

Reducing Write Latency by Integrating Advanced PreSET Technique and Two-Stage-Write with Inversion Schemes

Sa'ed Abed*

Computer Engineer Department, Kuwait University, Kuwait.

* Corresponding author. Tel.: +965 24983790; email: s.abed@ku.edu.kw

Manuscript submitted December 10, 2019; accepted January 8, 2020.

doi: 10.17706/jcp.15.1.22-36

Abstract: The need for a memory with larger capacity is increasing to accommodate the high performance in computers. As this need grows, there is a very noticeable technology gap between the development in memory and processor. Currently, one popular area of research is to find new technology to improve the memory efficiency in one way or another. This research area is significant as memory is an essential component in any computer and it constantly needs improvement and upgrading. To address this issue, many studies are carried out on Phase Change Memory (PCM) that has a high potential to replace DRAM. In this paper, a new technique is studied and evaluated to enhance the overall performance by combining the PreSET technique and the Two-Stage-Write with inversion to reduce the write latency. This is achieved by pre-setting (set to 1) the dirty cache line beforehand and if the process fails, it shifts to Two-Stage-Write with inversion. The proposed technique reduces the running time of the write operation while keeping the power constrain into consideration. By exploring and obtaining results of the techniques stated above, the Two-Stage-Write and Two-Write-Stage with inversion have reduction on the read latency by average of 45.8% and 68.4%, respectively. The Two-Stage-Write, Two-Stage-Write with inversion and PreSET provided performance improvement over the baseline by 21.9%, 33.9% and 27.8%, respectively. Hence, the proposed technique showed a significant improvement.

Key words: PCM, Pre_SET, two-stage-write with inversion, latency.

1. Introduction

According to Moore's Law, the transistors count per chip will double every two years. As a result, the microprocessors performance improves by almost 60% per year. On the other hand, the memory access time of main memory improves by less than 10% per year [1]. The different improvement rates between the processor and Dynamic Random Access Memory (DRAM) resulted in a gap between in performance among the processor and main memory [2]. As the number of cores in the processor increases, the processor-memory gap continues to increase with the time that may lead to a deadlock where there will be no balance between the improvements in the two semiconductor technologies. The main memory is unable to fulfil the processor data requests which results in the processor being idle until the needed data is received. DRAM is a low cost, high density and high-performance memory and has been used in manufacturing main memory for decades. However, it lacks for scalability and has a high-power consumption due to the periodic refreshes in order to keep the data available. For that, numerous researches are being conducted to enhance or to find other technology to replace DRAM in specific.

To address this issue, many studies are done on Phase Change Memory (PCM) which is a widely used

technology to replace DRAM. PCM is a new technology that promises to flourish the future of main memory by improving power consumption and scalability [3]. PCM is a non-volatile memory (NVM) that has a better scalability in comparison with DRAM, has less power consumption since it is non-volatile, and does not need to refresh periodically to keep data. However, the performance of the memory system will be affected if PCM will be used due to two main limitations: slow writes and weak write endurance. Moreover, the long write latency has negative impact on the read latency of the PCM when a read operation is issued to the same PCM bank, as the read operation must stay idle until the write operation is completed. In addition, the write operation encounters asymmetry that means that the latency of writing a one (slower) is not the same as the latency of writing a zero (faster). Therefore, the overall latency is calculated based on the slower write operation (assumption made that SET incurs 8 times the latency of RESET [4]). In addition, the power consumed is not the same for both writes. Fig. 1 shows the difference between writing a zero or a one using a PCM cell. Writing a one (SET) requires low power but has long latency while writing a zero (RESET) requires high power but has a short latency. As shown in Fig. 1, the read from PCM requires low energy and low latency. Second, it has limited write endurance such that it becomes damaged after a certain period. In this paper, we aim to focus on solving the write latency issue.

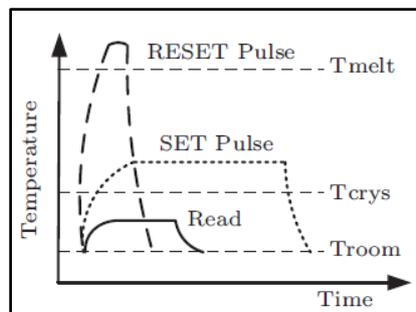


Fig. 1. Read and write mechanism in PCM [3].

We propose a new PCM write scheme, called Advanced PreSET that combines PreSET [5] write scheme with Two-Stage-Write with inversion scheme [4]. The main idea is to exploit the asymmetric writes and reduce the write command latency through separating the write operation into two stages write-1: writing one and write-0: writing zero. In addition, while being in the write-1 stage, the bits in the dirty cache line are set to ones in advance before an actual write operation while comparing the data in order to avoid overwriting ones in advance. Writing a zero can be completed at faster rate as they will be independent from the writing the slower write one operation. Moreover, as writing the one consumes less power, operation performed in write-1 stage can be done with more parallelism without violating the power constraints. The overall scheme will reduce the bits to be written and increase the parallelism.

The aim of this paper is:

- Utilizing the write asymmetry as much as possible.
- Accordingly, reducing the write latency by enhancing both factors:
 - Reducing the bits that need to be written into the PCM when a write operation is issued.
 - Enhancing the PreSET operation by comparing the bits.
- Increasing the parallelism on which bits are being written and hence utilized the power.

This paper is organized as follows. Section 2 presents a background information on PCM. Section 3 surveys the work carried out to overcome and reduce the processor-memory performance gap. Section 4 analyses the problem and demonstrates the proposed design methodology. The details of the how this technique was implemented is explained in Section 5 along with the suggested simulation tools that will be used. Finally, Section 6 concludes the paper and provides vision for future work.

2. Phase Change Memory (PCM)

PCM is a one-transistor and one-resistor device as shown in Fig. 2, while DRAM is a one-transistor and one capacitor device. Phase change material has two different states: an amorphous state and a crystalline state [5].

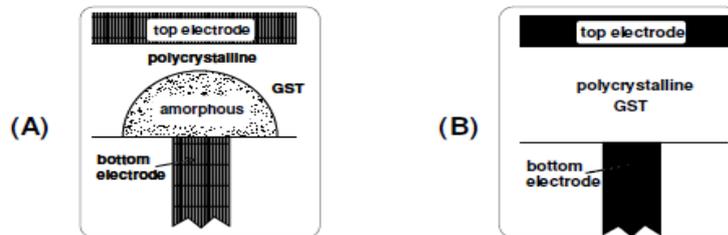


Fig. 2. PCM different phases [5].

The resistance of the state determines the bit stored in the memory cell. The amorphous state has a high resistance value, while the crystalline state has a lower resistance value. To read a PCM cell, we sense the current flow which is dependent on the resistance on the cell thus the phase of the phase change material. The write latency is much higher than the read latency due to the process of the phase change in the write operation. Moreover, a set operation (writing a one) has much higher latency than a reset operation (writing a zero) as shown in Fig. 3. Writing a zero requires a large current pulse for a short period in order to change the phase of the material into amorphous phase, while writing a one requires a small current for a long period in order to change the phase of the material into crystalline phase. Moreover, the memory controller chooses the slowest time i.e. time to write a one.

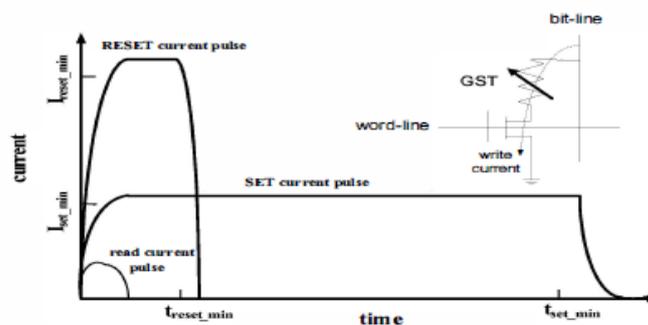


Fig. 3. PCM SET and RESET current pulses [5].

3. Related Work

In this section, a literature review is conducted where articles that describe similar topics are illustrated and explained.

In [5], the authors enhanced the PCM by a method called PreSET that benefits from the asymmetry write times. When a cache line becomes dirty and the main memory is idle, all the bits in the cache line will be set. If a write request is issued, only the bits that hold zeros are reset since the bits that hold ones are pre-set before issuing the write request. PreSET reduces the write latency by eight times, as all ones are written beforehand.

The authors in [4] proposed the idea of dividing the writes to PCM into two stages. This includes one stage that deals with writing only the ones (write-1-stage) while the other stage deals with writing only the zeros

(write-0-stage). The authors also suggested ensuring that ones are written in parallel such that it does not exceed the power constrain. Further to that idea, two stages write with inversion is proposed which combines two-stage-write with Flip-N-Write techniques.

In [6], [7], the authors demonstrated two different Partial techniques. Shi *et al.* in [6] proposed an enhancement to the PreSET scheme that is used to improve PCM performance through exploiting asymmetry in write operation. The main idea of the new scheme naming it as Partial-PreSET, is to SET the dirty bits of a cache line in a fine-grained fashion other than the previous PreSET which is conducted at a coarse-grained level. Li *et al.* in [7] proposed a different Partial-SET scheme to improve the long write latency problem of PCM. The focus was on improving the SET by reducing its latency period by exploiting the lower resistance value they found to endorse a short pulse i.e. Partial-SET pulse. The pulse programs the PCM cells to a pre-stable state thus accelerating the write operation.

Wang *et al.* in [8] proposed a new policy to schedule write operation in PCM. Write Power Asymmetry Scheduling (WPAS) considers the power constrains during while improving the write command parallelism of PCM memory. WPAS calculates the power consumption of every write command taking into consideration that the write power is asymmetric i.e. (Writing a zero consumes more power compared to writing a one).

In [9], [10], a 3D chips are integrated with PCM are used as replacement to DRAM. The work in [9] aims to solve the issues of high-power consumption, memory-processor speed gap, and low reliability. Integrating PCM with 3D integrated chips can reduce the access latency and power consumption due to the use of vertical interconnects. In addition, the friendly nature of PCM toward high temperature decreases the effect of the thermal issue even more in 3D ICs. The authors in [10] solved different issues related to the integration of 3D chips with PCM. 3D integrations have a drawback that is the constricted power that can be solved by using PCM that allows a zero cell leakage that will decrease the energy consumption. Nevertheless, integrating the PCM as main memory has two main drawbacks including the high write latency and the fact that PCM is limited, unlike the traditional DRAM memory. The authors solve these two issues by eliminating the number of repetitive writes which positively affects both the power and the PCM lifetime.

The work in [11] provided a summary of a variety of methods for controlling power consumption while using PCM. The authors summarized five different power management methods for PCM. The first technique is the hybrid PCM DRAM architecture. Another technique is read-before-write which solves the high write latency. A third technique is using a flag to indicate if the data is changed or not and then writing the unchanged data only. In addition, utilizing asymmetry in write times is another technique used. Lastly, memory access scheduling technique which focuses on decreasing the number of writes at the memory system or device level.

In [12], [13], the authors examined PCM with the addition of a buffer. The authors in [12] examined the architecture level model of PCM with a PCM-based hybrid main memory system. The advantages and disadvantages for a main memory system consisting of PCM storage with a small DRAM buffer are explored. They also addressed the issues in such hybrid memory architecture and demonstrate how to manage them. The authors in [13] replaced DRAM with PCM in general-purpose systems by re-designing the architecture to have high latency, energy writes, and finite endurance. It was concluded that a PCM system is slower than a DRAM system but with the addition of a buffer, the delay and energy gap is reduced by decreasing the write energy, enhancing the locality and merging the writes.

The research work in [14], [15] focused on enhancing MLC PCM using different techniques. The authors in [14] proposed a change in the architectural to improve the access time of MLC PCM. The main idea is to decrease the number of write operations using Write Truncation (WT) with the addition of an extra correction code (ECC). Moreover, Form Switch (FS) is also used to reduce the storage overhead of the ECC which also improves read latency. In [15], the authors exploited the MLC PCM access asymmetry and propose

TrisState-SET. The proposed technique uses the two-way program-n-verify (P&V) write approach that is a write technique that sets all dirty memory cells in full SET state.

The authors in [16] studied the replacement of Phase Change Memory (PCM) in main memory in a 3D die chip multiprocessor. On the other hand, 3D integrations have a drawback which is the constricted power. However, this can be solved by using PCM that allows a zero cell leakage which will decrease the energy consumption. Nevertheless, integrating the PCM as main memory has two main drawbacks including the high write latency and the fact that PCM is limited, unlike the traditional DRAM memory. The authors solved these two issues by eliminating the number of repetitive writes that positively affects both the power and the PCM lifetime.

The authors in [17] proposed a new write scheme which they named as Maximize the Power Budget utilization (MaxPB) to maximize the power utilization with minimum re-design efforts. The idea about the new scheme is to monitor the data units power requirements and then efficiently collect them into the least number of write units under the power constraints.

The authors in [18] proposed a new algorithm to map the hot virtual pages into the physical pages in order to improve the lifetime and the endurance of the PCM with almost no additions in cost in-term of energy and performance. Results of experiments conducted based on SPEC benchmarks show that the proposed technique can prolong PCM lifetime by hundreds of times within nearly zero searching and remapping overhead.

After extensive research on the topic of PCM and how it should replace the DRAM in the future, this paper will focus on exploring how to deploy PCM to enhance main memory speed and how to overcome the challenges of PCM by reducing the write latency. The closest to our work are the works carried out in [4] and [5].

4. Proposed Design Methodology: Advanced PreSET Technique

In this section, we are going to demonstrate our proposed technique in order to overcome the write latency. The Advanced PreSET technique aims to reduce the write latency of PCM by combing a technique called PreSET and Two-Stage-Write and Two-Stage-Write with inversion. Fig. 4 shows a comparison between the previous techniques discussed excluding PreSET.

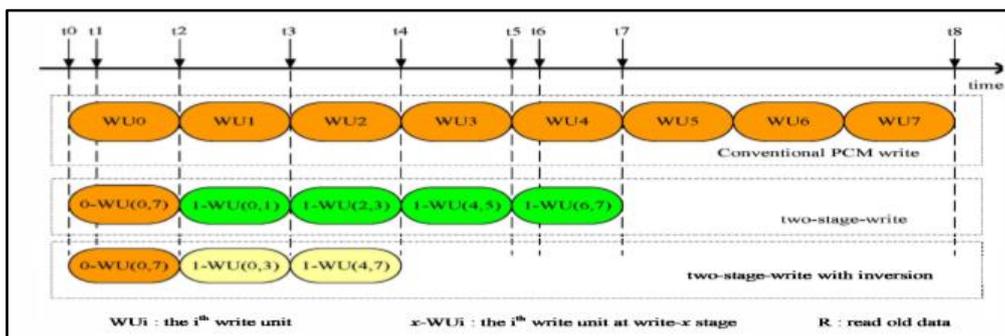


Fig. 4. Comparison between two-stage write and two-stage-write with inversion [4].

PCM requires remaining within the power constrains as well as reducing the write latency and there are two ways to reduce the latency, reducing the number of written bits or increasing the parallelism in which the bits are written. PreSET technique [4] reduces the write latency by reducing the number of written bits. Two-stage-write with inversion [5] reduces the number of bits written since only the changed bits will be written in parallel. The proposed technique, Advanced PreSET, combines the two techniques mentioned above by setting all the bits in the dirty cache line beforehand in parallel. In this way, the running time of the

pre-set operation will be reduced such that more than one pre-set operation can be complemented during the baseline pre-set run time without violating the power constrains as shown in the Fig. 5. Since the cache line will be pre-set, when a write request is issued, only zeros will be written so the running time of a write will decrease. This may provide an overall utilization which in return may enhance the performance.

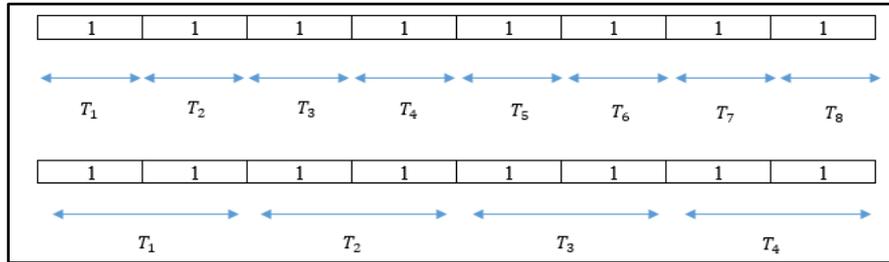


Fig. 5. Parallel pre-set.

PreSET sets all the bits in the dirty cache lines in advance such that if the cache line needed to be evicted from the cache and written back to PCM, only zeros need to be written resulting in faster write operation. PreSET can be enhanced even further by setting multiple bits at the same time as writing ones do not consume large power therefore speedups the operations and utilizes the power. In addition, only the bits that hold zeros are written and overriding the bits that already hold one is avoided. PreSET is performed when the memory is idle as write and read operations have higher priority than PreSET operations [5]. The proposed technique suggests the following, a PreSET operation in advance while the memory is idle of the projected write to the memory cell. If the operation is completed, then we move to the next operation which is the two-stage write. At this level, the dirty cache line is SET and only zeros will be written to the cells. If the memory is busy a PreSET will not be issued, and we will move into two-stage with inversion as shown in Fig. 6. Ideally, we want the PreSET operation for all writes to finish before the write reaches the memory system. However, PreSET is done only during idle periods and can be canceled many times in case there is heavy read/write traffic [5].

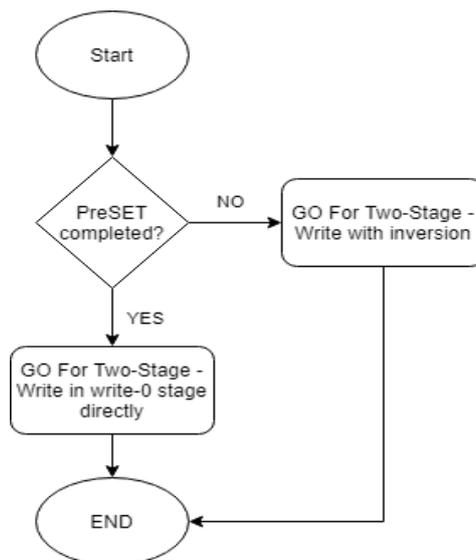


Fig. 6. The methodology operational flow.

A PreSET request to a memory line is issued as long a data is written to the cache line. A PreSET request should not be started without an actual write, as performing a PreSET operation for clean lines can result in potential data loss. Fig. 7 shows the PreSET window in which the PreSET can be done, PreSET window is the

duration in which a PreSET can be initiated and performed successfully. At time t_1 , the line is written (possibly by a writeback from the smaller caches) for the first time in the cache. After that, it can be written several times, before being evicted from the cache at time t_n , resulting in a writeback to memory. PreSET request can be sent to memory at any time after t_1 and must complete before time t_n [5].

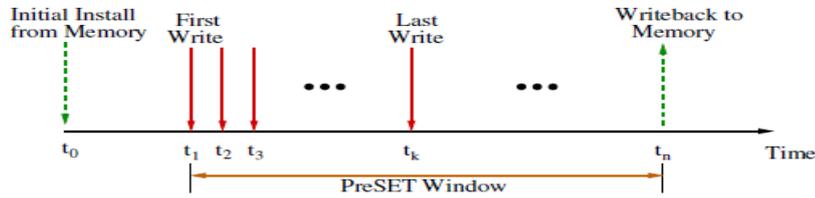


Fig. 7. The PreSET window [11].

A system that has a main memory of PCM has a read queue (RDQ) for read operations and a write queue (WRQ) for writes operations coming to fulfil processor requests as shown in Fig. 8. Fig. 8 shows that the PreSET technique [5] extends the design of the memory controller by adding a PreSET queue (PSQ) to store the addresses of the cache lines that are dirty and need to be pre-set. In addition, the cache design is extended by adding two extra bits to tag store entry to report the status of the pre-set operations as shown in Fig. 9.

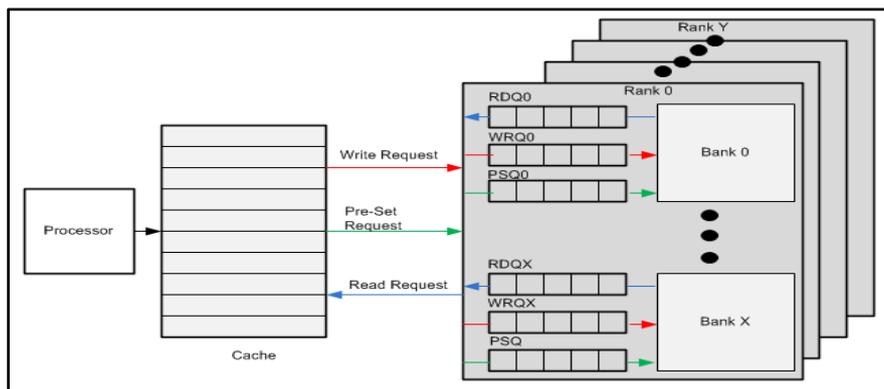


Fig. 8. Pre-set system [5].

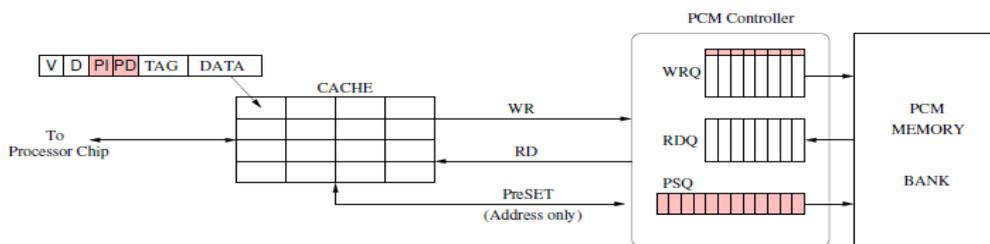


Fig. 9. System overview [5].

To avoid performing the pre-set operation for a line that is already being pre-set, PreSET Initiate (PI) bit is added to know whether a request need to be sent to PSQ or a request already has been sent. A request can be added to PSQ when it has an empty space. When the pre-set operation is performed, it is reflected in the PreSET Done (PD) bit which indicates that status of the pre-set operation. The pre-set operation is cancelled if the line needs to be evicted from cache and $PD=0$ while $PI=1$. In the PreSET technique, PD bit is added to the data and address that are in the WRQ to distinguish if the pre-set has been done so that only zeros need to be written or it has to write all the bits since PreSET operation is not performed [5]. In Advanced PreSET technique, the number of written bits will be decreased in any write request. When a line need to be written,

each bit of the old data is compared with the corresponding bit in the new data and only changed bits will be written (in the case of writing ones in the PreSET operation) i.e. a certain bit in a memory bank is 1, and a preSET operation I done a 1 will not be over written on the same spot. Therefore, the write latency will be reduced even further since only changed bits will be written and the overall execution time of the write operation will decrease. Table 1 shows the meaning of the bit values that are shown in Fig. 9.

Table 1. Bits in the Cache Line

Bit	Value	Meaning
D	0	The line holds the same data since been inserted.
	1	The data in the line has been modified.
PI	0	The pre-set operation is not initiated.
	1	The pre-set operation is initiated.
PD	0	The pre-set operation is not performed.
	1	The pre-set operation is performed.

Fig. 10 shows the flow that the cache line goes through starting from being dirty until being evicted from the cache and written back to main memory.

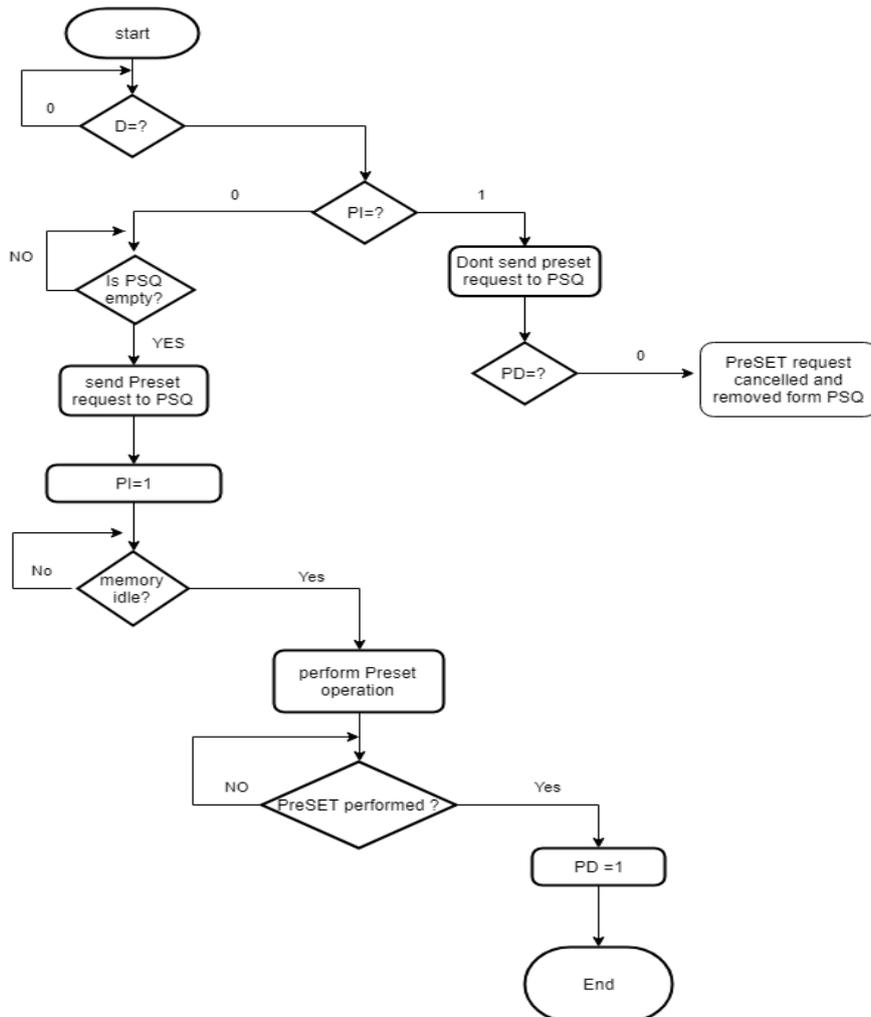


Fig. 10. Flowchart of the pre-set operation.

According to the design of the Samsung product-grade PCM prototype [4] the system has the ability write 16 bits in parallel. Parallelism can be increased for writing ones since writing a one requires half of the

required current to write a zero. Parallelism is helpful due to that fact that writing a one takes long time but requires less power such that the power can be utilized if more bits are written in parallel. As a result, 32 bits of ones can be written in parallel instead of 16. To set 32 bits in parallel, the flip flop of 32 bits will be enabled in parallel based on the dirty bit such that if the line is dirty (D=1), considered as a clock, and the input is Vcc.

5. Implementation and Results

The programmable logic device design software produced by Altera named Quartus II will be used to simulate Advanced PreSET technique. This program was chosen since it allows the user to analyze and synthesize VHDL codes. In other words, it gives the ability to compile designs, succeed timing analysis, view RTL diagrams and simulate a design's output depending on multisim. The development of the work proposed in this paper will be done using VHDL language that can be used for general purpose parallel programming. The proposed methodology details implemented in VHDL are shown in the Appendix.

The Finite State Machine for the proposed idea is shown in Fig. 11. It takes the old_data that is in the memory line, new_data that need to be placed in the memory line. If wr_op is 0, the write operation is a pre-set. Otherwise, it is a normal write. The output wr_op enables/disables the needed block to accomplish the write operation. Since a write request writes only 16 bits in parallel in each write unit, 8 write units are needed to write 128 bits. For pre-set requests, the request can be fulfilled within 4 write units since 32 bits are written at the same time. Therefore, 2 bits only can select what region to write in (ps_wu) for pre-set operation while 3 bits can select what region to write in (wr_wu) for write operation. The FSM produces the serial write units that it will write in depending on the wr_op. Fig. 12 shows that a demultiplexer is used to select what region to write.



Fig. 11. FSM of the proposed idea.

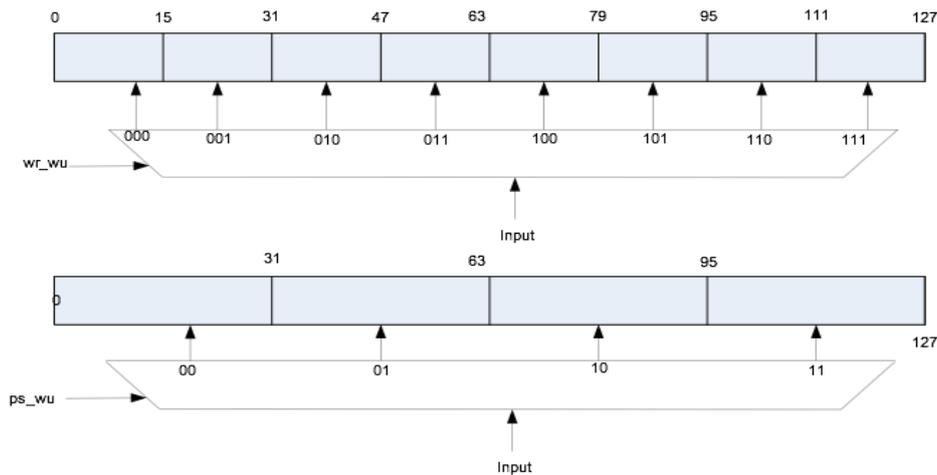


Fig. 12. Write regions for write and pre-set operations.

There are two different write operations: pre-set and write. Only one of these write operations can be enabled at a time by using an OR gate as shown in Fig. 13. If the wr_op input is 0, the write is a pre-set operation. The pre-set operation writes 32 ones in parallel into PCM to reduce the write latency when an actual write request is issued. To check if the bits needs to be written, it gets into an inverter, so it is written

only if the bit in the corresponding region in old_bit equals zero, otherwise, the bit does not need to be changed. As a result, it signals the AND gate to enable the pre-set operation. Table 2 shows the possible inputs and output of the pre-set operation.

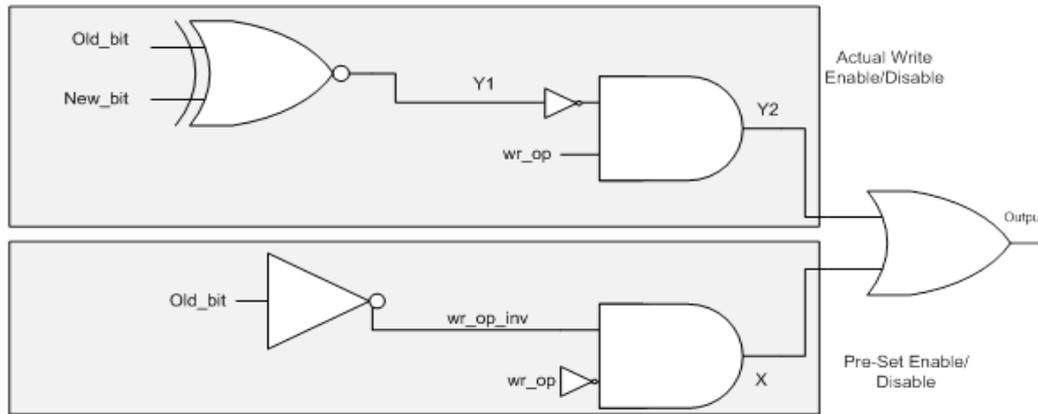


Fig. 13. Architecture of the write operations.

Table 2. Signal of the Pre-set Operation Block

Gate	Input 1	Input 2	Output	Meaning
INV	0	-	X1=1	Need to write a one
	1	-	X1=0	One is already written
AND	wr_op'=1	X1=0	X2=0	Disable Pre-set operation
	wr_op'=1	X1=1	X2=1	Enable pre-set operation

The write operation, based on the Advanced PreSET technique, will always compare the old bit with the new bit using a XNOR gate to reduce the number of written bits. The write operation is enabled only if the old data is not equal the new data and wr_op is 1 as shown below in Table 3.

Table 3. Signal of the Write Operation Block

Gate	Input 1	Input 2	Output	Meaning
XNOR	Old bit= 0	New bit = 0	Y1=1	No need to write
	Old bit= 0	New bit = 1	Y1=0	Data need to be changed
	Old bit= 1	New bit = 0	Y1=0	Data need to be changed
	Old bit= 1	New bit = 1	Y1=1	No need to write
AND	Enb=1	Y1'=1	Y2=1	Enable write operation
	Enb=1	Y1'=0	Y2=1	Disable write operation

The results of the proposed idea show improvements when all parts are combined and compared to baseline. The results seen from the Two-Write-Stage and Two-Write-Stage with inversion are on average 45.8% and 68.4% reduction on the read latency, respectively. The two-stage-write, two-stage-write with inversion and PreSET provides an improvement over the baseline of 21.9%, 33.9% and 27.8%, respectively.

6. Conclusion and Future Work

There is a technology gap between the memory and the processor. Many researchers are trying to find new methods to enhance the memory performance and how PCM is used to replace DRAM. This research showed two techniques were combined to create a new technique that has a promising result to improve the drawbacks of PCM that is its high write latency. By exploring and obtaining results of the techniques stated above, the Two-Stage-Write with inversion and PreSET have shown reduction on the read latency and provided performance improvement over the baseline by 21.9%, 33.9% and 27.8%, respectively. Hence, the

suggested technique has the potential to show an improvement over the baseline; as the write operation will be having two options either PreSET or Two-Stage-Write and both of them enhances the write operation and reduces the latency.

One of the future works is to test the proposed technique to produce all the possible results that can be obtained from using the DRAMsim or PCMSim tool. The PCM code is available online and can be checked and modified [19] as well as the DRAMsim [20]. Other basic techniques can be implemented and compared with the baseline technique to show how much improvement was achieved when using the new technique. PCM is a promising technology that constantly needs improvement due to some of its drawbacks.

Appendix

The proposed methodology is implemented in VHDL, the following is created:

Basic Components

The components of the data path described earlier in Fig. 13 are implemented.

- Registers: It is used to hold the inputs since each bit is stored in a register and the vector consist of many registers. Each bit is taken at a time during the one write unit; which consist of either 16 bit if the operation is a write or 32 bit if the operation is a preset.
- XNOR Gate: It is used to compare the old bit with the new bit to write only the bits that are changed.
- INVERTER: It helps to enable the preset operation only if the bit in the memory line is zero.
- AND Gate: It is used to enable one of the write operations only at a time depending on the wr_op. if wr_op = 0, the operations is a preset and the bottom shaded area in Fig. 13 is enabled. if wr_op = 1, the operations are a write and the top shaded area in Fig. 13 is enabled.
- OR Gate: It executes only the enabled operation that is selected using the AND gate.

Data Path

It combines all the needed components to create the data path of the technique shown in Fig. 13.

Control Unit

It implements the flow of how a memory line is written when a new data needs to be inserted in it. In the normal write operation, Advanced PreSET technique writes 16 bits in parallel in each write unit, state, so it needs 8 states to write a memory line of 128 bits. On the other hand, it writes 32 bits of ones in parallel in each write unit, state, in order to pre-set the line so it need 4 states to write a memory line of 128 bits. In each state, the specified number of parallel bits is compared; each old bit with its corresponding new bit. In the pre-set states (wr_op = 0), the bit in the memory line is compared with one and written only if it is zero. The 128 bits in the memory line is divided into 4 write units, each on the 32 bits which will be written in parallel. Each of the 4 units starts with a bit of an offset 0 and ends with a bit of an offset of 31. The following two statements are executed for each bit of the parallel 32 bits in each of the 4 write units:

```
s0 <= not ml_ps0(0);
...
s31 <= not ml_ps0(31);
if (s0 = '1') then ml_ps0(0) <= s0; end if;
...
if (s31 = '1') then ml_ps0(31) <= s31; end if;
```

In the write states (wr_op = 1), the bit in the memory line is compared with the bit in the new data and

written only if it is zero. The 128 bits in the memory line is divided into 8 write units, each on the 16 bits which will be written in parallel. Each of the 8 units starts with a bit of an offset 0 and ends with a bit of an offset of 15. The following two statements are executed for each bit of the parallel 16 bits in each of the 8 write units:

```
s0 <= not (ml_wr0(0) xnor wr_wu0(0));
...
s15 <= not (ml_wr0(15) xnor wr_wu0(15));
if (s0 = '1') then ml_wr0(0) <= NOT ml_wr0(0); end if;
...
if (s15 = '1') then ml_wr0(15) <= NOT ml_wr0(15); end if;
```

In general, new bits are written into the memory line only if the condition depending on the data path is true. Each state represents the operation happening to write the specified parallelized bits, and it moves to the next partition of the memory line.

Advanced PreSET System

This system combines both the data path and the control unit to simulate the technique. The Advanced Pre-Set Technique was successfully compiled under Quartus. It is assumed that the old data in the line, the new data that need to be written, the type of operation that need to be performed are given as inputs. Fig. 14 shows the implementation of the data path shown in Fig. 13.

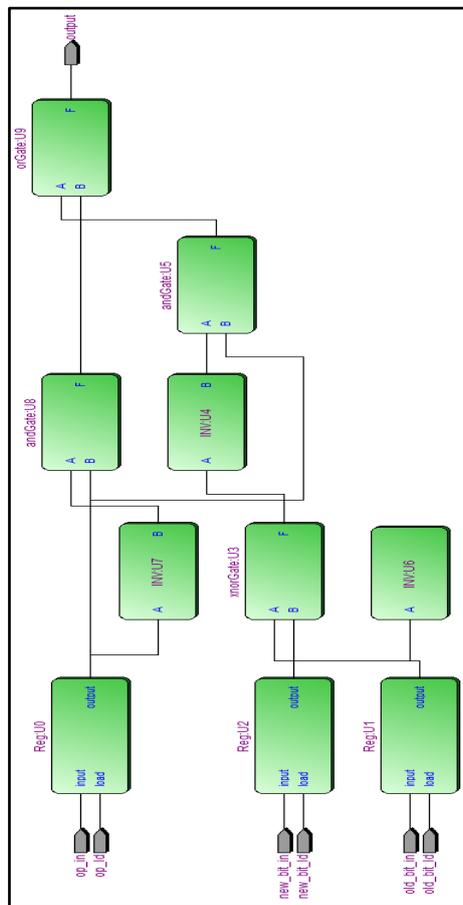


Fig. 14. RTL view of the proposed methodology data path.

The control unit that represents the FSM is shown in Fig. 15. The clock triggers the changes from one state to another and the write operation type determines how the many bits in the memory line depending on the write type. The inputs are the new data of 128 bits and the existing memory line of 128 bits. Since the new data will be written in the memory line, it serves as an output as well. The FSM shows that once the operation type, old data and new data loaded in the initial stage, the operation has two options as shown in Fig. 15. First, it goes into 4 states and finishes the operation if it is a pre-set. Second, it goes into 4 states and finishes the operation if it is a pre-set. As shown in Fig. 15, it goes back to the initial state to load another memory line once it is finished. Advanced Pre-SET system consists of both the data path and the control unit as shown in Fig. 16.

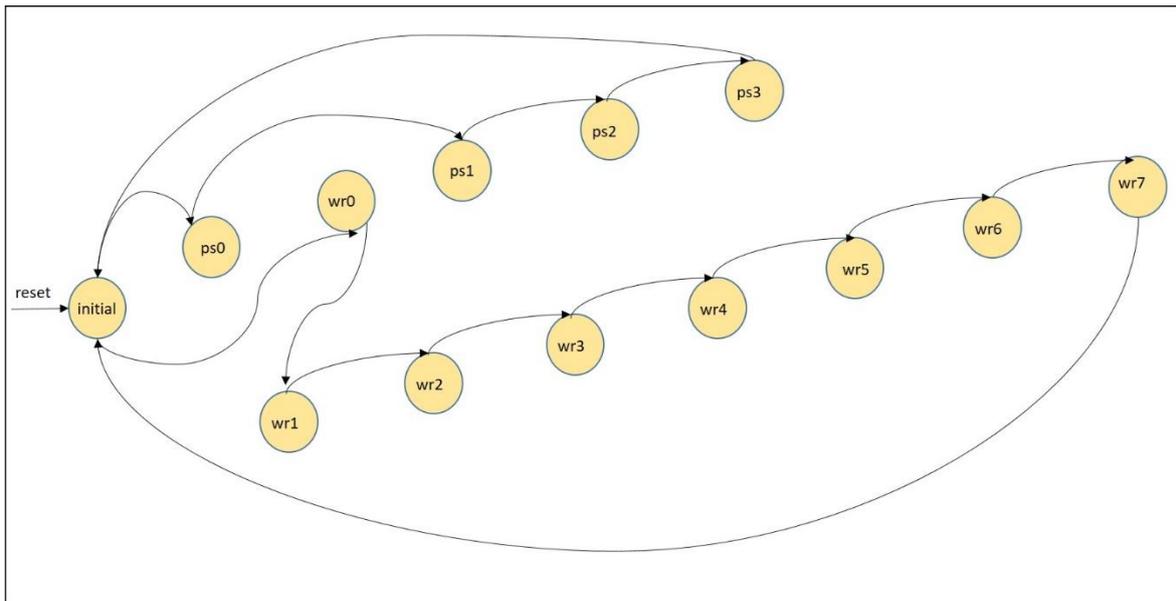


Fig. 15. The FSM view of Control Unit under Quartus.

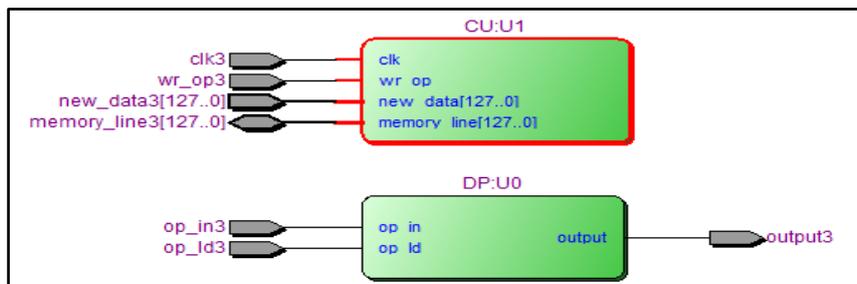


Fig. 16. The advance pre-set system.

Conflict of Interest

"The author declare no conflict of interest".

Author Contributions

SA conducted the research; analyzed the data; wrote the paper and approved the final version.

References

- [1] Liu, C. C., Ganusov, I., Burtscher, M., & Sandip, T. (2005). Bridging the processor-memory performance gap with 3D IC technology. *IEEE Design & Test of Computers*, 22(6), 556-564.
- [2] Hennessy, J. L., & David, A. P. (2011). *Computer Architecture A Quantitative Approach* (5th ed.). Morgan Kauffmann.

- [3] Xia, F., Jiang, D. J., Xiong, J., & Sun, N. H. (2015). A survey of phase change memory systems. *Journal of Computer Science and Technology*, 30(1), 121-144.
- [4] Yue, J., & Zhu, Y. (2013). Accelerating write by exploiting PCM asymmetries. *Proceedings of IEEE 19th International Symposium on High Performance Computer Architecture (HPCA'13)* (pp. 282-293). Washington, DC, USA.
- [5] Qureshi, M. K., Franceschini, M. M., Jagmohan, A., & Lastras, L. A. (2012). PreSET: Improving performance of phase change memories by exploiting asymmetry in write times. *Proceedings of the 39th Annual International Symposium on Computer Architecture (ISCA)* (pp. 380-391). Portland, OR.
- [6] Shi, Y., Zhu, Y., & Huang, L. (2018). Partial-PreSET: Enhancing lifetime of PCM-based main memory with fine-grained set operations. *Int J Parallel Prog*, 736-748. Retrieved from <https://doi.org/10.1007/s10766-017-0527-9>
- [7] Li, B. Hu. S., & Li, X. (2014). Partial-SET: Write speedup of PCM main memory. *Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1-4).
- [8] Wang, Q., Wang, D., & Hou, C. (2015). Exploiting write power asymmetry to improve phase change memory system performance. *Comput. Sci.*, 9(4), 566-575. Retrieved from <https://doi.org/10.1007/s11704-014-4185-4>
- [9] Zhang, W., & Li, T. A. (2010). *Exploring Emerging Three-Dimensional Integration and Memory Technologies in Processor Microarchitecture Design*. University of Florida.
- [10] Liu, C. C., Ganusov, I., Burtscher, M., & Sandip, T. (2005). Bridging the processor-memory performance gap with 3D IC technology. *IEEE Design & Test of Computers*, 22(6), 556-564.
- [11] Mittal, S. (2016). A survey of power management techniques for phase change memory. *Int. J. of Computer Aided Engineering and Technology*, 8(14), 424-444.
- [12] Qureshi, M. K., Vijayalakshmi, S., & Jude, A. (2009). Scalable high performance main memory system using phase-change memory technology. *ACM SIGARCH Computer Architecture News*, 37(3), 24-33.
- [13] Benjamin C. Lee, Engin Ipek, Onur Mutlu, & Doug Burger. (2009). Architecting phase change memory as a scalable dram alternative. *ACM SIGARCH Computer Architecture News*, 37(3). 2-13. DOI: <https://doi.org/10.1145/1555815.1555758>.
- [14] Jiang, L., Zhao, B., Zhang, Y., Yang, J., & Childers, B. R. (2012). Improving write operations in MLC phase change memory. *Proceedings of IEEE International Symposium on High-Performance Comp Architecture*, New Orleans, LA, (pp. 1-10).
- [15] Zhang, X., Zhang, Y., & Yang, J. (2015). TriState-SET: Proactive SET for improved performance of MLC phase change memories. *Proceedings of the 33rd IEEE International Conference on Computer Design (ICCD)*, New York, NY, (pp. 659-665).
- [16] Zhou, P., Zhao, B., Yang, J., & Zhang, Y. (2009). A durable and energy efficient main memory using phase change memory technology. *ACM SIGARCH Computer Architecture News*, 37(3), 14-32.
- [17] Li, Z., Wang, F., Feng, D., Hua, Y., Liu, J., & Tong, W. (2016). MaxPB: Accelerating PCM write by maximizing the power budget utilization. *ACM Trans. Archit. Code Optim.*, 13(4), 26.
- [18] Hoda, A. K., Xue, Y., & Yang, C. (2016). Fully exploiting PCM write capacity within near zero cost through segment-based page allocation. *J. Emerg. Technol. Comput. Syst.* 12(4), 26.
- [19] Peter, M. (2012). PCMsimsim. Retrieved from <https://github.com/huwan/pcmsim>
- [20] Bruce, J. (2005). DRAMsim. Retrieved from <https://eng.umd.edu/~blj/dramsim/v1/>

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).



Sa'ed Abed received his B.Sc. and M.Sc. in computer engineering from Jordan University of Science and Technology, Jordan in 1994 and 1996, respectively. In 2008, he received his Ph.D. in computer engineering from Concordia University, Canada. He has previously worked at King Faisal University in Saudi Arabia from 1997 to 2003. He joined Hashemite University, Jordan, as an assistant professor from 2008 to 2014. Currently, Dr. Abed is an associate professor in the Department of Computer Engineering at Kuwait University. His research interests include formal methods, VLSI design and image processing. Dr. Abed also served as a reviewer for various international conferences and journals.