Turkish Online Journal of Qualitative Inquiry
Volume 12, Issue 9, November 2021: 8182-8190

Research Article

Design of Parallel Memory Allocation using Error Resilient Ternary Content-addressable Memory for Fast Error Correction

Krishnaveni Bukkapatnam[1*], Jaikaran Singh[2]

Abstract

This article addresses the implementation of error resilient ternary content-addressable memory (ER-TCAM) based parallel memory allocation (PMA) systems with error resilient properties, which can capable of detecting and correcting the errors during the parallel data allocation. This technique uses simple, single-bit parity for fault detection which has a minimal critical path overhead. Synchronization issues generated during the parallel read, write operations are effectively minimized by using the priority circuit with crossbar switching. Further, the concept of PMA is implemented through data transfer between main memory to slave devices and vice-versa. The simulation results shows that the proposed ER-TCAM based PMA resulted in superior performance as compared to the conventional approaches.

Keywords: *Parallel memory allocation, Ternary content-addressable memory, Error detection and correction, static random-access memory, speed.*

---

[1]Research Scholar, Department of Electronics and Communication Engineering, LNCT University, Bhopal, Madhya Pradesh, India. E-mail: bukkapatamkrishna@gmail.com

[2]Professor, Department of Electronics and Communication Engineering, LNCT University, Bhopal, Madhya Pradesh, India.

Received: 19.09.2021, Accepted: 30.10.2021

Introduction

FPGAs have become the de facto implementation platform for a range of computer vision applications [2] due to advancements in FPGA technology [1]. Several methods, such as stereo matching [3], are not suitable to real-time processing on general-purpose processors and are better implemented in hardware [4]. The lack of a comprehensive, one-size-fits-all hardware pipeline for the computer vision domain [5] motivates the use of FPGAs in a variety of computer vision scenarios, particularly in applications where processing should be done in situ, such as smart cameras [6], where FPGAs embed data acquisition, processing, and communication subsystems. The emergence of High-Level Synthesis (HLS) tools, which enable FPGA design inside existing software design settings, has expedited FPGA adoption by the computer vision field in recent years. TCAMs are a sort of memory in which the incoming value is compared to the ones stored in parallel and the address of the highest priority matched one is returned. CAMs are more sophisticated and power-hungry than basic memories like SRAMs [7] due to the comparison logic. The memory is known as a TCAM and is considerably more sophisticated when the rules stored in it contain bits specified as "x" or wildcard, i.e., they match both a zero and a one. TCAMs can be used as stand-alone devices or as blocks within networking ASICs [8]. Hardwired TCAM blocks do not make sense to include in FPGAs since FPGAs are used in many applications other than networking, and such TCAM blocks are not employed in them. TCAMs are instead simulated on the FPGA using the logic and memory resources [9]. Several options have been offered in this regard. One possibility is to store the rules and logic for the comparisons in the FPGA's flip-flops. In terms of the TCAM size that can be implemented, this technique has limited scalability. Another option is to replicate the TCAM using the memory blocks included into the FPGA. LUTRAMs [10] or Block RAMs (BRAMs) are two forms of internal memory that can be utilized for it. On the FPGA, LUTRAMs are a unique type of LUT that may also be set to operate as a tiny RAM. Only a small percentage of FPGA LUTs can be set as LUTRAMs. TCAM solutions based on LUTRAMs have a lower per-bit cost, but the number of LUTRAM bits accessible inside the FPGA is often significantly fewer than the number of BRAM bits [11].

Additionally, various authors focused on implementation of the hybrid PMA architectures to overcomes the drawbacks of both SMA and DMA approaches. In [12] authors focused on implementation of Huffman encoding based PMA (HE-PMA), which is capable of detecting and correcting the multiple errors at a time. But this method is developed using SRAM modules, so computational complexity of PMA is increasing as the memory size and number

of slaves are increases. In [13] authors implemented the resource aware memory allocation (RAMA) for deep learning applications. This method is focused on resource constraint environment, which is achieved by using DRAMs. By using this memory allocation, performance speed of the deep learning accelerators is increased, but this method is suffering with the high-power consumption. In [14] authors focused on implementation of power minimization based optimized memory allocation (OMA) for image processing applications. This methos is focused on implementation on FPGAs by using HLS mechanism. This method is effectively used to perform the parallel processing among various frames. But this method is suffering with the synchronization issues. In [15] authors introduced the Hyper switching memory allocation (HSMA), it is implemented by using DRAM and non-volatile memories for high-speed applications. But this method is suffering with the complexity in the data transfer and also consuming the more area, delay and power properties. In order to reduce resource utilization and power consumption, this paper focuses on PMA-based on-chip memory resources. The suggested approach is utilized to generate on-chip memory architectures using HLS and Hardware Description Languages (HDL) designs, reducing FPGA memory resource utilization and power consumption for a variety of applications.

Proposed ER-TCAM based PMA

As the VLSI technologies are further developed, multiple numbers of memories are integrated into a single system and forming as the hybrid memories. These are giving effective solutions in several fields of applications, such as video processing and image processing-based processors, digital communication processors and consumer electronics. These systems are very scalable, requiring enormous parallel connectivity and sometimes stringent timescales. Thus, to achieve these requirements in the real time applications, PMA is an effective package-oriented communication structure that is highly flexible, scalable and parallel to satisfy these applications communication requirements. Furthermore, the slave devices require the PMA in real-world applications, which needs to be done in high-speed manner.

Figure 1 presents the proposed PMA architecture for sharing data to slave devices from the main memory. These PMA architectures include SRAM based main memory, TCAM based cache memory, and network interface, which are connected to the processors and memories through the processing elements, respectively. Here, cache memory is divided into two sub memories, they are data-TCAM and Address-TCAM. Here, the data-TCAM is used to store the output from the main memories in the temporary manner and address-TCAM is used to store the main memory addresses and slave address in the temporary manner. Here, usually

main memories are consisting of SRAM based buffers, crossbars and control logics and they transfers. The conventional cache memories are implemented with the standard SRAM prototypes. But they are suffering with low memory reading and writing operations during the parallel allocation of data. Conventional SRAM based PMA are suffering with the high error rates, which causes increment in power, area and delay consumptions for reducing the errors. Further, the area, power and delay properties of conventional SRAM based PMA are also increasing with the greater number of parallel memory allocations. Thus, this work introduces the TCAM based PMA schemes, which is prominent solution to the HLS scheme.

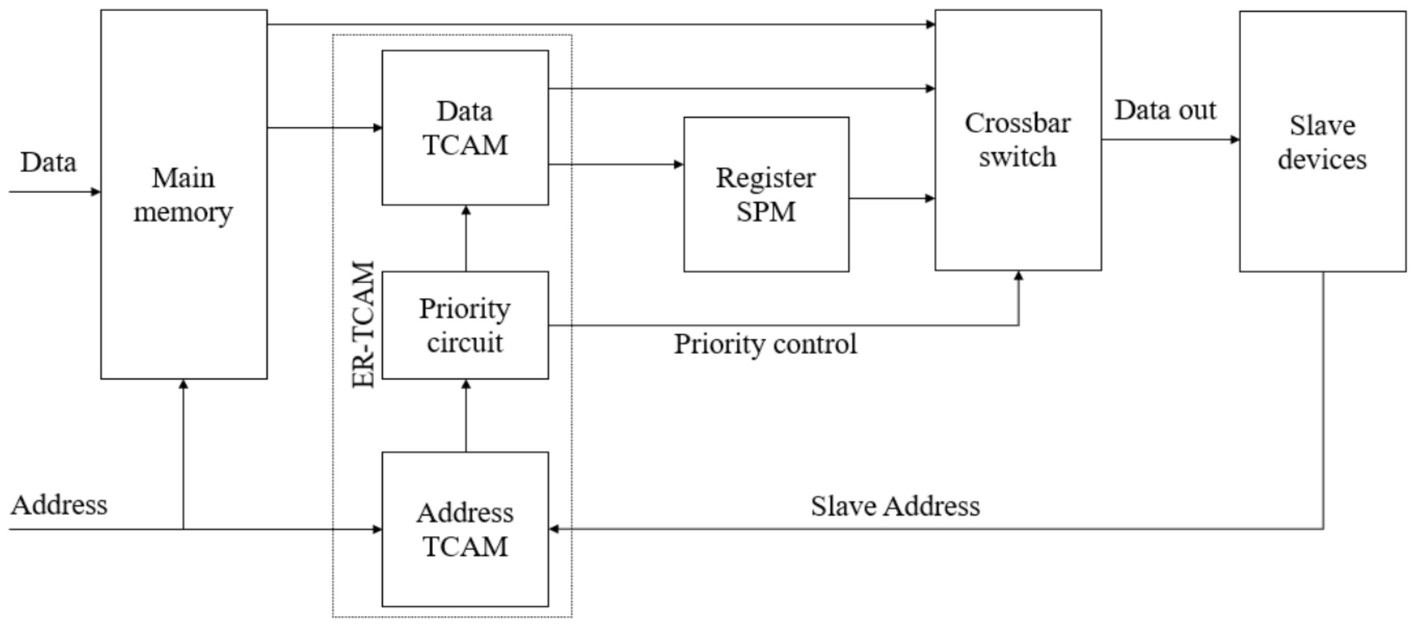

Figure 1: PMA between main memory to the slave devices.

Proposed ER-TCAM Architecture

The proposed ER-TCAM error discovery engineering is displayed in Figure 2. The error recognition calculation of ER-TCAM is displayed in Table 1.

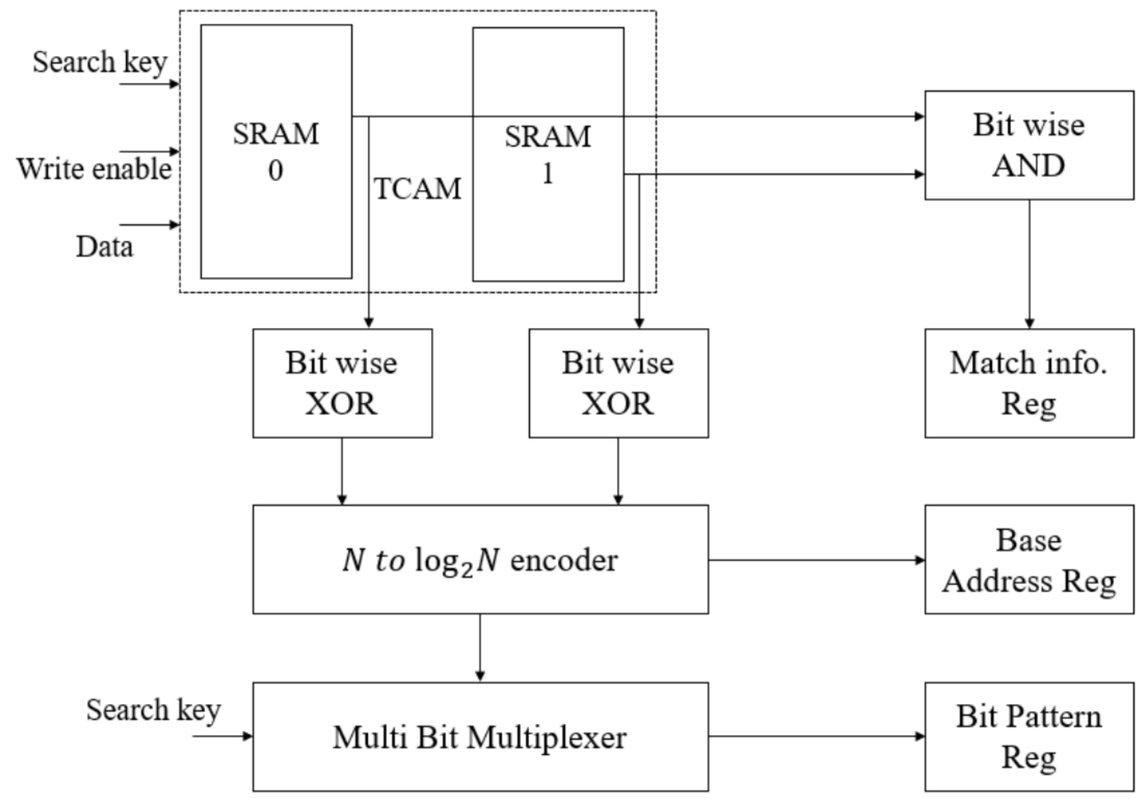

Figure 2. Proposed ER-TCAM architecture for error detection.

The addresses of the SRAM words read are EX-ORed to get an error signal when an information search key is utilized for query. The error signals from the TCAM plan's N SRAMs

are encoded to give a log2N-bit error code that interestingly distinguishes each defective SRAM.

Table 1. Error detection algorithm of ER-TCAM.

Input: Search key, write enable (WE), data **Output:** Base address, error bit patterns and matching info register values.
Step 1: ER-TCAM stores the data along with ever parity symbol based on search key.
Step 2: Apply the write enable, data and search key to the individual SRAMs of TCAM, where search key is act as the address.
Step 3: Perform the XOR operation between outputs from SRAMs, which generates the error signal.
Step 4: Apply the error signal to N to $\log_2 N$ encoder, which generates "base address" based on error syndrome. Here, if syndrome is zero no error is presented, else data is suffering with the error. Finally, address of the error data is treated as the base address.
Step 5: Apply the error signal to N to $\log_2 N$ encoder, which generates the "error bit patterns" with respect to original search key based on even parity probabilities.
Step 6: Bit wise AND operation is performed between SRAM stored data, which generates the "matching info register values".

Each cycle, the MOD-D counter makes another series of log2D bits. The SRAM address is executed with the goal that the main address of the SRAM ID alludes to the beginning of the initial sub-block in SRAM, while the lower log2D address of the counter pick SRAM words in the sub-block. The AGU then, at that point, approaches each of the paired encoded words in the TCAM table's important segment. The read TCAM words are coordinated with the C-address example to create a match bit for every cycle as shown in Figure 3, taking D clock cycles to figure the match bits and related equality bit, framing the ECV. The read/compose regulator sends a significant level compose empower sign to the initial SRAM, permitting the processed ECV to be composed over the harmed SRAM word.

The ER-TCAM gives search exercises all through the error revision process since SRAMs carrying out the TCAM work are available for query tasks. These SRAMs are arranged by the ER-TCAM as a fundamental double port RAM, which peruses and writes in equal at a similar clock cycle. The error remedying technique absolutely covers the inquiry activities in the ER-

TCAM after the ECV is determined and composed by means of the compose port of SRAM. Albeit a delicate error can happen in a SRAM putting away a paired encoded TCAM table, its error event likelihood is extremely low when contrasted with SRAMs accomplishing TCAM because of its unassuming size. In spite of this, the ER-TCAM might protect SRAM by utilizing ECCs to store the twofold encoded TCAM table at a low memory and errors also reduced.

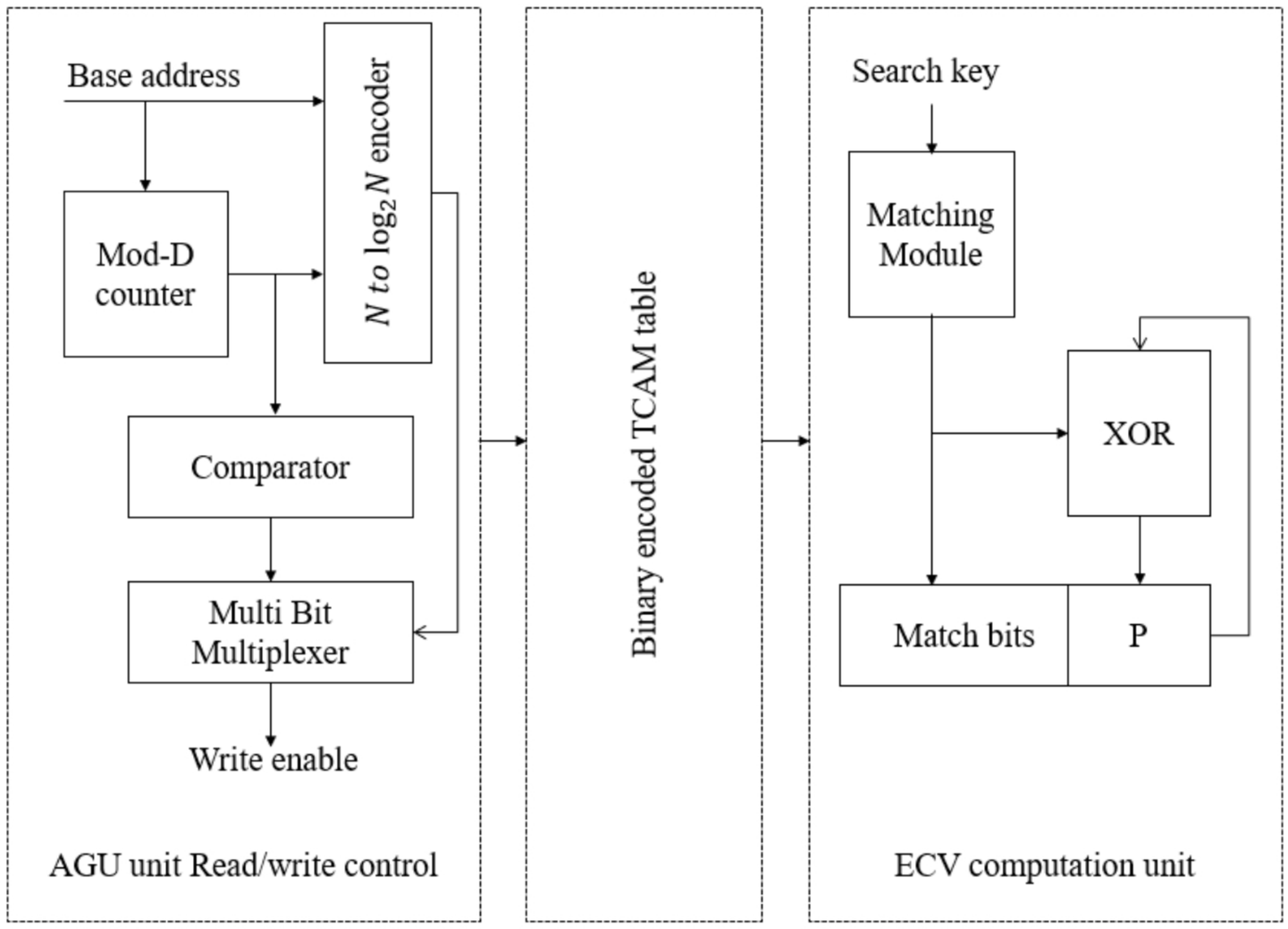

Figure 3. Proposed ER-TCAM error-correction module.

Simulation results

This section gives the detailed simulation and synthesis analysis of proposed ER-TCAM based PMA in comparison with several benchmark approaches, respectively. Throughout the simulations are carried out on Xilinx ISE software environment with FPGA prototyping and quantitative analysis is performed by measuring the various parameters like power, delay and area.

Quantitative results

This section compares the performance of proposed method with the several state of art PMA, DMA, SMA approaches with respect to the slice registers, look up tables (LUT), LUT-flip-flops (LUT-FF) s, delay, and power consumptions. Table 2 compares the performance of the proposed PMA with conventional PMA methods such as OMA [12], RAMA [13], HE-PMA [14] and HSMA [15] approaches. The conventional approaches resulting in the higher area, delay, and power consumption due to improper synchronization between various memories. Thus, the failure in synchronization resulting in the higher path delays, which makes the routed to deliver the data in undesirable manner. So, the proposed PMA equipped with ER-

TCAM block to maintain the routes congestion and resulted in better reading, writing performance.

Table 2. Performance comparison of various PMA approaches.

Method	Slice registers	LUTs	LUT-FF	Delay(nS)	Power(W)
OMA [12]	687	537	483	63.92	3.93
RAMA [13]	524	495	363	54.28	2.24
HE-PMA [14]	481	365	343	38.82	1.29
HSMA [15]	428	327	226	22.34	0.93
Proposed PMA	310	235	156	0.793	0.065

Conclusion

This article is focused on design and implementation of PMA architecture with ER-TCAM based error detection and correction properties. The proposed method is implemented with respect to the slave to main memory data sharing and vice-versa. The errors generated in the system are effectively reduced by using priority circuit, which was also used to introduce the various priorities to the slave addresses to solve the read, write synchronization issues. Further, state preserving mechanism was introduced to store the data in temporary registers for immediate operations. In addition, separate architectures were developed for slave to main memory writing and slave to main memory reading, which is used to analyze the process of PMA in end-to-end manner. The simulation results showed that the proposed PMA resulted in superior performance as compared to the SMA, DMA and PMA approaches for all area, power, and delay parameters. This work can be extended to implement the hybrid PMA system with novel routing algorithms to improve the data transfer speed between master to slave devices.

References

[1]. Chou, C.H.; Severance, A.; Brant, A.D.; Liu, Z.; Sant, S.; Lemieux, G.G. VEGAS: Soft Vector Processor with Scratchpad Memory. In Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, USA, 27 February–1 March 2011; pp. 15–24.

[2]. Liang, Tingyuan, et al. "HI-DMM: High-performance dynamic memory management in high-level synthesis." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.11 (2018): 2555-2566.

[3]. Giamblanco, Nicholas V., and Jason H. Anderson. "A dynamic memory allocation library for high-level synthesis." *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019.

[4]. Ahmed, Tanvir, and Johannes Maximilian Kühn. "Accuracy-Aware Memory Allocation to Mitigate BRAM Errors for Voltage Underscaling on FPGA Overlay Accelerators." *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 2020.

[5]. Wang, Guohua, Song Liu, and Jiangtao Sun. "A dynamic partial reconfigurable system with combined task allocation method to improve the reliability of fpga." *Microelectronics reliability* 83 (2018): 14-24.

[6]. Wang, Qinggang, et al. "GraSU: A Fast Graph Update Library for FPGA-based Dynamic Graph Processing." *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 2021.

[7]. Lee, Hayoung, et al. "Dynamic built-in redundancy analysis for memory repair." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.10 (2019): 2365-2374.

[8]. Khajekarimi, Elyas, Kamal Jamshidi, and Abbas Vafaei. "Integer linear programming model for allocation and migration of data blocks in the STT-RAM-based hybrid caches." *IET Computers & Digital Techniques* 14.3 (2020): 97-106.

[9]. Shekarisaz, Mohsen, et al. "MASTER: Reclamation of Hybrid Scratchpad Memory to Maximize Energy Saving in Multi-core Edge Systems." *IEEE Transactions on Sustainable Computing* (2021).

[10]. Takami, Alireza Lotfi, et al. "RIDE: Energy Efficient Data Allocation on Compound Racetrack-SRAM Scratchpad Memory for Real-Time Embedded Systems." *2020 CSI/CPSSI International Symposium on Real-Time and Embedded Systems and Technologies (RTEST)*. IEEE, 2020.

[11]. Zhang, Rui, et al. "SRAM stability analysis and performance–reliability tradeoff for different cache configurations." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28.3 (2020): 620-633.

[12]. Shao, Zhenyu, et al. "A High-Throughput VLSI Architecture Design Of Canonical Huffman Encoder." *IEEE Transactions on Circuits and Systems II: Express Briefs* (2021).

[13]. Nguyen, Duy Thanh, et al. "ShortcutFusion: From Tensorflow to FPGA-based accelerator with reuse-aware memory allocation for shortcut data." *arXiv preprint arXiv:2106.08167* (2021).

[14]. Garcia, Paulo, et al. "Optimized memory allocation and power minimization for FPGA-based image processing." Journal of Imaging 5.1 (2019): 7.

[15]. Prabhu, M., Har Narayan Upadhyay, and R. Vijay. "Hyper switching memory utilization on hybrid main memory for improved task execution and reduced power consumption." *Microprocessors and Microsystems* 72 (2020): 102891.