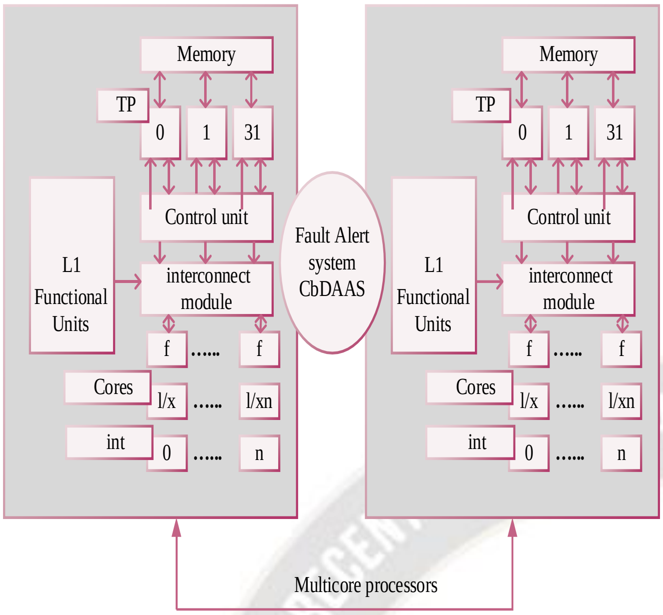

Fig.3 Internal process of CbDAAS

Here, the dynamic IoT microarchitecture was incorporated different processing modules like load controlling unit, data storing unit and data processing buffer. Hence, this architecture is suitable for multitasking. This microarchitecture is designed with a different pipeline and multiple-function process to offer the required, flexible range. The single chip controlled all the pipelines to offer the required energy and power resources.

Hence, the data initialization process in the dynamic microarchitecture is processed by Eqn. (1). Here, the data is denoted as T and the data initialization function parameter is determined as $F(T)$. Moreover, $1,2,3,4,5,6.....n$ determines the n number of data in the Process.

$$F(T)=T\{1,2,3,4,5,6.....n\} \tag{1}$$

After entering the data in the processing data model sample function was activated in the primary layer for normalizing the data by removing noise features. Hence, error removing sampling function is defined using Eqn. (2), which is the sampling function of the domain adaptation network.

$$T_s=\frac{T}{|T|^2+m} \tag{2}$$

The sampling process of the trained data is described as T_s and the m determines the clear data features. Hence, the filtered data is utilized for further Process after this sampling process. Then to design the fault alert system, the features variables were fixed for the data loading and malicious finding process. Here, the malicious event was alerted by fixing the average time for the data-sharing Process, which is equated in Eqn. (3).

$$m_c=\begin{cases} if(d_t\le 2.5ms) & normal \\ else & malicious \end{cases} \tag{3}$$

Here, d_t is the data broadcasting time of 1 byte; the optimal data sharing time range for sharing 1-byte data is 2.5ms, and the malicious condition is described as m_c. Here, the time range 2.5 is taken from the optimal chimp solution, which is the maximum iteration count of the chimp function. Here, the time of the each processing module in the microarchitecture is recorded by the memory buffer of the microcontroller(μc). Moreover, fixing the data load handling capacity is 0.7 GB for 1 large file. The value 0.7 is obtained from the random selection position of Chimp, which is 0.7. Hence, the data load parameter is defined, and the condition is equated in Eqn. (4).

$$T_l=\begin{cases} if(data\_size\le 0.7GB & optimal \\ else & fault \end{cases} \tag{4}$$

The two conditions are updated in the fault alert chip in the microarchitecture for finding the fault occurs during the data processing function.

```
Algorithm:1 CbDAAS
start
{
    $int T = 1,2...n;$
    // initializing the data transmission variables
    Data sampling ()
    {
        $int T_s, m;$
        // data sampling variables were initialized
        $T_s \to T(m)$
        // Collecting the noise-free features by Eqn. (2)
    }
    Fault Alert system ()
    {
        $int m_c, d_t;$
        Malicious fault checking
        $if(d_t \le 2.5ms)$
        {
            normal
        }
        Else (malicious fault)
Data load checking
$if\,(data\_size \le 0.7GB$
{
    Optimal data load
}
Else faults( data overload)
//Design of fault alert chip process by Eqn. (4) and eqn. (5).
}
Data transfer
}
stop
```

The dynamic IoT communication microarchitecture has incorporated different functional cores. Those core operations were controlled by the microcontroller, which is connected to the control unit of the microarchitecture. In this present study, the fault management system is connected between the functional core processors to identify the faults at any time. At every data broadcasting process, the function of the fault alert unit is activated to find the present fault in the microarchitecture. This function has tremendously increased the microarchitecture matching performance in every application. Also, resource usage like power and energy usage were minimized.

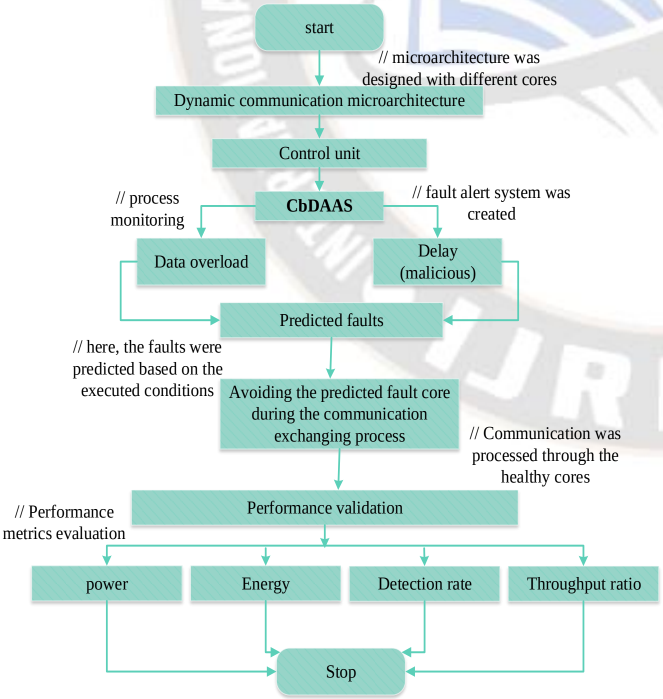
start
// microarchitecture was designed with different cores
Dynamic communication microarchitecture
Control unit
// process monitoring
CbDAAS
// fault alert system was created
Data overload
Delay (malicious)
Predicted faults
// here, the faults were predicted based on the executed conditions
Avoiding the predicted fault core during the communication exchanging process
// Communication was processed through the healthy cores
// Performance metrics evaluation
Performance validation
power
Energy
Detection rate
Throughput ratio
Stop

Fig. 4 CbDAAS work flow

After forecasting the faults, the communication exchange process was initiated. During the data broadcasting process, the predicted faults core in the microarchitecture was avoided. The expressed mathematical formulations are gathered in the flow model in fig.4 and exposed in algorithm.1.

V. RESULTS AND DISCUSSION

The planned intelligent fault alert microarchitecture is tested in the MATLAB platform. For scheduling the microarchitecture function in a scalable narrow way, the fault alert system was introduced to identify the improper workflow of data load scheduling and malicious recognition. The processing parameter constraints are exposed in table.1.

Table.1 Parameter constraints

Execution constraints	
Programming language	MATLAB
version	R2021a
Running environment	Windows 10
Microarchitecture	Fault alert microarchitecture
Application	Dynamic IoT communication

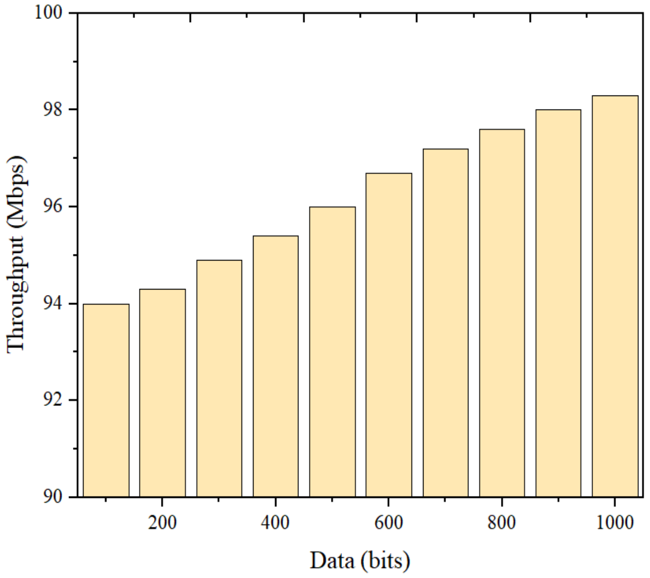

Fig.5 Throughput validation

Here, the throughput ratio was measured by taking the average time to process the data request. Here, the data sharing performance was measured in the form of Mbps, and the throughput formulation is equated in Eqn. (5).

$$Throughput = \frac{Data \quad request}{Time} \tag{5}$$

For the wireless communication system, measuring the throughput is the chief metric for data transmission performance. If the wireless framework was reported to have a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7600
***Article Received**: 30 May 2023 **Revised**: 28 July 2023 **Accepted**: 12 August 2023*

high throughput range, then those system is high in data transmission rate. Hence, the throughput assessment is exposed in fig.5. Here, the throughput measure is valued based on different data bits. Hence, data bits are varied based on 200 to 1000 bits. Here, the throughput varied between 94Mbps to 98.3Mbps. Moreover, the less throughput, 94 Mbps was gained, the less data, that is 100 bits and 98.3 Mbps was reported for the 1000 bits data.

5.1 **Performance analysis**

To determine the robustness of the intelligent microarchitecture, some of the functioning parameters were obtained and valued with other recent associated studies. The parameters considered for measuring fault alert microarchitecture performance are power, energy usage and detection rate.

5.2 Power usage

In this designed fault alert microarchitecture, measuring the power resource utilization is the most important factor. Before the initialization of the data processing function, the fault-alerted cores in the microarchitecture were avoided from the data processor. This Process can reduce the utilization of the microprocessor resources more than the traditional microarchitecture.

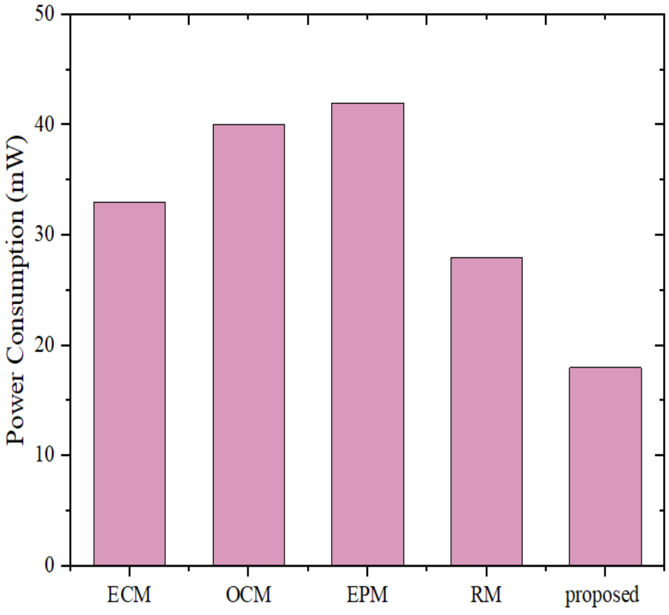

Fig.6 Evaluation of power consumption

The adopted microarchitecture to justify the results are Executable pipeline Microarchitecture (EPM) [30], Executable Core Microarchitecture (ECM) [30], Reconfigurable microarchitecture (RM) and Opposite Configuration Microarchitecture (OCM) [30]. Hence, the recorded power utilization range by the ECM is 33mW, OCM 40mW, EPM 42mW and RM 28mW. Considering these existing approaches, the designed, optimized microarchitecture was reported 18mW as the minimal power consumption. The comparison validation is viewed in fig.6.

5.3 Energy usage

The microcontroller process's processing duration determined the microarchitecture's energy utilization. The processing time was defined by considering the time required to design and execute the fault alert microarchitecture system. In addition, for evaluating the energy usage, some of the associated past microarchitecture models were taken that are Sensor Microcontroller (SM) [30], Crypto microcontroller (CM) [30], and Reduced instruction Microarchitecture (RIM) [30].

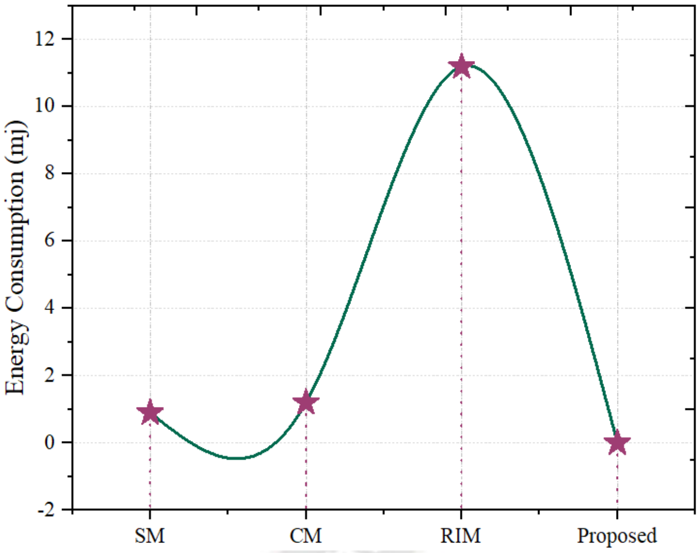

Fig.7 Energy utilization

The SM framework was reported at 0.057mj, CM was gained at 0.456mj, and the method RIM was registered at 11.19mj. Considering all these mechanisms, the designed microarchitecture reported a less energy usage rate of 0.025mj. These comparison statistics are described in fig.7.

5.4 Fault Detection rate

The fault detection rate is the chief parameter for this research study, which gives the exactness of the fault alerting system. This fault prediction accuracy was recorded based on the correct fault forecasting rate. Hence, to justify the robustness of the novel CbDAAS, some of the associated models were taken, such as self-test online Pro (SOP) [26] microarchitecture, Continuous stack Monitor (CSM) [26] and Snapshot Stack Monitor (SSM) [26].

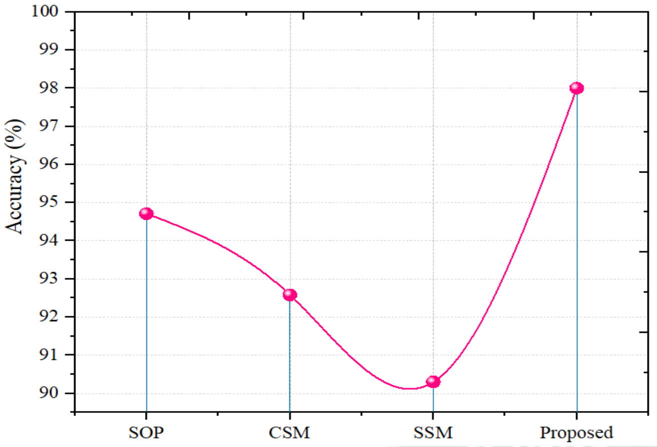

Fig.8 Fault forecasting exactness

The SOP model reported a 94.71% fault prediction exactness score; the CSM model reported a 92.58% detection rate, and the model SSM recorded a 90.3% detection score. The proposed model scored 98% fault recognition score, which is quite better than the compared models. Comparison statistics are defined in fig.8. The overall communication performance of the developed microarchitecture is tabulated in table.2.

Table.2 Communication performance

IoT dynamic communication metrics				
Data (bits)	**Data transfer rate (%)**	**Communication delay (ms)**	**Execution time (ms)**	**Throughput (Mbps)**
100	**80**	**14**	**9**	**94**
200	**82**	**14.5**	**10.5**	**94.6**
300	**86**	**15**	**12**	**94.8**
400	**88**	**15.3**	**13**	**95.3**
500	**90**	**15.7**	**14**	**95.8**
600	**92**	**16**	**15**	**96.3**
700	**93**	**16.2**	**16.5**	**96.8**
800	**94.2**	**17**	**17.2**	**97.3**
900	**95**	**17.5**	**18**	**97.8**
1000	**97**	**18**	**20**	**98.3**

5.2 Discussion

Besides, the execution time of the designed fault alert microarchitecture is in the 20s, and the maximum data transfer rate is 97%. In addition, the recorded communication delay score is 18ms. Hence, the overall outcome is exposed in table.2.

Table.2 Overall performance

Overall performance of designed microarchitecture	
Fault detection rate (%)	98
Energy utilization (mj)	0.025
Power (mW)	18
Throughput (Mbps)	98.3
Data transfer rate (%)	97
Communication delay (ms)	18
Execution time (ms)	20

The introduced novel CbDAAS has gained the finest outcome from all the metric performance evaluations. In addition, the wrong fault detection score reported by the proposed Design is 2%, which is quite less and in an acceptable state. Moreover, this error rate doesn't affect the fault detection performance. Hence, it was verified that the developed microarchitecture is suitable for dynamic IoT communication to enrich the data broadcasting performance.

VI. CONCLUSION

A novel CbDAAS mechanism was introduced in the microarchitecture application as a fault detection module. Before the data broadcasting process, these fault alerting functions were activated to find the highly overloaded data core and malicious behaviour. Here, the malicious events were predicted by the recorded time duration for forwarding the data bits. Finally, the designed microarchitecture improved the fault forecasting score up to 4% and reduced the power usage by 10% compared to the existing models. Also, the energy usage score was minimized by 1% than the previous approaches. Hence, the fault alert module in the communication microarchitecture has enriched the dynamic communication process more than the traditional dynamic microarchitecture. However, the security module is not executed in the designed microarchitecture. In future, designing the security process cores in the IoT dynamic communication microarchitecture will enrich the malicious event prediction and prevention process.

REFERENCES

[1] Parepalli, Ramanamma, and Mohan Kumar Naik. "Design alternatives of Network-on-Chip (NoC) Router microarchitecture for future Communication System." 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI). IEEE, 2022.

[2] Mirhosseini, Amirhossein, et al. "Q-zilla: A scheduling framework and core microarchitecture for tail-tolerant microservices." 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2020.

[3] Bourgeat, Thomas, et al. "Casa: End-to-end quantitative security analysis of randomly mapped caches." 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2020.

[4] Abu Al-Haija, Qasem, and Saleh Zein-Sabatto. "An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks." Electronics 9.12 (2020): 2152.

[5] Ghodrati, Soroush, et al. "Planaria: Dynamic architecture fission for spatial multi-tenant acceleration of deep neural networks." 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2020.

[6] Omar, Hamza, Brandon D'Agostino, and Omer Khan. "OPTIMUS: A security-centric dynamic hardware partitioning scheme for processors that prevent microarchitecture state attacks." IEEE Transactions on Computers 69.11 (2020): 1558-1570.

[7] Vicarte, Jose Rodrigo Sanchez, et al. "Opening pandora's box: A systematic study of new ways microarchitecture can leak private data." 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021.

[8] Tao, Wen, et al. "Review of the internet of things communication technologies in smart agriculture and challenges." Computers and Electronics in Agriculture 189 (2021): 106352.

[9] Dai, Yi, et al. "Microarchitecture of a Configurable High-Radix Router for the Post-Moore Era." International Conference on High Performance Computing. Springer, Cham, 2021.

[10] Nambiar, Vishnu P., et al. "Energy efficient 0.5 V 4.8 pJ/SOP 0.93 μW leakage/core neuromorphic processor design." IEEE Transactions on Circuits and Systems II: Express Briefs 68.9 (2021): 3148-3152.

[11] Rogers, Samuel, et al. "gem5-salam: A system architecture for llvm-based accelerator modeling." 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2020.

[12] Kumar, Chanchal, et al. "Post-silicon microarchitecture." IEEE Computer Architecture Letters 19.1 (2020): 26-29.

[13] Nour, Boubakr, et al. "A survey of Internet of Things communication using ICN: A use case perspective." Computer Communications 142 (2019): 95-123.

[14] Sreekumar, Rahul, and Mircea R. Stan. "Microarchitecture Optimization for Asynchronous Stochastic Computing." 2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS). IEEE, 2021.

[15] Barbirotta, Marcello, et al. "Evaluation of Dynamic Triple Modular Redundancy in an Interleaved-Multi-Threading RISC-V Core." Journal of Low Power Electronics and Applications 13.1 (2022): 2.

[16] Karthick, R., et al. "Overcome the challenges in bio-medical instruments using IOT–A review." Materials Today: Proceedings 45 (2021): 1614-1619.

[17] Souri, Alireza, et al. "A systematic review of IoT communication strategies for an efficient smart environment." Transactions on Emerging Telecommunications Technologies 33.3 (2022): e3736.

[18] Jeong, Ipoom, et al. "CASINO core microarchitecture: Generating out-of-order schedules using cascaded in-order scheduling windows." 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2020.

[19] Hamdan, Omar, et al. "IoT-based interactive dual mode smart home automation." 2019 IEEE international conference on consumer electronics (ICCE). IEEE, 2019.

[20] Tibaldi, Mattia, Gianluca Palermo, and Christian Pilato. "Dynamically-Tunable Dataflow Architectures Based on Markov Queuing Models." Electronics 11.4 (2022): 555.

[21] Dharsni, I. Thanga, Kirti S. Pande, and Manoj Kumar Panda. "Optimized Hazard Free Pipelined Architecture Block for RV32I RISC-V Processor." 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC). IEEE, 2022.

[22] Guan, Xiuxian, et al. "ROG: A High Performance and Robust Distributed Training System for Robotic IoT." 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2022.

[23] Prabha , G. ., Mohan, A. ., Kumar, R. D. ., & Velrajkumar, G. . (2023). Computational Analogies of Polyvinyl Alcohol Fibres Processed Intellgent Systems with Ferrocement Slabs. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 313–321. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2669

[24] Reddy, M. Lakshmi Prasad, and Sumanth Sakkara. "QNOC Isochronous Router with Efficient Dynamic Virtual channel and Error Termination." 2020 30th International Telecommunication Networks and Applications Conference (ITNAC). IEEE, 2020.

[25] Ji, Baofeng, et al. "Performance analysis of UAV relay assisted IoT communication network enhanced with energy harvesting." IEEE Access 7 (2019): 38738-38747.

[26] El Zouka, Hesham A., and Mustafa M. Hosni. "Secure IoT communications for smart healthcare monitoring system." Internet of Things 13 (2021): 100036.

[27] Banerjee, Mandrita, Carlo Borges, Kim-Kwang Raymond Choo, Junghee Lee, and Chrysostomos Nicopoulos. "A hardware-assisted heartbeat mechanism for fault identification in large-scale iot systems." IEEE Transactions on Dependable and Secure Computing (2020)

[28] Garcia, P., Martin, I., Garcia, J., Herrera, J., & Fernández, M. Enhancing Cyber security with Machine Learning-Based Intrusion Detection. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/157

[29] Omar, Hamza, Brandon D'Agostino, and Omer Khan. "OPTIMUS: A security-centric dynamic hardware partitioning scheme for processors that prevent microarchitecture state attacks." IEEE Transactions on Computers 69.11 (2020): 1558-1570.

[30] Paquola, Casey, et al. "Closing the mechanistic gap: the value of microarchitecture in understanding cognitive networks." Trends in Cognitive Sciences (2022).

[31] Tan, Beng-Liong, et al. "RISC32-LP: Low-Power FPGA-Based IoT Sensor Nodes With Energy Reduction Program Analyzer." IEEE Internet of Things Journal 9.6 (2021): 4214-4228.

[32] Kiat, Wei-Pau, et al. "An energy efficient FPGA partial reconfiguration based micro-architectural technique for IoT applications." Microprocessors and Microsystems 73 (2020): 102966.

[33] Baek, Eunjin, et al. "STfusion: Fast and Flexible Multi-NN Execution using Spatio-Temporal Block Fusion and Memory Management." IEEE Transactions on Computers (2022).

[34] Zhang, Yuelin, et al. "TDACNN: Target-domain-free domain adaptation convolutional neural network for drift compensation in gas sensors." Sensors and Actuators B: Chemical 361 (2022): 131739.

[35] Jia, Heming, et al. "An enhanced chimp optimization algorithm for continuous optimization domains." Complex & Intelligent Systems 8.1 (2022),65-82