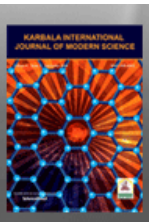

Karbala International Journal of Modern Science

Volume 8 | Issue 2 Article 7

# Partial Pseudo-Random Hashing for Transactional Memory Read/Write Data Processing and Validation

G M Sridevi
*Research Scholar, JSS Academy of Technical Education (Affiliated to Visvesvaraya Technological University, Belagavi)*, sridevi.gereen87@gmail.com

Ashoka D V
*Department of Information Science and Engineering, JSS Academy of Technical Education, Bengaluru, India*, dr.dvashoka@gmail.com

B V Ajay Prakash
*VTU-RC, Belagavi, India*, ajayprakas@gmail.com

# Partial Pseudo-Random Hashing for Transactional Memory Read/Write Data Processing and Validation


## Abstract

Development of a bypass parallel processing block is one of the emerging and interesting research areas in memory read/write application domain. Many Random Number Generation (RNG) techniques have been introduced for processing the data in storage memory. But the limitations include reduced efficiency, increased computational complexity, high area consumption and higher cost. This paper presents a novel dynamic memory register with optimal XOR design based on partial pseudo-random hashing to process transactional memory read/write data. Transfer characteristics of the current are analysed based on the pseudo differential pair for proficient memory utilization. A memory window is then created and adjusted for obtaining an optimal power flow with lesser data loss. The performance of the proposed design is evaluated using different performance measures. The power consumed for processing the data using the proposed design is reduced to nearly 25% when compared to other proposed designs along with reduced component usage. Delay is reduced to 2.31ns and a 15% improvement in frequency and nearly 4% increase in throughput is seen when compared to existing methods.



## Keywords

Bypass logic, Dynamic Memory Register, Partial Pseudo-Random based Hashing, Random Number Generator (RNG), Transactional memory

RESEARCH PAPER

# Partial Pseudo-Random Hashing for Transactional Memory Read/Write Data Processing and Validation

G.M. Sridevi [a,b,*], D.V. Ashoka [c], B.V. Ajay Prakash [d]

[a] Research Scholar, JSS Academy of Technical Education (Affiliated to Visvesvaraya Technological University), Belagavi, India
[b] Department of Information Science and Engineering, SJB Institute of Technology, Bengaluru, India
[c] Department of Information Science and Engineering, JSS Academy of Technical Education, Bengaluru, India
[d] VTU-RC, Belagavi, India

**Abstract**

**Development of a bypass parallel processing block is one of the emerging and interesting research areas in memory read/write application domain. Many Random Number Generation (RNG) techniques have been introduced for processing the data in storage memory. But the limitations include reduced efficiency, increased computational complexity, high area consumption and higher cost. This paper presents a novel dynamic memory register with optimal XOR design based on partial pseudo-random hashing to process transactional memory read/write data. Transfer characteristics of the current are analyzed based on the pseudo differential pair for proficient memory utilization. A memory window is then created and adjusted for obtaining an optimal power flow with lesser data loss. The performance of the proposed design is evaluated using different performance measures. The power consumed for processing the data using the proposed design is reduced to nearly 25% when compared to other proposed designs along with reduced component usage. Delay is reduced to 2.31ns and a 15% improvement in frequency and nearly 4% increase in throughput is seen when compared to existing methods.**

*Keywords:* Bypass logic, Dynamic memory register, Partial pseudo-random based hashing, Random number generator (RNG), Transactional memory

## 1. Introduction

Hashing is a technique that generates key−value pairs and is used for fast data access. Different hashing techniques have been studied and applied extensively for data organization since early 60s. Some of the hashing techniques are static hashing, dynamic hashing techniques like linear hashing and extendible hashing, perfect hashing and so on [1]. Much work was done to enhance these techniques for improving the performance and to simplify their design [2−6]. Hashing techniques have also been extended for hardware applications in the areas of cryptography, networking, transactional memory (TM), page table translation etc. [7,8].

Memory read/write architecture with XORing for Transactional Memory has seen significant attention in the recent days [9,10]. With the development of CMOS (Complementary metal oxide semiconductor) technology, XOR based RNG designs have been used to attain sampling rates with high power efficiency and reduced area consumption in Transactional memory systems for GPU [11,12]. Various RNG designs were developed to improve the power efficiency with the help of parallel processing schemes. Sahay et al. provide a detailed review on advanced RNG/PUF (Physically Unclonable Function) methods for Non-Volatile Memory (NVM) devices [13]. Bypass logic is a kind of parallel processing scheme, in which a window can be created when the input data sequence is located in a specified range [14]. The bypass window helps in reducing the number of switching cycles required for the conversion of signals and thereby reducing

Received 11 November 2021; revised 17 February 2022; accepted 22 February 2022.
Available online 1 May 2022.

* Corresponding author at: Research Scholar, JSS Academy of Technical Education (Affiliated to Visvesvaraya Technological University), Belagavi, India.
E-mail addresses: sridevi.gereen87@gmail.com (G.M. Sridevi), dr.dvashoka@gmail.com (D.V. Ashoka), ajayprakas@gmail.com (B.V. Ajay Prakash).

https://doi.org/10.33640/2405-609X.3223

the energy consumed. The register is an essential block of optimal XOR that is used to reduce the power consumption in an efficient manner.

When compared to other XORing techniques, hashing with optimal XOR has the ability to increase the energy efficiency and is highly suitable for processing memory read/write operations [15]. In this technique, a predefined bypass window is constructed to avoid unnecessary operations by sampling the input data with respect to clock pulses [16,17]. The time required for optimal XOR conversion is considerably lower than the normal cases, when triggering a bit cycle bypass [18]. Typically, the bypass window construction can be incorporated in the register design to boost the conversion rate. Many existing works explore the design of hashing with optimal XOR in memory read/write applications. But their limitations include high computational complexity, increased cost consumption, reduced energy efficiency, and high area consumption. This research work aims at developing a dynamic memory register design based on hashing with optimal XOR for processing the memory read/write data in an efficient way.

In the proposed bypass parallel processing scheme, external reference data sequence and additional registers have been utilized for bypass detection. The proposed dynamic memory register design can be adapted with the current controller, so it has the ability to bypass the data during the alteration of regular polarity outputs. The major objectives of this paper are listed as follows:

- To design a bypass parallel processing scheme based on hashing with optimal XOR with a dynamic memory register block for memory read/write operation.
- To analyze and process the memory read/write data using a dynamic memory register.
- To study the transfer characteristics of the current based on the pseudo differential pair for proficient memory detection.
- To obtain an optimized power flow with low data sequence sensitivity for various sensing applications by adjusting the window size.
- To reduce the complexity of the block design by minimizing the number of logic gates and parallel processing power devices.
- To validate the performance of the proposed hashing with optimal XOR with dynamic memory register using different evaluation measures such as maximum operating frequency, offset data sequence and power consumption.

When compared to the models proposed for Random Number Generation, the proposed design for transactional memory read/write data processing using Optimal XOR gives better results in terms of frequency of data transfer, throughput and delay when compared to the recent advanced models presented using RNG. Area consumption is reduced with the usage of fewer components when compared to existing methods. Usage of optimal XOR improves the energy efficiency of the system. Also, the design can be easily adapted with the existing controller.

Rest of the sections present in the paper are structured as follows: Section II surveys the existing techniques and designs related to the memory read/write data processing applications. Section III provides the clear description of the proposed dynamic memory register based on hashing with optimal XOR design. Section IV evaluates the performance of the proposed methodology using different measures, and its superiority is proved in comparison with some of the state-of-the-art techniques. Finally, the paper is concluded and the future work is stated in Section V.

## 2. Related work

This section surveys the existing techniques and methodologies for enhancement of data read/write process in transactional memory.

Castro et al. proposed a novel model for Non-Volatile Hardware Transactional Memory (NV-HTM) [19]. The design involves postponing the externalization of events that are committed and pruning the committed logs. These parameters are implemented in the transactional memory to enhance the throughput of memory read and write functions. For optimal writing process, FPGA Accelerated concurrent control was proposed by Li et al. [20]. The authors integrated reachability based Optimistic Concurrency Control (ROCoCo) in Transactional memory. An innovative formalization of mainstream Concurrency Control (CC) algorithm was developed to control unnecessary blocking of memory. The aborting of memory transfer process can be resolved by Reachability Optimization (RO) method. The RO system validates a cyclic check without basis on the timestamps. This can detect and abort a transaction that cause cycles and can also validate the sequences. This extracts the pattern of message sequence to improve the performance of TM. Transactional memory can also be used for the protection of private keys against memory disclosure.

Hemattil et al. present a pseudo-random number generator based on non-linear feedback shift register (NFSR) and chaos generator which enhanced the output when compared to linear feedback shift registers and pseudo-random chaos generator [21]. A Fibonacci NFSR was used for the design. HTM was used to block leakage of information through side channels in cache memory [22]. Conflicts in memory access by multiple threads were handled by rolling back to the previous state. Parallel access was allowed by providing a memory snapshot to work on. Cloak stores sensitive data in transactional memory during a memory transaction.

In transactional memory, parallel execution of read and write operations needs to be uninterrupted. To achieve this, the address storage memory needs to be made as a random value to store the data. Billmann et al. presented a novel open-source crypto IP core that was implemented on FPGA design [23]. Here, a hardware security building block was structured for the cryptographic function, to improve the security system in IoT architecture. In this, the Random Number Generation (RNG) process mainly focused on the IP cores to enhance the secure data transmission in wireless communication system. Random number generators are used for the prevention of attacks in data transmission. For this, a Physical Un-cloneable function (PUF)-based authentication protocol was used to get the random number for key extraction [24]. This type of random key extraction using PUF concept improves the security level and reduces the attacks. The random number generator design was implemented and analyzed on CMOS [25]. In this paper, temporal majority voting (TMV)-assisted Entropy Source (ES) array segregation was processed in the tri-level hierarchical Von Neumann (VN) extraction to maximize entropy harvesting. This type of entropy estimation improves the performance of PUF model in generation of random number when compared to the traditional design. This type of PUF model was used to co-optimize the ES array for static and dynamic entropy with bias awareness.

While considering the size of components in the FPGA design architecture and to reduce the number of Look-Up Tables (LUTs) and Flip-Flops (FFs) in the random number generation, a true random number generator (TRNG) was proposed [26]. The proposed TRNG was tested on Xilinx Ultra scale XCKU040 FPGA board. This TRNG design can alternatively use jitter and metastability as seeds of entropy in FPGA model. It employs a single Phase-Locked-Loop (PLL) and three on-board primitives to measure the count of logical elements that are extracted from the summary report of implementation. This enhances the transmission speed in the range of 100 Mbps. A VLSI implementation of cryptographic system was presented based on the reversible logic gates with Linear feedback shift register (LFSR) key generation system [27]. This was implemented and tested on FPGA and ASIC platform to validate the performance in both logical block structure and the application module. The Reversible logic cryptography design (RLCD) architecture for encryption and decryption process was designed with LFSR based random key generation which enhances the cryptographic process than the traditional model of encryption and decryption process.

Zeng et al. presented an investigation of semantic features of transactional memory [28]. This was simulated and tested on transactional memory architecture in Java. Isolation and atomicity semantics of transactional features are simulated and analyzed to explore different trade-offs between ease of use and efficiency of the TM system. Unbounded Hardware Transactional Memory (UHTM) system was proposed based on the hybrid combination of DRAM and Non-Volatile Memory (NVM) [29]. This proposed structure was implemented for atomic-durable updates and for isolation in hardware transactional memory (HTM). The UHTM integrated the cache coherence protocol and address signatures to detect conflicts of data in internal memory space architecture. This enhanced the HTM by 56% compared to the traditional model and other state-of-art methods.

To address the challenge of potentially high abort rates and missing support for unbounded transactions in HTM, Piatka et al. presented a Transaction Management Unit to enable different contention management techniques [30]. The test was simulated with STAMP benchmark suite on gem5 simulator to estimate the performance of the proposed model.

Issa et al. proposed an extended hardware transactional memory system to mitigate the limitation on IBM's POWER8 architecture [31]. This was achieved by leveraging a key combination of hardware and software techniques to provide support on different execution paths. For improving the synchronization functionality of HTM memory, barrier latency and other related parameters need to be considered for better performance. Pedrero et al. proposed speculative barriers (SBs) for transactional memory in place of locks [32]. The SBs leverage HTM support to elude barriers speculatively. In this, the HTM integrates a procedure to check the speculation state in threading functionality. Using SBs, the updates are isolated to thread process.

RNG methods have been used for different applications. While there are a few models proposed for different applications using RNG, their limitations include reduced efficiency, increased computational complexity, high area consumption and higher cost. Some methods were found to give good results but they can be further enhanced to reduce the area utilized and to increase the throughput. Hash based techniques provide better performance in terms of faster access and in detection of conflicts. In this paper, we present a XOR based RNG for read/write processing in Transactional Memory which gives better performance when compared to the existing methods.

## 3. Proposed methodology

This section presents the overall description of the proposed bypass parallel processing optimal XOR partition design with its clear illustration. The main aim of this work is to develop an optimal XOR based RNG with a new register design for storing data in memory and to validate the data with reduced error rate compared to other application processes. This implementation reduces the amount of data that needs to be stored at various stages, and the amount of time essential for processing. An appropriate low power hardware architecture has been designed to implement a real time high performance and low cost optimal XOR block with the proposed decoder algorithm. This block design is highly suitable for mobile applications. Distinct data can be simulated based on the random data generation, where the sampling rate of 360 Hz is relevant to clock frequency. The overall block representation of the proposed architecture design of PPRH hashing in the transactional memory is represented in Fig. 1. The major stages comprised in the proposed register design are as follows:

- RNG design
- Hashing Model
- Data Transfer Architecture

### 3.1. RNG design

The register used in this block design is one of the essential components in the electronic block after each XOR is connected to the latches. Typically, it is

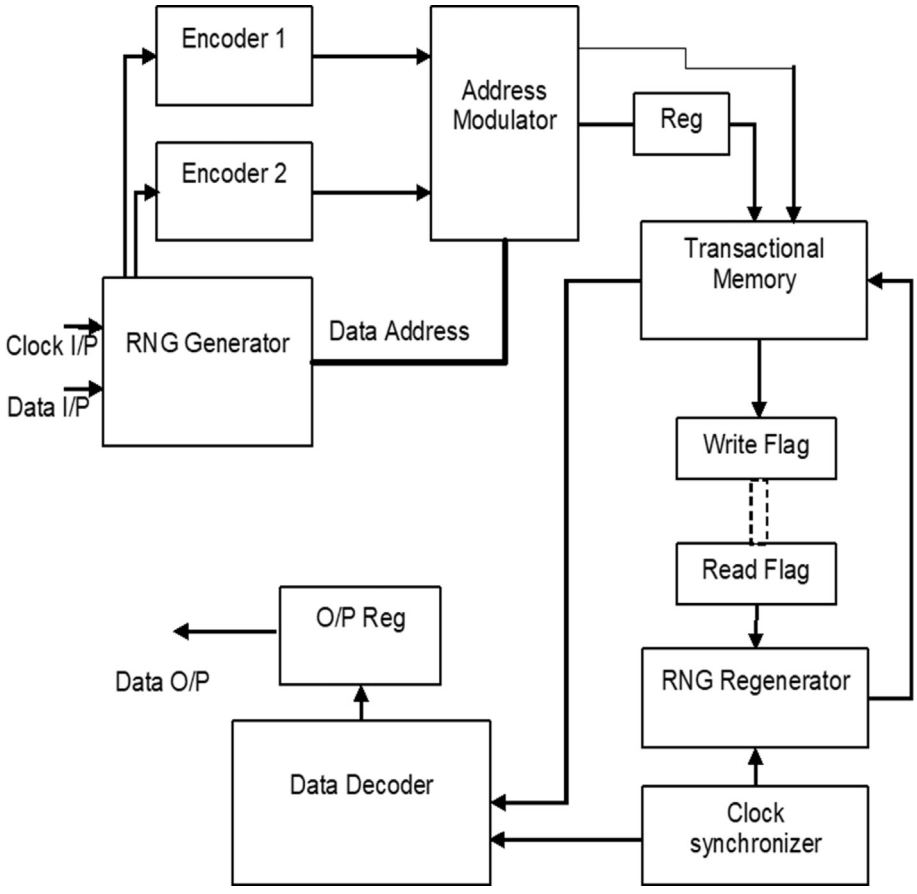


*Fig. 1. Block diagram of Proposed Hashing technique in Transactional memory.*

a major building block for most of the applications, in which the logical information has to be recovered from sequential data specifically in optimal XORs. This register is more suitable for logic blocks, where the latch provides large and fast output data. Its waveform and amplitude are independent of input data. Generally, two types of registers have been used named static and dynamic, in which each gate output is connected with flags by the static registers through a write line path. Similarly, the dynamic registers rely on the temporary storage of data values in the binaries of highly sensitive block nodes. The dynamic memory register holds the information about the memory and is represented as the index. It holds the memory information dynamically and updates with the counters and shifters to generate random number/address. Static registers are more suitable for medium speed applications and the dynamic registers are more suitable for highly sensitive block nodes. But it is highly sensitive to noise and varying sampling rates can be used for different applications. When compared to the static registers, the dynamic registers gained a significant attraction in medium to high-speed applications. Furthermore, it is energy efficient owing to the non-consumption of static current contrast. Figure 2 shows the schematic representation of the XOR and register allocation flow with corresponding output units. It comprises of the following stages: pre-amplifier, track and latch, where the pre-amplifier is larger than the input data and is sufficient to drive the logical block. The track and latch stages are used to amplify the data with the use of positive feedback loop. The major benefits behind this register design are high input sensitivity, no static power consumption, full swing outcome, and fast decision rate.

### 3.2. Hashing model

The proposed hashing with optimal XOR design is fully based on the binary search algorithm, where the partition is a logical controller block which has the responsibility to run the binary search procedure. The output of hashing with optimal XOR can be determined with respect to the register output. Hence, it has the momentous effect on improving the overall performance of optimal XOR design. The register contains n-bit optimal XOR, with two possible values for each bit, either 0 or 1. Initially, MSB can be set as 1 and other bits can be set as 0, and the logical word is converted to the sequential value via the RNG unit. Then, the output of sequential data can be inserted as the input to register, and is compared with the sampled input. Based on this outcome, the partition controller can estimate the value of MSB, where if the input is higher than the RNG output the value of MSB can

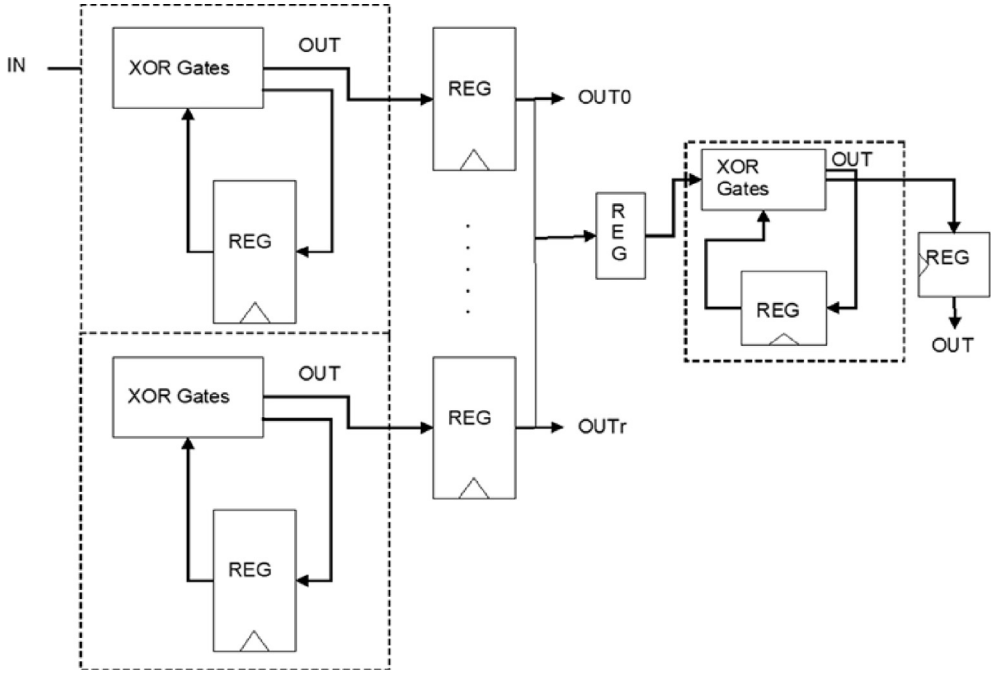


*Fig. 2. RNG register allocation design.*

be set to 1; else, set to 0. During the last cycle, the converted logical word is stored, and N+1 clock cycles could be performed for conversion. The partition is designed using a register and the controlling block for encoding the address to transactional memory, and the conversion process can be continued only when the XOR input is low. Figure 3 shows the schematic representation of hashing with optimal XOR logic. The major building blocks are dynamic latched register and inverters. Algorithm-I represents the steps for proposed hashing model for random address generation.

**Algorithm I – Random Address Generation.**

**Input:** Input Bits, $I_n$

**Output:** Random Address Points, $R_v$

**Step 1.** Initialize weight parameter as 'w'.

**Step 2.** Initialize the temporary matrix, 'q' as in equation (1).

**Step 3.** Estimate the signature pattern from the matrix as in equation (2). This can be represented as $h_q(x)$.

**Step 4.** For x = 2 to m-1loop // Loop run for 2 to 'm-1' size of 'q' matrix.
Update counter as counter ++.
Estimate the patterns *ch* and *mj* from equations (5) and (6).
Update $S_1$ and $S_2$ for each counter update.
End loop 'x'
$R_v = h_q$ (*counter*)
Let the input binary bits for the memory stream be represented as $I_n$. For the random address generation, the address should be regenerative based on the input matrix. For that, the bit stream can be normalized by using the matrix 'q' represented as (1).

$$q = \begin{bmatrix} 32'h6a09e667 \\ 32'hbb67ae85 \\ 32'h3c6ef372 \\ 32'ha54ff53a \\ 32'h510e527f \\ 32'h9b05688c \\ 32'h1f83d9ab \\ 32'h5be0cd19 \end{bmatrix} \quad (1)$$

From this matrix 'q', the sequences are updated and arranged for each counter iteration based on $S_1$ and $S_2$ parameters and for each time instant, the $h_q(x)$ matrix is updated by shifting the binary streams based on the index update. This can be represented using equation (2).

$$h_q(x) = \begin{cases} h_q(x-1) & \forall x = \{2,3,\ldots m-1\} \\ S_1, & if\,(x=1) \\ S_2, & if\,(x=5) \end{cases} \quad (2)$$

where, $S_1$ and $S_2$ are evaluated using equations (3) and (4) respectively.

$$S_1 = h_q(m) + (A \oplus B \oplus C) + (D \oplus E \oplus F) + w + k + ch + mj \quad (3)$$

$$S_2 = h_q(4) + h_q(m) + (D \oplus E \oplus F) + w + k + ch \quad (4)$$

where,

$$A = \{q[1][1:0], q[1][32:2]\}$$

$$B = \{q[1][12:0], q[1][31:13]\}$$

$$C = \{q[1][21:0], q[1][31:22]\}$$

$$D = \{q[5][5:0], q[1][31:6]\}$$

$$E = \{q[5][10:0], q[1][31:11]\}$$

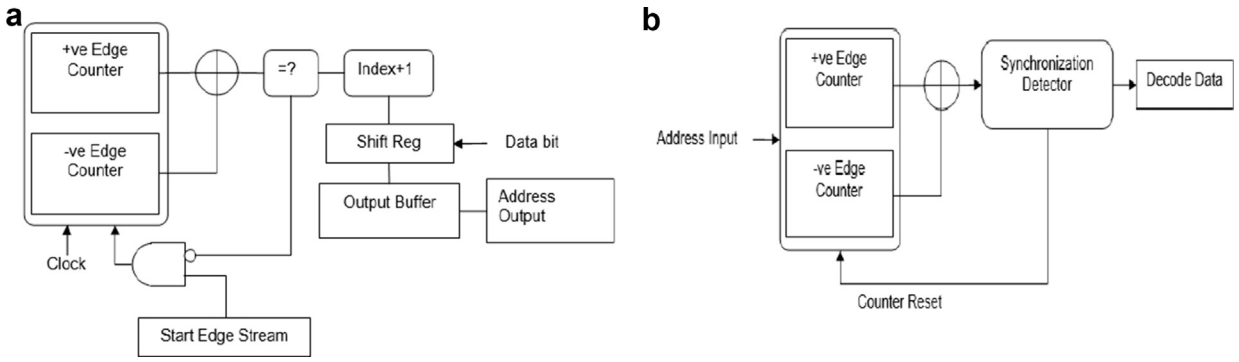


*Fig. 3. (a) Block diagram for address passing in write process. (b) Block diagram for address passing in read process.*

$$F = \{q[5][24:0], q[1][31:25]\}$$

'k' is the register value for corresponding counter index of each clock cycle. '*ch*' and '*mj*' are temporary variables that are used to calculate the pattern. *ch* and *mj* can be evaluated using equations (5) and (6) respectively.

$$ch = (q[5] \bullet q[6]) \oplus (!(q[5] \bullet q[7])) \quad (5)$$

$$mj = ((q[1] \bullet q[2]) \oplus (q[1] \bullet q[3]) \oplus (q[2] \bullet q[3])) \quad (6)$$

This cross computing generates the random address for memory storage which can be represented as $R_V$. The reverse process can regenerate the random address to read the data from address location in TM.

### 3.3. Data Transfer Architecture

Typically, RNG is mainly used to convert the logical input data into sequential output data. Logical data indicates individual bits and the sequential data represents the bit stream. The result is highly proportional to the logical value. The conversion tool acts as an interface between the logical and sequential blocks. Moreover, it provides feedback data to correct the errors and also estimates the reference data sequence during the process of conversion. During the design of RNG, the values of the supporting blocks should be determined based on the manual estimation process. It can then be adjusted further during the time of simulation, where some of the parameters can be considered with respect to the AMS technology. The architecture design process can be defined according to the AMS technology as follows:

- Arrangement of XOR blocks
- Combination of Registers and XOR.

The schematic illustration of RNG design is depicted in Fig. 4, which comprises of three stages listed below.

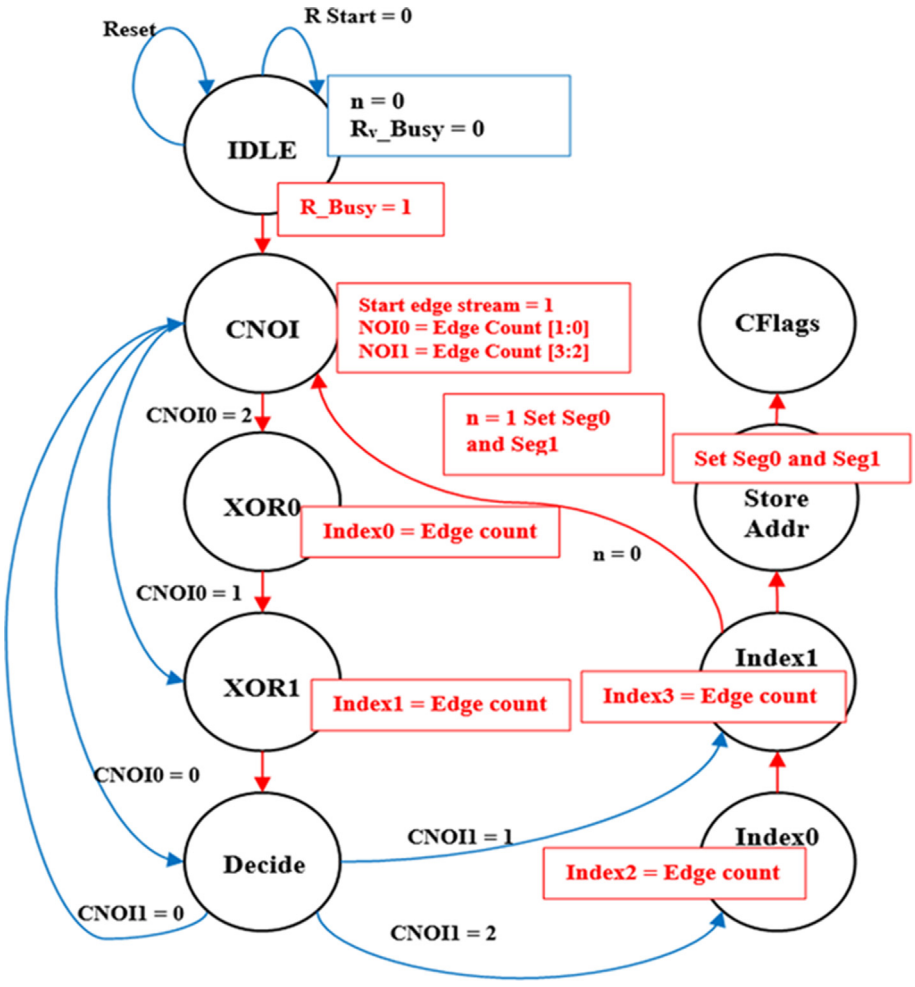


*Fig. 4. State flow for RNG generator.*

- XOR block combination
- Decision block
- Indexing to address

The major supporting components of this device are transistors. There are 2 optimal XOR blocks which are used to set the values at index positions 0 and 1 by varying the clock cycles. Based on the output, the decision block evaluates the values at index positions 2 and 3. The decision block may restart the index value evaluation if the MSB bit is set to 1. At each stage, the edge count is assigned to respective index positions. The final index values are converted to a sequential address.

## 4. Results and discussion

This section presents the simulation results and compares the results of the proposed work with existing systems of random number generation in transactional memory. The proposed design was implemented on Xilinx platform with output waveforms of CMOS technology. The proposed method is compared with the True Random Number Generator design presented in [33] for RAM memory in Behavioral Cellular Automaton Generator (CAG) and Primitive Instantiation [34] for Field Programmable Gate Arrays. The results were compared with the state-of-the-art models using the measures of frequency, supply data sequence, power, area, sampling rate and propagation delay.

Table 1 shows the comparison of the existing random number generation block in the resistive RAM design and proposed technique of RNG in Transactional Memory with respect to the number of 2-input XOR's, 3-Input XOR's and the number of Flip-Flops utilized [33]. The results show that the proposed dynamic memory register-based on hashing with optimal XOR uses fewer components when compared to the existing model of RNG.

Table 2 compares the area consumption in $\mu m^2$ in the existing [33] and proposed hashing technique with optimal XOR. For the proposed work, the power consumption is reduced to nearly ~25%. From the evaluation, it is evident that the proposed method could efficiently reduce the power and area consumption, when compared to the existing RNG models.

*Table 1. Component Utilization comparison of RNG models.*

| Gates | Utilization | |
|---|---|---|
| | RNG Resistive RAM | Proposed RNG |
| 2-input XOR | 4 | 4 |
| 3-Input XOR | 7 | 5 |
| FFs | 7 | 6 |

*Table 2. Comparison of area ($\mu m^2$).*

| Gates | Area ($\mu m^2$) | |
|---|---|---|
| | RNG Resistive RAM | Proposed RNG |
| 2-input XOR | 43.04 | 43.04 |
| 3-Input XOR | 72.62 | 51.87 |
| FFs | 118.4 | 101.48 |

*Table 3. Comparison of FF, LUT and slice count.*

| *Methods* | *Register Width* | *FFs* | *LUTs* | *Slices* |
|---|---|---|---|---|
| Behavioural CAG | 32 | 146 | 176 | 76 |
| Primitive Instantiation | 32 | 146 | 145 | 49 |
| Proposed | 32 | 137 | 135 | 47 |

*Table 4. Delay rate (ns), Frequency (MHz) and Throughput (Gbps) comparison.*

| Methods | Delay (ns) | Frequency (MHz) | Throughput (Gbps) |
|---|---|---|---|
| Behavioural CAG | 3.503 | 285.47 | 9.135 |
| Primitive Instantiation | 2.67 | 374.53 | 11.99 |
| Proposed | 2.31 | 432.9004 | 12.34 |

Table 3 illustrates the utilization count of FFs, LUTs and Slice count of the existing [34] and proposed RNG technique. From the analysis, it is proved that the proposed method provides better results when compared to the other techniques.

Table 4 shows the performance of existing methods [34] and the proposed technique of RNG in terms of delay (ns), frequency (MHz) and throughput (Gbps). The propagation delay is reduced to 2.31ns by the proposed hashing with optimal XOR design. The results show that the proposed design could efficiently reduce the delay, when compared to the other techniques. Based on the overall analysis, it is evident that the proposed dynamic memory register design using hashing with optimal XOR is more suitable and efficient for processing the memory read/write data.

## 5. Conclusions

This paper presents a new register design involving hashing with optimal XOR for processing the memory read/write operations. This methodology reduces the amount of data to be stored at various stages, and the time required for processing the data. The proposed dynamic memory register adapts with the current controller, which generates both polarity outputs and bypass trigger data during conversion. Here, a pseudo differential pair is constructed to simulate the transfer characteristics. In this design, the charging speed can be adjusted with the process variation and component mismatch along the path. Hashing with optimal XOR provides

low data sequence sensitivity, increased efficiency and low power consumption. During experimental evaluation, different metrics were used to analyze the performance of the proposed register design. From the results, it is proved that the proposed technique provides better results, when compared to the traditional techniques. In future, this work can be extended for cloud and big data applications for faster data retrieval. The register design can be enhanced further to reduce the number of components used and the area utilized and to increase the throughput and to reduce the delay.

## Acknowledgements

This research was supported by Visvesvaraya Technological University (VTU), India, Jnana Sangama, Belagavi.

## References

[1] G.M. Sridevi, M.V. Ramakrishna, A survey of hashing techniques for high performance computing, Int J Recent Innov Trends Comput Commun 4 (2016) 619–623.
[2] V. Suma, A novel information retrieval system for distributed cloud using hybrid deep fuzzy hashing algorithm, J Inform Technol Digital World 2 (2020) 151–160.
[3] P. Zuo, J. Sun, L. Yang, S. Zhang, Y. Hua, One-sided RDMA-conscious extendible hashing for disaggregated memory, in: 2021 USENIX annual technical conference- USENIX ATC 21 vol. 27, 2021, pp. 15–29.
[4] M. Nam, H. Cha, Y.R. Choi, S.H. Noh, B. Nam, Write-optimized dynamic hashing for persistent memory, in: 17th USENIX conference on file and storage technologies FAST vol. 19, 2019, pp. 31–44, 17.
[5] G.M. Sridevi, D.V. Ashoka, BLEH: bit-less extendible hashing for DBMS and hard disk drives, Int J Innovative Technol Explor Eng 9 (2019) 2191–2197.
[6] G.M. Sridevi, D.V. Ashoka, M.V. Ramakrishna, Extendible hashing with universal class of hashing functions, Int J Inf Technol Security 13 (2021) 51–66.
[7] K. Saravanan, S.K. John, R. Cheriyan, A. Senthilkumar, Hardware based cyber system using high performance crypto hash bloom filter for network security and privacy preserving applications, in: A handbook of internet of things in biomedical and cyber physical system vol. 165, Springer, Cham, 2020, pp. 39–59.
[8] X. Li, A. Gulila, Optimized memory allocation for less false abortion and better performance in hardware transactional memory, Int J Parallel, Emergent Distributed Syst 35 (2020) 483–491.
[9] R. Quislant, E. Gutierrez, E.L. Zapata, O. Plata, Leveraging irrevocability to deal with signature saturation in hardware transactional memory, J Supercomput 73 (2017) 2525–2557.
[10] Z. Yan, H. Jiang, W. Srisa-an, S. Seth, Y. Tan, Leverage redundancy in hardware transactional memory to improve cache reliability, in: Proceedings of the 47th international conference on parallel processing vol. 47, 2018, pp. 1–10.
[11] S. Irving, S. Chen, L. Peng, C. Busch, M. Herlihy, C.J. Michael, CUDA-DTM: distributed transactional memory for GPU clusters, in: International conference on networked systems vol. 4, 2019, pp. 183–199.
[12] X. Ren, M. Lis, High-performance GPU transactional memory via eager conflict detection, in: 2018 IEEE international symposium on high performance computer architecture-HPCA vol. 24, 2018, pp. 235–246.
[13] S. Sahay, M. Suri, Recent trends in hardware security exploiting hybrid CMOS-resistive memory circuits, Semicond Sci Technol 32 (2017) 123001–123018.
[14] S. Chang, X. Zhou, Z. Ding, Q. Li, A 12-bit 30MS/s SAR ADC with VCO-based comparator and split-and-recombination redundancy for bypass logic, in: 2019 IEEE international symposium on circuits and systems-ISCAS vol. 23, 2019, pp. 1–5.
[15] R. Zhang, S. Wijeratne, Y. Yang, S.R. Kuppannagari, V.K. Prasanna, A high throughput parallel hash table on FPGA using XOR-based memory, in: 2020 IEEE high performance extreme computing conference-HPEC vol. 24, 2020, pp. 1–7.
[16] X. Pan, X. Zhou, S. Chang, Z. Ding, Q. Li, A 12-bit 30-MS/s VCO-based SAR ADC with NOC-assisted multiple adaptive bypass windows, J Semiconduct 41 (2020) 1–11.
[17] T.Y. Wang, H.Y. Li, Z.Y. Ma, Y.J. Huang, S.Y. Peng, A bypass-switching SAR ADC with a dynamic proximity comparator for biomedical applications, IEEE J Solid State Circ 53 (2018) 1743–1754.
[18] Q. Fan, J. Chen, A 2.4 GS/s 10-bit time-interleaved SAR ADC with a bypass window and opportunistic offset calibration, in: IEEE 45th European solid state circuits conference-ESSCIRC vol. 45, 2019, pp. 301–304.
[19] D. Castro, P. Romano, J. Barreto, Hardware transactional memory meets memory persistency, J Parallel Distr Comput 130 (2019) 63–79.
[20] Z. Li, L. Liu, Y. Deng, J. Wang, Z. Liu, S. Yin, S. Wei, FPGA-accelerated optimistic concurrency control for transactional memory, In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture. 52 (2019) 911–923.
[21] M. Hemattil, A. Ahmadi, S.V. Makkil, M. Ahmadi, Hardware design of chaotic pseudo-random number generator based on nonlinear feedback shift register, in: 2018 IEEE 61st international midwest symposium on circuits and systems-MWSCAS vol. 61, 2018, pp. 980–983.
[22] D. Gruss, J. Lettner, F. Schuster, O. Ohrimenko, I. Haller, M. Costa, Strong and efficient cache side-channel protection using hardware transactional memory, in: 26th USENIX security symposium-USENIX security 17 vol. 26, 2017, pp. 217–233.
[23] M. Billmann, S. Werner, R. Höller, F. Praus, A. Puhm, N. Kerö, Open-source crypto IP cores for FPGAs–overview and evaluation, in: 2019 austrochip workshop on microelectronics-austrochip vol. 27, 2019, pp. 47–54.
[24] C. Gu, C.H. Chang, W. Liu, S. Yu, Q. Ma, M. O'neill, A modeling attack resistant deception technique for securing PUF based authentication, in: 2019 asian hardware oriented security and trust symposium-AsianHOST vol. 4, 2019, pp. 1–6.
[25] S.K. Satpathy, S.K. Mathew, R. Kumar, V. Suresh, M.A. Anders, H. Kaul, et al., An all-digital unified physically unclonable function and true random number generator featuring self-calibrating hierarchical Von Neumann extraction in 14-nm tri-gate CMOS, IEEE J Solid State Circ 54 (2019) 1074–1085.
[26] G.D.P. Stanchieri, A. De Marcellis, E. Palange, M. Faccio, A true random number generator architecture based on a reduced number of FPGA primitives, AEU Int J Electron Commun 105 (2019) 15–23.
[27] S. Karunamurthi, V.K. Natarajan, VLSI implementation of reversible logic gates cryptography with LFSR key, Microprocess Microsyst 69 (2019) 68–78.
[28] J. Zeng, S. Issa, P. Romano, L. Rodrigues, S. Haridi, Investigating the semantics of futures in transactional memory systems, in: Proceedings of the 26th ACM SIGPLAN symposium on principles and practice of parallel programming vol. 26, 2021, pp. 16–30.
[29] J. Jeong, J. Hong, S. Maeng, C. Jung, Y. Kwon, Unbounded hardware transactional memory for a hybrid DRAM/NVM memory system, in: 2020 53rd Annual IEEE/ACM

International Symposium on Microarchitecture-MICRO 53, 2020, pp. 525–538.
[30] C. Piatka, R. Amslinger, F. Haas, S. Weis, S. Altmeyer, T. Ungerer, Investigating transactional memory for high performance embedded systems, in: International conference on architecture of computing systems-ARCS 2020 vol. 5, 2020, pp. 97–108.
[31] S. Issa, P. Felber, A. Matveev, P. Romano, Extending hardware transactional memory capacity via rollback-only transactions and suspend/resume, Distr Comput 33 (2020) 327–348.
[32] M. Pedrero Luque, R. Quislant, E.D. Gutierrez- Carrasco, E. López-Zapata, O.G. Plata-Gonzalez, Speculative barriers with transactional memory, IEEE Trans Comput 71 (2020) 197–208.
[33] H. Aziza, J. Postel-Pellerin, H. Bazzi, P. Canet, M. Moreau, V. Della Marca, et al., True random number generator integration in a resistive RAM memory array using input current limitation, IEEE Trans Nanotechnol 19 (2020) 214–222.
[34] A. Palchaudhuri, A.S. Dhar, Speed-area optimized VLSI architecture of multi-bit cellular automaton cell based random number generator on FPGA with testable logic support, J Parallel Distr Comput 151 (2021) 13–23.