International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I11P201
Volume 69 Issue 11, 1-8, November, 2021

Design and Development of Microarchitecture for Dynamic IoT Communication

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 Scienceand Technology.*
[2]*Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Scienceand Technology.*

niteshgaikwad78@gmail.com, drshiyamala@veltech.edu.in

Abstract *— Nowadays, microarchitecture has utilized many digital applications to enrich the performance of the gadget. The microarchitecture is effectively applicable in the Internet of Things (IoT) to maximize communication performance by designing a specific processor. Usually, the microarchitecture in IoT is structured in a dynamic environment to handle the multiple diverse works simultaneously. But the damaged or weak processor can push the process of microarchitecture into trouble by consuming more energy. So, the awareness of microarchitecture merits and limitations is the needed assessment to select a good processor. Hence, this current article has prepared a detailed review of energy efficiency microarchitecture in IoT gadgets and their functions to accelerate communication. Several literature works were discussed with their advances and limitations in both table and graphical way. Finally, the discussion section has elaborated on the common defeats in the reviewed literature and its reason. Finally, future works have directed the following studies to improve the microarchitecture efficiency score.*

Keywords - *Internet of Things, microarchitecture, dynamic communication, frequency, processor*

I. INTRODUCTION

The edge-level computing model in IoT Reconfigurable Microarchitecture was often utilized as the feed circuit [1]. In addition, the stages in the pipeline differ based on each application's speed requirements [2]. Diverse structure pipelines examined various speed measurement schemes to find the suitable module in edge-level IoT [2]. The approaches based on pulse-signaling have been transmitted the attributes like word and binary instead of modulated bits [3]. Hence, the key process for executing this process is the encoding module [4]. The main concern of energy consumption in IoT applications is communication bandwidth and response time; these elements are considered overhead of cloud computing [5]. To activate the customization of the protocol [6], the needed registers were added to update the hardware module configuration like input, output buffering, and data flow [7]. The function of the Communication interface pulsed-index architecture (CIPA) has included encoding and segmentation process while the features of the structural component have contained flags and inter-symbol delay packets [8].

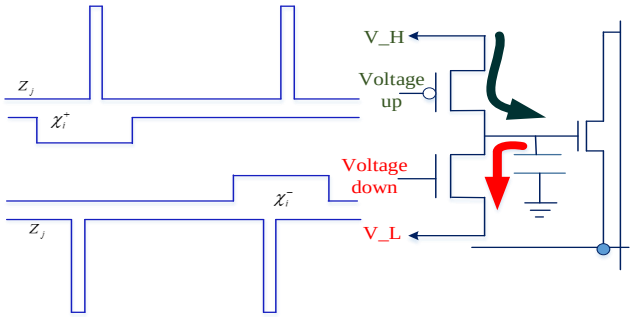

Fig.1 microarchitecture pulse

The pulsed index architecture is described in fig.1. Here, B_p, B_q, B_r are the register number of 3 bits [9], bit constant is denoted as C, B_t is the second-bit register number and Control bits of the instruction set are represented as CD [10]. The design of IS is detailed in fig.1.

For the communication application, the measure of throughput is based on the specific pipeline [11]. Hence, the pipeline in the architecture is designed before the initialization of data transmission; also, it is the finest module to advance the computing platform [12]. If the depth of the pipeline is increased, it maximizes the consumption range of power [13]. In digital applications, information technology (IT) is considered the backbone [14]. So, the different microarchitecture was utilized to enrich the communication system [15]. Hence, the reliability of the microprocessor is based on its lifetime [16]. Also, the raise of error in the microarchitecture program execution has tended to crash the application and consume maximum power [17]. The common advance of IoT applications is communication [18], where the data is broadcasted to numerous nodes [19]. Moreover, a specific microarchitecture is designed to offer communication flexibility to the IoT gadgets [20].

II. MICROARCHITECTURE FOR IOT COMMUNICATION

Nowadays, IoT technology transfers information to the entire world with a tightly interconnected specialty [36]. Here, the information is considered smart objects, having three kinds of operational functions such as gathering, handling, and allocating different data streams [37].

Moreover, the embedded system is also represented as a computational backbone like smart objects; this application is constantly growing the market requirements [38].

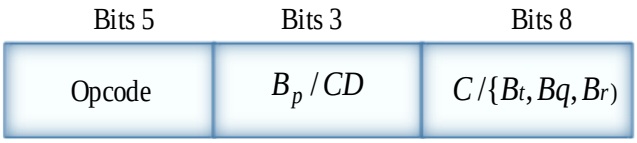

Fig.2 Instruction set (IS)

Conventionally, 32-bit microcontrollers are utilized to confirm the vital function performance when implementing the microarchitecture-based ASIC contribution [39,35]. The updating of IS of microarchitecture is detailed in fig.2. Consequently, the ASIC implementation is added to contest the small area and low power footprints necessities. Microarchitecture is termed hardware implementation based on the instruction set (IS) architecture [40]. It is the most important operation and command structure to communicate between software and hardware [41]. Here, the microarchitecture includes specific implementation requirements above the process technology and level transistors [42]. IoT communication is one of the essential applications due to the several connected nodes [43].

Moreover, Bluetooth, Wi-Fi is the IoT communication technologies based on the microarchitecture structure [44]. The specific processor of microarchitecture is the computing platform that is tailored by well-defined characteristics. Moreover, the benefit of the microarchitecture is enhancing the performance of hardware, and it will capture specific instruction sets [45]. Moreover, the enlargement microprocessor allows IoT platform to the nodes and the feasibility of developing micro architecture validated through motion control by a feedback circuit [46]. The read and write operation of microarchitecture is exposed in fig.3.

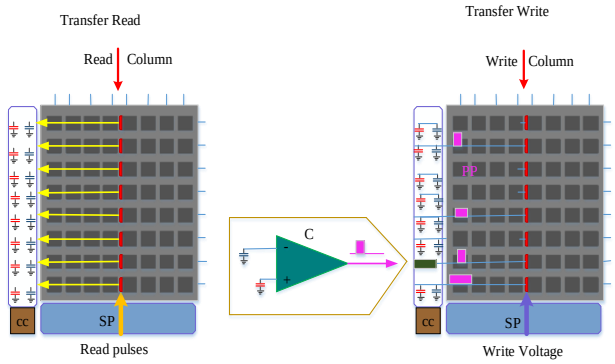

Fig.3 Read and write

Configurability is the capability for changing the configuration of the microprocessor during runtime, and it will ensure the performance of power, area, and IoT application [47]. The specific processor of microarchitecture is the best window resource to balance both Memory Level (ML) and Instruction Level (IL) based [48]. Thus, the ML has shrunken, and IL tends to lead to better performance of the processor [49]. Based on the practical time, the microarchitecture will observe the time of instruction spent on every buffer and register file [50]. In this stage, all static and dynamic register file power consumption is minimized, and the estimated instructions are high performance for improving IoT also stops the register leading requirements.

Table.1 Variable description

Components description	
RE	Raster Engine
SMP	Streaming Multi Processor
IC	Instruction Code
PB	Processing Block
T	Texture
SM	Shared Memory
IB	Instruction Buffer
WS	Warp Scheduler
DU	Dispatch Unit
RF	Register File
C	Core
M^*	Streaming Multiprocessor
T^*	Transaction processing
$\hat{L}$	Load
S^*	Special Functional Unit
CRB	Control-Register Bus
VDM	Vertex Data Master
TC	Tilling Co-processor
CSC	Combined Shading Cluster
PDM	Pixel Data Master
CDM	Compute Data Master
CGC	Coarse-Grain Scheduler
TU	Texture Unit
PC	Pixel Co-processor
2D-C	2D-Core
CMU	Core Management Unit
SF	Special Function
SBI	System Bus-Interface
SMB	System Memory Bus
MCU	Multi-Level memory Cache Unit
$16-bF^*$	16 Bit Floating-point unit
$32-bF^*$	32 Bit Floating-point unit
χ_i	Analog Voltage
SP	South Periphery
PP	Programming Pulses
V_H	Voltage high
V_L	Voltage low
C	Comparator
CC	Cubic Capacity

IoT also allows the user to interact with elegant things and gives the capability to visualize, analyze, and acquire information. The main drawback of IoT applications is the design of the fixed, embedded system. The description of the variable is exposed in table.1.

A. Dynamic microarchitecture for diverse application

Yangdong Deng *et al.* [21] have elaborated the deep study in ray procedure for the graphics application to trace the object. Initially, this procedure tracked the primitive ray in the nearest location; then, based on system requirements, accelerator frames were designed. Usually, in microarchitecture, the accelerator models are structured to diminish the complexity of the search process. Hence, the computation period taken for the dynamic series is 110 ms.

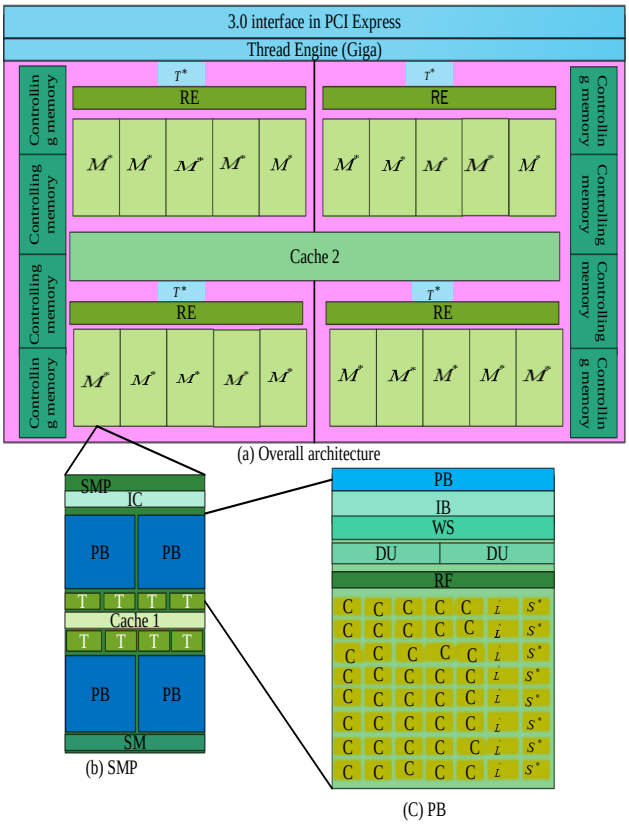

Fig.4 microarchitecture design for GPU

In the field of microarchitecture, the ray module has faced a series of threats for designing the hard such as design and energy usage. Also, to reduce the expensive score, the optimization replica has to be utilized. Moreover, the allotted bandwidth for this scheme is a maximum of 720GB/s,384 memory bus size, and 4MB cache capacity. The microarchitecture-based GU is mentioned in fig.4, and the power series is mentioned in fig.5.

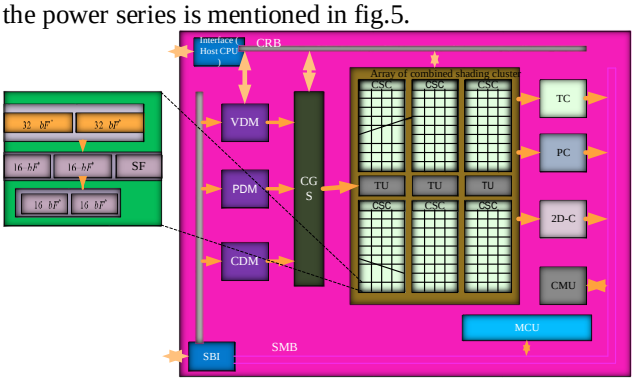

Fig.5 power series of microarchitecture

Sean Murray *et al.* [22] have developed microarchitecture for robot application and validated the proficient score of the modeled replica in the real-time environment. The reason for modeling this microarchitecture is to plan the robot motion; here, the required cores are designed in CUDA and executed in hardware FPGA platform. In addition, Siyuan Xu & Benjamin Carrion Schafer [23] have created hardware microarchitecture with reconfigurable constraints for dynamic jobs. Hence, the created paradigm has consisted of dual phases that are mapped microarchitecture, and the second phase is based on the creation of accelerators libraries. In another literature, the microarchitecture design is implemented for deep neural models, which are created for analog memories crossbar arrays. The total required energy to execute the function is 48nJ, and the crossbar arrays were developed based on 512 × 512 [24]. The merits and limitations of the analyzed methods are detailed in table.2.

Table.2 Merits and Limitations of Dynamic microarchitecture

Author	methods	merits	limitation
Yangdong Deng *et al.* [21]	Ray procedure	It has provided better outcome visualization of graphics	It requires high energy to execute the operation. It is not fixed for all hardware platform
Sean Murray *et al.* [22]	Robot prototype	Here, the microarchitecture is modeled based on robot motion planning, and good is gained from the FPGA	It is highly expensive, also consumes more energy
Siyuan Xu & Benjamin Carrion Schafer [23]	Wide level synthesis of hardware architecture	The execution time of this model is very less.	Complex in design
Hung-Yang Chang *et al* [24]	Microarchitecture for deep neural framework	To balance the workloads, cross memory was utilized	It has needed a wide measure of power
Mingcong*et al* [25]	Microarchitecture for GAN	At final, the acceleration of GAN was maximized by the newly developed microarchitecture	To complete the function, it has needed large resources

Supervised learning has been implemented for all digital applications because of their ease properties and flexibility. Furthermore, the neural model generative adversarial is effectively applied in many big data applications. So, Mingcong*et al* [25] designed an accelerating training model to maximize the neural model's rapidity and diminish the energy utilization. At final, it has reported the average speedup as 8.3X and energy efficiency as 6.2X. The performance of the Generative Adversarial Neural (GAN) model was improved with the usage of the trained accelerators. However, it has needed more resources to complete the execution.

B. Hardware Accelerators in Modern Microprocessors

The modern microprocessors are fabricated with several units that have included special functions like cryptoprocessor, arithmetic logic, and control procedures. In addition, it is proficiently utilized in many applications like web-based and cloud-based applications. Here, the main module of the recent

Microarchitecture is a hash frame; it has diminished the computational complexity. Abbas A. Fairouz*et* al [26] have created the hash-based microarchitecture to diminish the complexity of the search process. Moreover, the recorded total power is 4501 mW and 0.4143 mW dynamic powers, potentially executed in the HSPICE environment. Licheng Guo *et al.* [27] have illustrated the HM to address the problems like data overlapping and power usage. Here, the FPGA model has gained 28 × acceleration; this model is chiefly implemented for genome sequencing to solve the time consumption issues. These kinds of issues often happen because data overlap. To execute the GPU design, the diverse map level should be defined in detail. Hence, the not only held because of the wide instruction set but also met the resource usage issues since a wide range of threads has remained idle in execution. The hash-based architecture is effectively utilized in the co-structured virtual machine (VM). Also, the hash configuration in this setup is functions to hash the memory address to secure the designed microarchitecture. Moreover, to handle the multiple works, several layers are upgraded in the VM. The chief concern of the configuration administration is matching the adaptable configuration with current program phase requirements [28].

Here, the designed model performance is analyzed using diverse datasets. Brian Grayson *et al.* [29] have developed the microarchitecture for the Samsung appliance. The purpose of this design is to maximize the function accelerator and diminish the power consumption. Also, to upgrade the security function of the microarchitecture hash model is connected with the memory frame. Tosiron Adegbija *et al* [30] has designed an efficient microarchitecture for traceable IoT application to enhance communication. Hence, the developed microarchitecture has been incorporated several key attributes like 1.9 GHz frequency maximum, RAM 1TB, memory 512KB flash, and cache 32KB. In addition, IoT-based gadgets are mostly employed in unlock environments, where malicious activities are more vulnerable. This reason has degraded the integrity of the data and gadgets. Considering these issues, hash architecture was designed to afford data privacy in the IoT channel.

Furthermore, to enrich the IoT communication the

Dynamic data broadcasting environment should be developed. The reason for making a dynamic environment is to fulfill the needs of all applications

C. Dynamic IoT communication

High execution performance and less power consumption are the dual key processor structure to attain the IoT application goal. But attaining these two modules in a single microarchitecture is a complicated job. Wei-Pau Kiat et al. *[31] h*ave designed optimized instruction set in microarchitecture to advance the IoT application. Moreover, this kind of processor has been reconfigured with multiple execution cycles. Finally, it has diminished energy usage, but it needs extra time to finish the process. The used LUT is 7421, Flipflop 5223, static energy 72.50mJ, dynamic energy 16.03mJ, and total energy 88.5mJ. Hence, these energy consumption readings are gained for 20MHz frequency. The tedious module in IoT is developing the optimized embedded process. Also, the wide performance processors have been sold with a high budget. So, before choosing the processor type, the budget of every processor should be analyzed. Prasanna Kansakar and Arslan Munir [32] have made a deep study to choose the right processor for the IoT communication module. The model is named as microarchitecture configuration, and once the processor is selected, their function is analyzed by applying in the 2 standard datasets. Finally, the two-tiered processor based on heterogeneous function was chosen to enrich the communication of IoT. Moreover, the recorded speedup for the selected processor is 24.16 × accelerate on maximum frequency 75MHz, Cache c1 size 8KB, cache C2 size 256 KB, C3 cache size 4096KB, total power 0.0935W, and execution duration 332ms.

The IoT-based specific architecture has often shared knowledge about each task based on how IS was designed. Shahzad Muzaffar and Ibrahim M. Elfadel [33] have structured the microarchitecture based on the pulsed signal for the finest data sharing process among the communication devices.

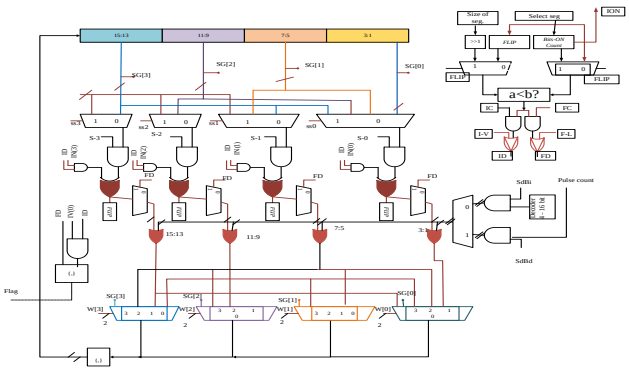

Fig.6 hardware blocks of microarchitecture

Shahzad Muzaffar &Ibrahim [34] have designed the new instruction set for a communication channel that is designed based on 2 or 3 operands at a single time. The hardware frames of microarchitecture are exposed in fig.6. Also, the structure consists of 3 registers; type 1 and type 2 registers have included 8 bits. Here, the type 1 register is utilized for the program accessible frame, and the type 2register is designed to save the configuration of the communication protocol.

III. PERFORMANCE METRICS

In microarchitecture design, clock specification is a key concern for chip-based technologies. So, the parameters like speedup and power-consuming are the key factor for the clock cycle specification. Based on the allotted clock cycle frequency limits, the latency was recorded [41]. Also, the size of the bus size was changed concerning the microarchitecture requirements. Moreover, the overall performance metrics are described in table.3.

Table.3 Overall Performance metrics

Author	methods	Cache	Bus size(b)	Computation time	Visualization platform	Hardware platform	cores	Clock rate
Yangdong Deng *et al*[21]	Ray procedure	4MB	384	110ms	CUDA	FPGA	2880	533MHZ
Sean Murray *et al* [22]	Robot protype	-	278	-	-	FPGA	-	125 MHZ
Abbas A. Fairouz*et* al [26]	Hash-microarchitecture (HM)	2MB	64	50ms	-	HSPICE	1GHz (single core)	1.39GHz
Licheng Guo *et al*[27]	HM for overlapping issues	-	16	1420.0s	CUDA	FPGA-vivado	14	250MHz
Brian Grayson *et al* [29	Cache algorithm for Samsung	4096 KB		-	Exynos	Galaxy S7	2.9GHz	8.3MHz
TosironAdegbija *et al* [30]	Microarchitecture for traceable IoT	32KB	16	-	Cortex-A15,	-	4	1.9 GHz
Wei-Pau Kiat et al. *[31]*	Reconfigure microarchitecture (RM)	16MB		42.58ms	Zigbee	FPGA	-	-
Prasanna Kansakar and Arslan Munir [32]	Two-tiered Microprocessor(2TM)	75KB	-	332ms	Benchmark dataset	cycle-accurate simulator	1	-

In microarchitecture, the memory cache size is the significant parameter to advance the performance of gadget applications. Also, the acceleration of software and hardware module is based on the cache size. In the processor module, if it downs the accelerator ten, the entire process gets denied. It has tended to cause device failure and cost complexity because the components in microarchitecture are high in cost. Once the components get damaged, there is very little possibility to reuse again; the damaged components have tended to cause device death in an unrespectable manner.

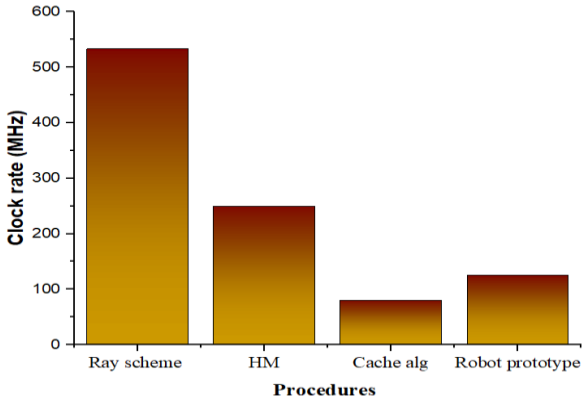

Fig.7 Comparison of different methods clock rate

The comparison of clock rate is described in fig.7; here, the ray scheme has gained a high clock rate while comparing other models like HM, cache algorithm, and robot prototype.

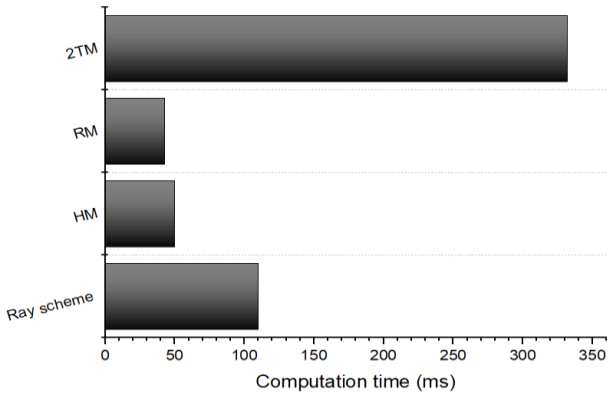

Fig.8 Comparison of different methods computation time

In the electronic system, the performance of the gadgets is based on computation time. There are 2 reasons for the long-running time duration: large memory size and multiple tasks. The comparison of time consumption is described in fig.8; here, the approach RM has attained the finest result by gaining less running time.

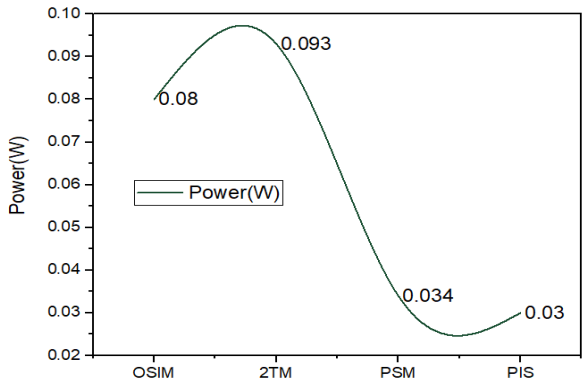

Fig.9 power consumption

The consumption of power and energy always depends on device coefficients. Hence, the validation of power consumption with diverse approaches is illustrated in fig.9. Here, the method PIS has consumed less power while comparing other models.

IV. DISCUSSION

A diverse kind of microarchitecture was analyzed in IoT communication, and several processors have met some of the common issues like high power consumption and less searching space. Also, some architecture has been designed with a large memory cache, but it takes more time to complete the process. Hence, the overall summary of this review is elaborated in fig.10.

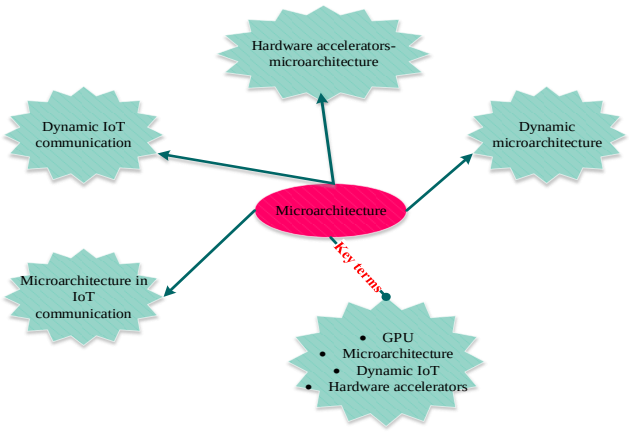

Fig.10 review summary

At the same, there are many merits with the use of microarchitecture in digitations applications. Hence, the key devices are device acceleration and easy to carry; however, choosing the wrong process has provided the temporary solution of gadget application. In the future, designing the optimized neural network has afforded the finest results while comparing the present techniques.

V. CONCLUSION

All digital applications are advanced with the usage of microprocessors with a specific processor for a particular application. But causing damage in the microarchitecture component has affected the entire system by consuming large power and resources. So the schedule of the program is carefully designed with incorporating all those parameters. During the program accessible, if one parameter is not included, that function can't be updated in the microarchitecture. The present review study has revealed that the microarchitecture processor often suffers from high energy consumption and large memory space.

Furthermore, the data overlapping is caused because of less cache memory with multiple data. So, in the future, designing the optimization approach in the microarchitecture framework will help reduce memory usage and power consumption. Also, before designing the microarchitecture, the need of the application should be identified to find the suitable processor and cache sizes. Hence, to schedule the application requirements, one of the neural frameworks will be structured.

REFERENCES

[1] Wang, Xi-Xi, et al., Assembling nano–microarchitecture for electromagnetic absorbers and smart devices., Advanced Materials 32(36) (2020) 2002112.

[2] Whittier, D. E., et al., Guidelines for the assessment of bone density and microarchitecture in vivo using high-resolution peripheral quantitative computed tomography., Osteoporosis International 31 (2020) 1607-1627.

[3] Piccoli, Alessandra, et al., Sclerostin regulation, microarchitecture, and advanced glycation end-products in the bone of elderly women with type 2 diabetes., Journal of Bone and Mineral Research 35(12) (2020) 2415-2422.

[4] Whittier, Danielle E., et al., Sex-and site-specific reference data for bone microarchitecture in adults measured using second-generation HR-pQCT., Journal of Bone and Mineral Research 35(11) (2020) 2151-2158.

[5] Pecci, Raffaella, et al., 3D printed scaffolds with random microarchitecture for bone tissue engineering applications:

Manufacturing and characterization., Journal of the mechanical behavior of biomedical materials 103 (2020) 103583.
[6] Han, Xuequan, et al., Association between knee alignment, osteoarthritis disease severity, and subchondral trabecular bone microarchitecture in patients with knee osteoarthritis: a cross-sectional study., Arthritis Research & Therapy 22(1) (2020) 1-11.
[7] Kumar, Chanchal, et al., Post-silicon microarchitecture., IEEE Computer Architecture Letters 19(1) (2020) 26-29.
[8] Shrestha, Rahul., A Multiple-Radix MAP-Decoder Microarchitecture and Its ASIC Implementation for Energy-Efficient and Variable-Throughput Applications., IEEE Transactions on Very Large Scale Integration (VLSI) Systems 29(1) (2020) 65-75.
[9] Reyes, Karen J. Campoverde, et al. ,Bone density, microarchitecture and strength estimates in white versus African American youth with obesity., Bone 138 (2020) 115514.
[10] Antonov, Alexander, Pavel Kustarev, and Sergey Bikovsky. ,MLIP Cores: Designing Hardware Generators with Programmable Microarchitectural Mechanisms., 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, (2020).
[11] Sepúlveda, Johanna., Secure Cryptography Integration: NoC-Based Microarchitectural Attacks and Countermeasures., Network-on-Chip Security and Privacy (2021) 153.
[12] Wang, Lei, et al., Wpc: Whole-picture workload characterization across intermediate representation, isa, and microarchitecture., IEEE Computer Architecture Letters (2021).
[13] Mao, Yuxiao, Vincent Migliore, and Vincent Nicomette. ,REHAD: Using Low-Frequency Reconfigurable Hardware for Cache Side-Channel Attacks Detection., 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, (2020).
[14] Antonov, Alexander, Pavel Kustarev, and Sergey Bikovsky. ,MLIP Cores: Designing Hardware Generators with Programmable Microarchitectural Mechanisms., 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, (2020).
[15] Găitan, VasileGheorghiță, and IonelZagan., An Overview of the nMPRA and nHSE Microarchitectures for Real-Time Applications., Sensors 21(13) (2021) 4500.
[16] Shashidhara, H. R., et al., Design and Implementation of Argo NI-NoC Micro-architecture for MPSoC Using GALS Architecture., Emerging Trends in Electrical, Communications, and Information Technologies. Springer, Singapore, (2020) 451-463.
[17] Semal, Benjamin, et al., A Study on Microarchitectural Covert Channel Vulnerabilities in Infrastructure-as-a-Service., International Conference on Applied Cryptography and Network Security. Springer, Cham, (2020).
[18] 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.
[19] Antonov, Alexander, and Pavel Kustarev. ,Strategies of Computational Process Synthesis—a System-Level Model of HW/SW (Micro) Architectural Mechanisms., 2020 9th Mediterranean Conference on Embedded Computing (MECO). IEEE, (2020).
[20] Jiao, Qiang, et al. ,Design of a Convolutional Neural Network Instruction Set Based on RISC-V and Its Microarchitecture Implementation., International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, (2020).
[21] Deng, Yangdong, et al., Toward real-time ray tracing: A survey on hardware acceleration and microarchitecture techniques., ACM Computing Surveys (CSUR) 50(4) (2017) 1-41.
[22] Murray, Sean, et al. ,The microarchitecture of a real-time robot motion planning accelerator., 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, (2016).
[23] Xu, Siyuan, and Benjamin Carrion Schafer., Approximate reconfigurable hardware accelerator: Adapting the micro-architecture to dynamic workloads., 2017 IEEE International Conference on Computer Design (ICCD). IEEE, (2017).
[24] Chang, H. Y., Narayanan, P., Lewis, S. C., Farinha, N. C., Hosokawa, K., Mackin, C., ... & Burr, G. W., AI hardware acceleration with analog memory: Microarchitectures for low energy at high speed. IBM Journal of Research and Development, 63(6) (2019) 8-1.
[25] Song, Mingcong, et al., Towards efficient microarchitectural design for accelerating unsupervised gan-based deep learning., 2018 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE, (2018).
[26] Fairouz, Abbas, et al., Hardware Acceleration of Hash Operations in Modern Microprocessors., IEEE Transactions on Computers (2020).
[27] Guo, Licheng, et al. ,Hardware acceleration of long read pairwise overlapping in genome sequencing: A race between fpga and gpu., 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). IEEE, (2019).
[28] Smith, James E., and Ashutosh S. Dhodapkar. ,Dynamic microarchitecture adaptation via co-designed virtual machines., 2002 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 02CH37315). IEEE, 1 (2002).
[29] Grayson, Brian, et al. ,Evolution of the Samsung Exynos CPU microarchitecture., 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA). IEEE, (2020).
[30] Adegbija, Tosiron, et al. ,Enabling right-provisioned microprocessor architectures for the internet of things., ASME International Mechanical Engineering Congress and Exposition.. American Society of Mechanical Engineers, 57571 (2015).
[31] Kiat, Wei-Pau, et al., An energy-efficient FPGA partial reconfiguration based micro-architectural technique for IoT applications., Microprocessors and Microsystems 73 (2020) 102966.
[32] Vitullo, Francesco, et al., Low-complexity link microarchitecture for mesochronous communication in networks-on-chip., IEEE Transactions on Computers 57(9) (2008) 1196-1201.
[33] Kansakar, Prasanna, and Arslan Munir. ,Selecting Microarchitecture Configuration of Processors for Internet of Things., IEEE Transactions on Emerging Topics in Computing 8(4) (2018) 973-985.
[34] Muzaffar, Shahzad, and Ibrahim M. Elfadel., A Domain-Specific Processor Microarchitecture for Energy-Efficient, Dynamic IoT Communication., IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27(9) (2019) 2074-2087.
[35] Adegbija, Tosiron, et al., Microprocessor optimizations for the internet of things: A survey., IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 37(1) (2017) 7-20.
[36] Cheikh, Abdallah, et al., The microarchitecture of a multi-threaded RISC-V compliant processing core family for IoT end-nodes., International Conference on Applications in Electronics Pervading Industry, Environment and Society. Springer, Cham, (2017).
[37] Soliman, Wasim Ghder, et al., Reconfigurable Microarchitecture-Based PMDC Prototype Development for IoT Edge Computing Utilization., IEEE Sensors Journal 21(2) (2020) 2334-2345.
[38] Chen, Yajing, et al., A low power software-defined-radio baseband processor for the Internet of Things., IEEE international symposium on high-performance computer architecture (HPCA). IEEE, (2016).
[39] Limaye, Ankur, and TosironAdegbija., HERMIT: A benchmark suite for the internet of medical things., IEEE Internet of Things Journal 5.5 (2018) 4212-4222.
[40] Saini, Harpreet Singh, and R. D. Daruwala. ,Human-machine interface in the internet of things system., International conference on computing communication control and automation (ICCUBEA). IEEE, (2016).
[41] Pereira, Pablo Punal, et al. ,Enabling cloud connectivity for mobile internet of things applications., IEEE seventh international symposium on service-oriented system engineering. IEEE, (2013).
[42] Rahmani, Amir-Mohammad, et al. ,Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems., 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). IEEE, (2015).
[43] Satapathy, Lalit Mohan, Samir Kumar Bastia, and Nihar Mohanty. ,Arduino based home automation using Internet of things

(IoT)., International Journal of Pure and Applied Mathematics 118(17) (2018) 769-778.

[44] Ayadi, Amira, and Salma Sassi. ,Privacy in the age of the internet of things: Challenges and prospects., 2016 Global Summit on Computer & Information Technology (GSCIT). IEEE, (2016).

[45] Maarala, AlttiIlari, Xiang Su, and Jukka Riekki. ,Semantic reasoning for context-aware Internet of Things applications., IEEE Internet of Things Journal 4(2) (2016) 461-473.

[46] Liu, Zhiguo, et al. ,Saving energy on processor micro-architecture level for big data stream mobile computing., 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC). IEEE, (2017).

[47] Jain, Saurabh, Longyang Lin, and Massimo Alioto. ,Dynamically adaptable pipeline for energy-efficient microarchitectures under wide voltage scaling., IEEE Journal of Solid-State Circuits 53(2) (2017) 632-641.

[48] Reaz, Mamun Bin Ibne, Md Shabiul Islam, and Mohd S. Sulaiman. ,A single clock cycle MIPS RISC processor design using VHDL., ICONIP'02. Proceedings of the 9th International Conference on Neural Information Processing. Computational Intelligence for the E-Age (IEEE Cat. No. 02EX575). IEEE, (2002).

[49] Mullins, Robert, Andrew West, and Simon Moore. ,The design and implementation of a low-latency on-chip network., Asia and South Pacific Conference on Design Automation,. IEEE, (2006).

[50] Cavallaro, Joseph R., and Mani Vaya. ,Viturbo: a reconfigurable architecture for Viterbi and turbo decoding., IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). IEEE, 2 (2003).