# Flashmark: Watermarking of NOR Flash Memories for Counterfeit Detection

Prawar Poudel
*The University of Alabama in Huntsville*
*Huntsville, AL 35899*
pp0030@uah.edu

Biswajit Ray
*The University of Alabama in Huntsville*
*Huntsville, AL 35899*
biswajit.ray@uah.edu

Aleksandar Milenkovic
*The University of Alabama in Huntsville*
*Huntsville, AL 35899*
milenka@uah.edu

***Abstract*— Counterfeit electronics represents a significant concern because recycled, over-produced, out-of-spec, or cloned chips can enter globalized supply chains. This paper introduces *Flashmark*, a technique for watermarking of NOR flash memories for counterfeit detection. Flashmark relies on novel approaches for (a) imprinting watermarks into dedicated blocks of flash memory chips by repeated stressing, thus irreversibly changing physical properties of flash memory cells, and (b) reading out watermarks by sensing the changes in physical properties of flash cells through standard digital interfaces. The paper demonstrates Flashmark on embedded NOR flash memories in a family of low-cost microcontrollers.**

***Keywords—Flash Memories, Watermarking***

## I. INTRODUCTION

Electronic counterfeiting has been a longstanding problem due to the harmful impacts on many application domains, ranging from consumer electronics to mission-critical applications such as transportation, healthcare, and the military [1]. Unfortunately, this problem has become even more prevalent with increased globalization and trade. For example, reports of counterfeit integrated circuits (ICs) in the supply chain quadrupled from 2009 to 2011 [2]. In 2017, the U. S. military estimated that nearly 15% of its supply chain is made up of counterfeit parts [3], [4]. Sales of counterfeit parts result in substantial economic losses to the electronics industry, reportedly as high as $7.5 billion per year for U.S.-based chip companies alone [3]–[5].

Flash memory chips have become one of the primary targets of counterfeiters [6], [7]. There can be multiple pathways for the counterfeit flash memory chips to enter into the supply chain. First, the flash chips in the electronic gadgets in most cases remain functional even after the end of the product lifecycle. This provides an opportunity for the counterfeiters to retrieve the used flash memory chips from the printed circuit boards and recycle them as new with higher prices. Second, the rejected dies or the fall-out chips that fail post-fabrication tests can enter the supply chain through counterfeiters having access to the chip packaging sites, which are located in various countries. Even though the flash foundry marks some of the dies as rejected during die-sort testing, those dies can re-enter the supply-chain through counterfeiters. Third, the counterfeiters may buy inferior flash chips from less reputed manufacturers and sell them for a higher price by re-branding the chip packaging. The use of inferior or defective counterfeit non-volatile flash memories results not only in economic losses for original chip manufacturers, but may also result in failures in end-user applications, ranging from a loss of data and premature end-of-life to more serious catastrophic events.

A number of recent research efforts focuses on counterfeit IC detection. Most of the proposed techniques rely on physical and electrical inspection tests using high-tech imaging solutions and parametric/functionality tests [1], [8]–[11]. These techniques are costly as they require equipment for imaging and rigorous analysis by an expert, may be invasive, and suffer from limited effectiveness. Another approach for tracking ICs is to use Electronic Chip Identifiers (ECIDs) that are programmed into antifuse one-time-programmable memory [12]. Unfortunately, ECIDs are not common in flash memory chips; in addition, they require changes in physical chip masks and dedicated on-chip resources. Another class of techniques relies on deriving Physical Unclonable Functions (PUFs) that are unique for each chip [13]–[15]. However, the use of PUFs for detection of counterfeits require lengthy PUF extraction as well as maintenance of large databases with entries for every manufactured chip. In addition, it requires a method for contacting the chip manufacturer to verify the authenticity of each chip, which may place an additional burden on system integrators, as the time and costs of verification are increased. Two techniques have been recently introduced that specifically address the concern of recycled flash memory chips entering the supply chain [6], [7]. These techniques detect the changes in the physical properties of flash memory cells induced by prior flash operations using sweeping partial program [6] and partial erase operations [7] to characterize flash memory. Whereas these techniques can detect recycled flash memory chips, they may not be readily applied to detect other types of counterfeiting. Thus, developing a cost-effective technique for preventing the proliferation of counterfeit flash memory chips remains a significant challenge.

This paper introduces *Flashmark* – a technique for securing global supply chains of non-volatile NOR flash memory chips (or chips containing embedded NOR flash blocks) by using digital imprints or watermarks. The proposed method relies on (a) imprinting watermarks into physical properties of flash memory cells; and (b) extracting physical properties of flash memory cells through standard digital interfaces. The imprinting watermarks exploits a known property of flash memory chips that repeated program-erase operations degrade flash cell oxides. These changes in physical properties of flash memory cells are permanent and cannot be reversed [16], [17]. For example, if flash memory manufacturers watermark "accept" or "reject" information on every die they produce, they will prevent the counterfeiters from entering the out-of-spec or fall-out chips into the supply chain. Even if the counterfeiter gets physical access to the fall-out chips, they will fail to convert the "reject" watermark into "accept." The paper discusses an implementation of Flashmark on a low-cost microcontroller with an embedded NOR flash memory module and describes metrics of interest and design trade-offs.

Flashmark addresses many of the limitations of existing anti-counterfeiting methods. For example, writing and reading watermarks can be done from the flash controller with standard system commands. Hence the writing/reading of watermarks can be automated on numerous chips without any manual intervention. Second, the method for imprinting watermarks does not require inclusion of any extra hardware (e.g., antifuse memory or aging sensor circuitry) or modification of chip design/fabrication process. Thus, the method is universally applicable to all flash memory chips irrespective of the manufacturer. Third, unlike PUF-based anti-counterfeit methods, the

978-1-7281-1085-1/20/$31.00 ©2020 IEEE

watermark-based chip verification process will not require maintenance of chip-specific large databases at the manufacturing sites nor will the system integrators need to contact the original chip manufacturer to verify the authenticity of the chip. Since the method for imprinting watermarks is an irreversible process, the presence of correct watermarks in chips will guarantee their authenticity to the system integrators.

The rest of the paper is organized as follows. Section II gives NOR flash memory preliminaries. Section III describes a method for characterization of changes in physical properties of NOR flash memory cells caused by stressing through a standard digital interface. Section IV describes the proposed Flashmark procedures for imprinting and extracting watermarks. Section V describes the experimental evaluation and Section VI concludes the paper.

## II. NOR Flash Memory Preliminaries

### A. Flash memory cells and organization

Flash memories are non-volatile memories widely used for storing code and data in a variety of computer systems. They consist of a matrix of memory cells. Each memory cell is a MOSFET with an additional floating gate (FG) placed between the Control Gate (CG) and the conductive channel that is formed in the substrate between the source (S) and the drain (D) as shown in Fig. 1(a). Fig. 1(b) shows a symbol for a floating-gate transistor. The floating gate, surrounded by the oxide and electrically insulated from the transistor terminals, can trap negative charge that stays on it even when power supply is turned off. A flash memory cell typically keeps one bit of information (single-level cells or SLCs), though multi-level cells (MLCs) are used in high-density flash memories that can store multiple bits in a single cell. An SLC flash memory cell is in a so-called erased state (reads as a logic '1') when there is no trapped negative charge on the floating gate, whereas it is in a programmed state (reads as a logic '0') when there is negative charge.

To change the state of a flash memory cell, two operations are performed: *program* and *erase*. The program operation charges the floating gate with electrons, whereas the erase operation removes the charge from the floating gate. These operations require high voltages and are carried out through the oxides as shown in Fig. 1(a). To program a cell, a large voltage is applied to the source terminal inducing source-side hot carrier injection (SSI) that traps electrons on the floating gate ("Program" arrow). The negative charge on the floating gate reduces the voltage between the control gate and the source, thus increasing the threshold voltage ($V_{TH}$=$V_{THP}$) as shown in Fig. 1(c). To erase flash memory cells, a large positive voltage is applied on the control gate to remove electrons from the floating gate via Fowler-Nordheim tunneling ("Erase" arrow). The removal of electrons decreases the threshold voltage ($V_{TH}$=$V_{THE}$) as shown in Fig. 1(c). Threshold voltages $V_{THP}$ and $V_{THE}$ are not discrete values, but rather occupy a range of values with probability density functions illustrated on Fig. 1(d). A read operation from flash memory involves applying a read voltage on the control gate, $V_{CG}$=$V_{READ}$~3V, and a sense voltage on the drain, $V_D$=$V_{SENSE}$~2V, and sensing the threshold voltage. An erased cell conducts the current and that is sensed as a logic '1,' whereas a programmed cell does not conduct the current and that is sensed as a logic '0.' Program and erase operations degrade the oxide layers separating the floating gate from the substrate and the control gate, reducing the ability of flash memory cells to hold a charge for an extended period of time. A flash memory cell can sustain a finite number of program/erase cycles before it becomes unreliable, meaning it may still function but not consistently.

Based on the arrangement of flash cells, flash memories can be classified into NOR or NAND flash memories. NOR flash memories offer random read access through address and data buses, fast reads, high reliability, and low standby power. NAND flash memories are

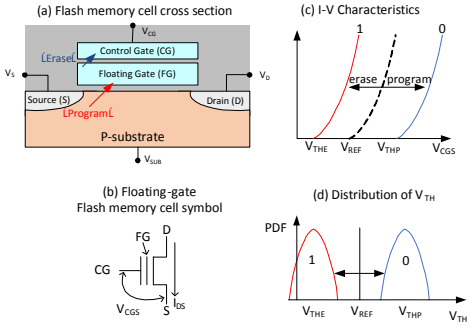


Fig. 1. (a) Cross section of a floating gate flash memory cell; (b) Symbol of FG-MOSFET; (c) I-V characteristics of FG-MOSFET; (d) Threshold voltage ($V_{TH}$) distribution of erased and programmed states.

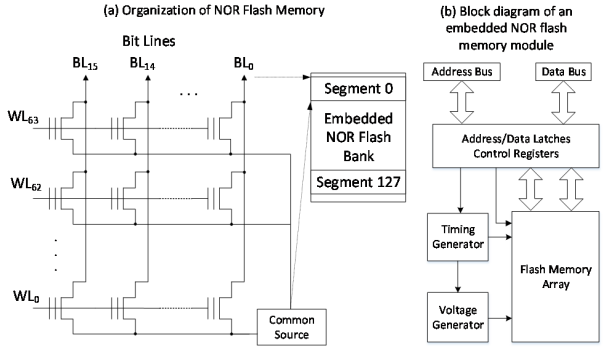


Fig. 2. (a) NOR flash memory organization; (b) Block diagram of an embedded NOR flash memory.

designed for high-capacity and low-cost storage solutions, and they do not allow for random access. Rather, data can be read from and written into the flash memory at the granularity of pages (2 Kbits to 32 Kbits). In NOR flash memories, the control gates of cells in each row are electrically connected through a line called Word Line (WL) as shown in Fig. 2(a). The drains of all cells in a single column are electrically connected through a single line called Bit Line (BL). The source terminals of all cells are also electrically connected to a common source terminal. The number of cells in each row defines a word size. Multiple words create a segment, and multiple segments create a bank. NOR flash memory reads are performed at a byte or a word granularity as all the bits in a word share the same WL.

### B. NOR flash memories in microcontrollers

Modern microcontrollers targeting low-cost and low-power applications are typically designed as a system-on-a-chip that integrates a processor core, flash memory, RAM memory, clock subsystems, and a number of input/output peripherals. As non-volatile memories, NOR flash memories are used to store code and data constants. In the default mode of operation, the microcontroller flash memory behaves as a ROM, allowing instruction fetches and data reads at a byte or word level. However, flash memories are often in-system programmable through a flash memory controller that supports programming individual bits, bytes, or words. Each flash memory bit can be programmed from logic '1' to logic '0' individually, but an erase operation is required to convert a bit from logic '0' to logic '1'. In other words, overwriting an existing flash content does not work without an intermediate erase operation. Flash memory controllers typically support an erase operation that will erase an entire flash segment and often a mass erase operation that will erase an entire flash memory bank.

```
CharacterizeSegment(SegAddr, TERASE)
for (tPE = 0; tPE = TERASE; tPE += Δt):
  Erase the entire segment (all cells read as 1s);
  Program all words in the segment (all read as 0s);
  Initiate the segment ERASE operation;
  Wait for tPE;
  Abort the ERASE operation;
  AnalyzeSegment(segaddr, N, cells_1, cells_0);
end for
```

```
AnalyzeSegment(SegAddr, N, cells_1, cells_0)
cells_0 = 0; cells_1 = 0;
for each word W in the segment:
  Read W N times;
  for each bit W[i] in the word:
     if (sum(W[i])>N/2): cells_1 += 1;
     else: cells_0 += 1;
  end for
end for
```

Fig. 3. Pseudocode used for characterization of flash cells using partial erase operation.

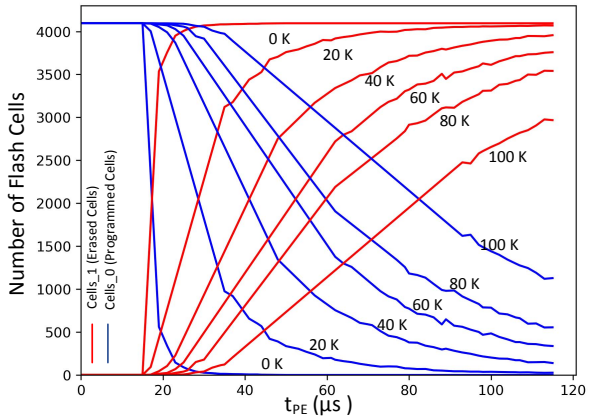


Fig. 4. State of flash cells in a segment as a function of the partial erase time.

Fig. 2(b) shows a block diagram of a flash memory module in a microcontroller that includes a flash memory with one or more banks and a flash memory controller. The controller includes voltage generators that supply voltage levels needed for program and erase operations, as well as timing generators that control duration of operations that span multiple clock cycles. A flash memory program (write) operation can be initiated by a microcontroller code that is running from within the flash memory itself or from RAM. When initiating a write operation from within the flash memory, the processor is typically halted until the program operation completes. When a flash operation is initiated from RAM memory, the processor can continue executing the code from the RAM, provided it does not interact with the flash memory bank that contains a location currently being programmed. A typical flash segment erase/program operation cycle encompasses time to bring up voltage generators, perform the erase/program operation, and remove programming voltages to allow flash memory to resume operation in its default mode. The total segment erase and word program times in a microcontroller used in our study are $T_{ERASE}$≈23–35 ms and $T_{PROG}$≈64–85 µs, respectively [18]. The erase and program operations can be aborted before they are completed by issuing an emergency exit command to the flash controller. Such operations leave flash memory cells in an undefined state. However, in this work we use these operations to extract analog physical properties of flash memory cells.

## III. Characterizing Flash Cell Physical Properties

As discussed in the previous section, flash memory cells subjected to repeated program/erase operations (P/E) will change their physical properties, namely their threshold voltages may deviate from the 'normal' levels. However, as long as the number of program/erase operations is below the endurance level, the distance between the threshold voltages is sufficiently large, ensuring correct behavior; thus, these changes in physical properties are not directly observable through a standard digital interface. This section describes experiments used to extract 'analog' physical properties of flash memory cells through the standard digital interface using so-called partial erase operation.

In this experiment, a number of flash memory segments is preconditioned by repeated program and erase operations, from fresh memory segments (0 K) to worn out flash memory segments (100 K). An unused flash memory segment is marked as 0 K, meaning it is not subjected to any P/E operations. A segment marked as 10 K is subjected to 10,000 P/E operations. One P/E operation or cycle here means that each word in a segment and each bit in a word is programmed and then the segment is erased. By repeating these steps 10,000 times, we create a segment that is moderately stressed at 10 K P/E cycles.

Fig. 3 describes an algorithm used to extract physical properties of flash memory cells in a segment. For a given memory segment at the address SegAddr subjected to a certain wear-out level, a sequence of operations is performed as follows. The segment is first erased and then fully programmed, ensuring that all cells read as a logic 0. Then, an erase operation is initiated. However, this operation is aborted after a period of time called partial erase time, $t_{PE}$. The segment is then analyzed by repeated reading of each word. Generally, each word is read $N$ times ($N$ is an odd number) and the bit map of a segment state after the partial erase operation is determined using the majority vote. Thus, if a word is read 3 times ($N$=3) and a particular bit is read as a logic 1 two or three times, it is marked as being a logic 1 (erased); otherwise it is marked as being logic 0 (programmed). The output of the algorithm is the number of cells that are characterized as a logic 0 (cells_0) and as a logic 1 (cells_1). The partial erase time is increased and the experiment is repeated until the nominal erase time, $T_{ERASE}$, is reached.

Fig. 4 shows the results of the characterization on a family of ultra-low-power microcontrollers from Texas Instruments. The figure shows cells_0 (blue lines) and cells_1 (red lines) as a function of the partial erase time (0 µs ≤ $t_{PE}$ ≤ 120 µs) for different segment stress levels. Note that nominal $T_{ERASE}$ is ~24 ms. When the segment is unstressed (0 K), all cells in the segment are programmed (4,096) for 0 µs ≤ $t_{PE}$ ≤ 18 µs. As the partial erase time increases, the cells abruptly switch their states. Thus, all 4,096 cells are in the erased state for $t_{PE}$ ≥ 35 µs. However, the shape of *cells_1* and *cells_0* curves is different on segments subjected to P/E stresses. The transitions are more gradual, and it takes markedly more time to erase all cells in the given segment. Thus, all cells become erased when $t_{PE}$ ≥ 115 µs on the segment subjected to 20 K P/E operations. The minimum $t_{PE}$ times when all cells in a segment read as erased are increasing with a stress level and they are 203 µs, 226 µs, 687 µs, and 811 µs (not shown on the graph due to scale) for 40 K, 60 K, 80 K, and 100 K P/E cycles, respectively.

The results of this experiment demonstrate that analyzing the states of flash segment cells after a partial erase operation could be used as a means to determine the stress level or analog properties of flash memory cells. By choosing appropriate partial erase time, $t_{PEW}$, one characterization round of operations is sufficient to determine the state of individual flash memory cells. Fig. 5 illustrates this process. A round of characterization with $t_{PEW}$ is conducted on a flash memory segment. If the majority of flash cells are programmed, that means that the cells are worn out resisting the erase operation (the cells have been subjected to 50 K stress cycles in this example). If the majority of cells already transitioned to the erased state, they are considered fresh. This way we

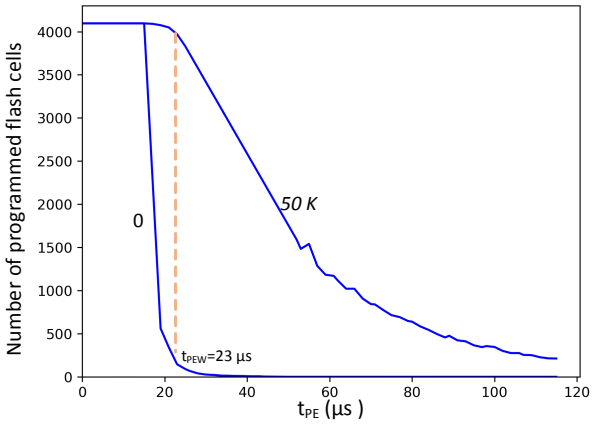


Fig. 5. Detecting changes in physical properties caused by stressing. By using $t_{PEW}$=23 µs we can distinguish 3,833 bits out of 4,096.

can extract the analog properties of flash memory cells (stressed/unstressed) through a digital interface. These properties are used in the proposed Flashmark.

## IV. Proposed Watermarking Scheme: FlashMark

The proposed imprinting of watermarks into a NOR flash memory is performed by chip manufacturers during the die-sort testing phase. The watermark may include manufacturer identifier, die identifier, chip speed grade, chip testing status (e.g., "accept" or "reject"), and other manufacturing related information. The current practice is that a chip manufacturer performs an erase followed by a program operation on a flash segment reserved for keeping manufacturing information. This metadata can be used by system integrators to verify chip origins. Unfortunately, this information can easily be erased, forged, or fabricated by counterfeiters.

Instead, we propose a more elaborate process of imprinting watermarks that will be resilient to counterfeiters' tampering efforts. For the sake of simplicity, let us assume that a word is reserved for a watermark and that we want to imprint ASCII code for the string "TC", a short for a virtual *Trusted Chipmaker* company. Initially, the reserved flash block has never been used, so all memory cells are "fresh" or "good," having their physical properties correspond to near perfect ones. To imprint the watermark, repeated erase-program operations are performed as illustrated in Fig. 6. The erase operation (E) removes

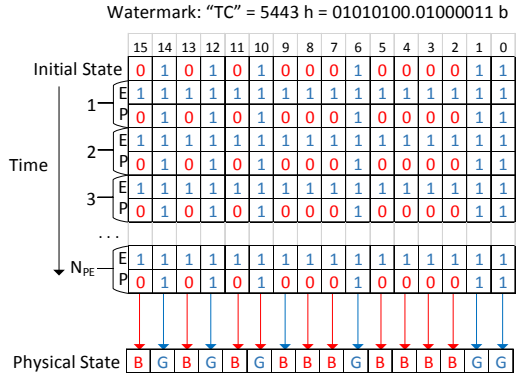


Fig. 6. Imprinting into flash memory word: an illustration.

**ImprintFlashmark(SegAddr, $N_{PE}$, Watermark)**

```
for stress = 1 to NPE
  Erase the entire segment (read as all 1s)
  for i = 1 to num_words
     Program each Word[i] with Watermark[i]
  end for
end for
```

Fig. 7. Algorithm for imprinting watermarks.

**ExtractFlashmark(SegAddr, $t_{PEW}$)**

```
Erase the entire segment (read as all 1s)
Program the entire segment (read as all 0s)
Initiate the segment erase operation
Wait for tPEW
Abort the erase operation
Read all flash cells
```

Fig. 8. Algorithm for extracting watermarks.

charges from the floating gates, which is equivalent to having all memory cells in a "1" state, whereas the program operation (P) writes the content that corresponds to the digital equivalent of the watermark. The process of repeated erasing and programming is continued until there are sufficient differences in physical properties between (a) memory cells that never change their state - let us call these cells "good," or G, and (b) memory cells that continually go through charge/discharge cycles - let us call these cells "bad," or B. The bad flash memory cells accumulate defects in the oxides as they are program-erased a number of times. These defects or degradations are permanent and thus cannot be reversed by motivated counterfeiters. Thus, the watermarks are imprinted into the physical properties of flash memory cells and their physical states "good" or "bad" are used to encode the digital watermark. Fig. 7 describes steps taken in imprinting a watermark into a segment of NOR flash memory.

The reading of watermarks is performed by system designers to verify that chips are genuine, before they are incorporated into their products. Please note that we use the term "reading" to describe the process of retrieving or extracting watermarks and status information that is imprinted into flash memory, i.e., it is not equivalent to a common flash read operation and typically involves an elaborate sequence of steps. In order to read the watermark, we need to extract physical properties of memory cells and characterize them as either "good" or "bad". However, it is a challenging task to distinguish between "good" and "bad" memory cells using a digital interface only. Additional challenges arise from the fact that semiconductor manufacturing processes induce significant cell-to-cell variations that need to be distinguished from those that are induced by the imprinting process. As described in the previous section, we utilize the partial erase operations to distinguish between the "good" and "bad" cells. Fig. 8 outlines steps taken in extracting the watermark. As an input parameter we use the partial erase time that brings the flash segment containing the watermark into the state that maximizes likelihood of extracting signatures. This time (or rather a time window) is determined by the manufacturer using the characterization process described in Section III for each family of devices and can be publicly communicated to system integrators.

## V. Experimental Evaluation

The experimental evaluation is performed on a family of low-power microcontrollers from Texas Instruments, MSP430F5438 and MSP430F5529. They feature a processor core, RAM memory, DMA, a range of analog and digital peripherals, and an in-system programmable multi-bank NOR flash memory. The flash memory banks are divided into 512-byte segments. Multiple chip samples are used and we find that flash memories within the same family show consistent behavior when subjected to proposed techniques.

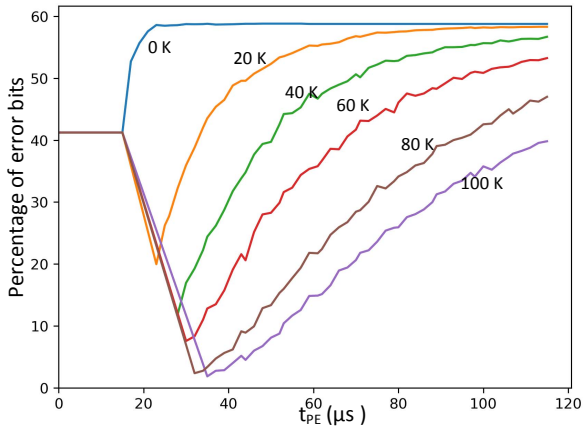


Fig. 9. Bit error rate for a single-read 512-byte watermark extraction as a function of the partial erase time.

The goal of the experimental evaluation is to determine feasibility of the proposed watermarking as well as to explore its design space and design trade-offs. Specifically, we want to determine the key parameters and design metrics such as: (a) bit error rate – the percentage of watermark bits that can be correctly retrieved using the ExtractFlashmark procedure; (b) imprint time – the time needed to imprint the watermark in a flash memory segment using ImprintFlashmark procedure; (c) extract time – the time needed to extract watermark using the ExtractFlashmark procedure, and (d) the flash memory overhead.

The feasibility of the proposed watermarking is explored as follows. A watermark that consists of upper-case ASCII characters is imprinted into flash memory segments while varying the number of P/E cycles, $N_{PE}$, from 0 K (no imprinting) to 100 K times, which matches the endurance of the flash memory. The watermark extraction procedure is used as described in Fig. 8 (with a single read) with varying $t_{PE}$. Fig. 9 shows the bit error rate as a function of $t_{PE}$ for different levels of segment stressing. In the case of the unstressed segment (0 K, blue line), the bit error rate matches the percentage of bits set to logic 1 in the watermark for small $t_{PE}$ and it reaches the percentage of bits set to a logic 0 for larger $t_{PE}$. For the segment subjected to $N_{PE}$ = 20 K P/E cycles, the bit error rate drops to as low as 19.9% for a particular partial erase time (orange line). By increasing the parameter $N_{PE}$, the physical differences between unstressed and stressed cells increase, and thus our ability to recover the original watermark. Thus, the minimum bit error rates are 11.8%, 7.6%, and 2.3% for $N_{PE}$ = 40 K, 60 K, and 80 K, respectively. From this experiment, we can conclude that higher the number of stresses is, the lower is the bit error rate when extracting the watermark. Next, we can observe that there is a particular window for

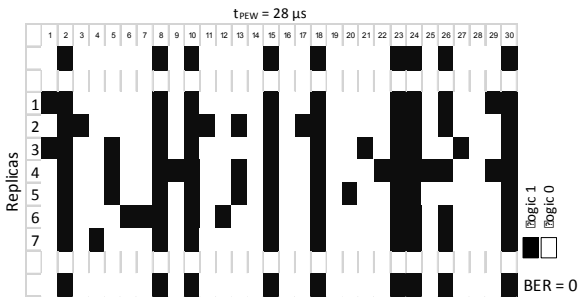


Fig. 10. Extracting watermarks from replicated copies using majority voting.

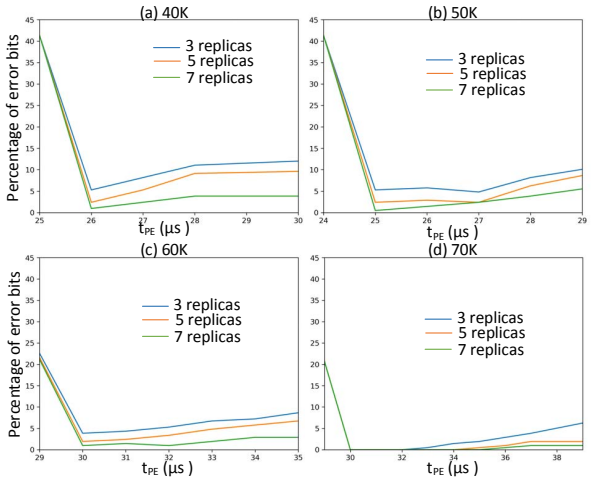


Fig. 11. Impact of watermark replication on bit error rates.

the partial erase time that minimizes the bit error rate in the extraction procedure. This time window slightly shifts to the right as we increase the number of stresses.

Ideally, we would like to have a minimum number of P/E stresses and thus reduce imprint time and to have no bit errors during extraction procedure. As shown in Fig. 9 these two are conflicting requirements. In addition, achieving bit error rates of zero is not feasible due to manufacturing variations that result in different physical characteristics of flash memory cells, random noise effects, charge retention effects, and others physical processes. Thus, extensions to the baseline technique are desirable to reduce the bit error rates.

As watermarks require modest memory footprint, watermark data can be imprinted at multiple locations. An alternative to watermark data replication is to use error correction techniques. In this paper, we consider effects of simple data replication – the watermarks are replicated 3, 5, or 7 times. The actual bit value is determined using majority voting on bit values retrieved from each replica. Fig. 10 illustrates 7-way replication of a 30-bit vector (a portion of a watermark). The segment has been stressed 50 K times and the watermark is extracted from each replica using the partial erase time $t_{PEW}$=28 µs. The top row shows the actual watermark; the black squares represent bits at a logic 1 and white square represent bits at a logic 0. Rows 1 to 7 show extracted watermark bits from seven replicas. For example, the extracted watermark from first replica has errors on bits 1 and 29. The last row shows the recovered watermark from all seven replicas utilizing the majority voting that fully matches the imprinted watermark.

Fig. 10 also shows that bit errors occur frequently on stressed bits – i.e., the probability of mischaracterizing a "bad" bit as a "good" bit is significantly higher than the probability of mischaracterizing "good" bit as a "bad" bit. This observation can be utilized for further tuning of watermark extraction procedures. Another advantage of using replicated watermarks is that the range of suitable partial erase times $t_{PEW}$ (those that provide low error rates) widens when compared to cases when there is no replication. Fig. 11 shows the impact of watermark replication on the bit error rates for a range of the partial erase times of interest for four segments that are imprinted 40 K, 50K, 60K, and 70K times. The top left graph shows that the minimum bit error rates are 5.2%, 2.4%, and 0.96% for 3, 5, and 7 replicas on the watermark that used as low as 40 K P/E cycles. This is a significant improvement when compared to 11.8% observed with no replication. The watermark prepared using $N_{PE}$=70 K can be fully recovered with no errors using 3-way replication. Fig. 11 also shows that the bit error rate is less sensitive

to the partial erase time used during extraction procedure than in the case of non-replicated watermarks.

It should be noted that counterfeiters may try to tamper with the watermark by turning remaining "good" cells into "bad" ones, but such an effort will produce illegitimate watermarks that can be easily uncovered by system integrators. To ensure the integrity of information in watermarks, we may impose constraints on watermarks, e.g., to contain an equal number of "good" and "bad" bits so any tampering can be detected. Alternatively, in addition to watermarks we may imprint watermark signatures that will ensure that concurrent tampering by attackers cannot go undetected.

The time to imprint the watermark is directly proportional to $N_{PE}$. The baseline implementation that utilizes full segment erase cycles (25 ms) and block writes (~10 ms) requires 1380 s for 40 K P/E cycles and 2415 s for 70 K P/E cycles. Fortunately, these times can be significantly reduced by using a premature exit of erase cycles without any negative impact on the wear level of flash memory cells. The accelerated imprint procedures reduce the imprint times for slightly over 3.5 times to 387 s for 40 K and 678 s for 70 K P/E cycles. It should be noted that the imprint times are bounded by the type of the NOR flash memory used in this study. A number of stand-alone NOR flash memory chips have significantly faster erase and program operations and we expect that their imprint time will be significantly smaller. In addition, NOR flash memory cells are typically designed to be more robust and resilient to stresses than their NAND counterparts. Unlike the ImprintFlashmark procedure, the ExtractFlashmark procedure is quite fast. Our baseline implementation requires only 170 ms when multiple replicas of the watermark are used. The memory overhead is acceptable as a single flash memory segment is sufficient for storing the original watermark with its replicas.

## VI. Conclusions

This paper describes Flashmark – a technique for watermarking NOR flash memory chips for counterfeit detection. The technique relies on repeated program/erase operations of watermarks to change physical properties of flash memory cells that can be extracted through a digital interface using partial erase operations. The proposed technique is successfully demonstrated on embedded NOR flash memories in a low-power microcontroller, but the proposed method is applicable broadly to NOR and NAND flash memories. The paper explores the metrics of interest such as the bit error rates, the watermark imprint time, and the watermark extraction time.

Methods for the cost-effective and robust watermarking and tracking of the usage of memory chips or SoCs will enable chip manufacturers to eliminate economic losses due to counterfeit chips as well as to better protect their brand names. Likewise, system integrators and end users will be able to detect counterfeit chips before integrating them in their products, preventing damages that may arise from using counterfeit chips.

## References


[1] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug. 2014.

[2] "Reports of Counterfeit Parts Quadruple Since 2009, Challenging US Defense Industry and National Security - Omdia." https://tinyurl.com/spusztx (accessed Aug. 21, 2019).

[3] "National Defense Authorization Act for Fiscal Year 2012." [Online]. Available: https://tinyurl.com/rc4bpzt.

[4] EETimes, "Chip counterfeiting case exposes defense supply chain flaw," *EETimes*, 24-Oct-2011. https://tinyurl.com/u7wvlkh (accessed Oct. 27, 2019).

[5] "Counterfeits Costing Semiconductor Industry Billions - EE Times Asia." https://tinyurl.com/wcuzooz (accessed Oct. 23, 2019).

[6] Zimu Guo, X. Xu, M. M. Tehranipoor, and D. Forte, "FFD: A framework for fake flash detection," in *Proceedings of the 54th Annual Design Automation Conference (DAC)*, 2017, pp. 1–6.

[7] S. Sakib, P. Kumari, B. M. S. B. Talukder, M. T. Rahman, and B. Ray, "Non-Invasive Detection Method for Recycled Flash Memory Using Timing Characteristics," *Cryptography*, vol. 2, no. 3, p. 17, Sep. 2018.

[8] S. Shahbazmohamadi, D. Forte, and M. Tehranipoor, "Advanced Physical Inspection Methods for Counterfeit Detection," in *Proceedings of the 40th International Symposium for Testing and Failure Analysis (ISTFA)*, Houston, TX, USA, 2014, pp. 55–64.

[9] K. Ahi, N. Asadizanjani, S. Shahbazmohamadi, M. Tehranipoor, and M. Anwar, "Terahertz characterization of electronic components and comparison of terahertz imaging with x-ray imaging techniques," in *Terahertz Physics, Devices, and Systems IX: Advanced Applications in Industry and Defense*, 2015, vol. 9483, p. 94830K.

[10] B. B. Hu and M. C. Nuss, "Imaging with terahertz waves," *Optics Letters*, vol. 20, no. 16, pp. 1716–1718, Aug. 1995.

[11] U. Guin, D. DiMase, and M. Tehranipoor, "A Comprehensive Framework for Counterfeit Defect Coverage Analysis and Detection Assessment," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 25–40, Feb. 2014.

[12] N. Robson *et al.*, "Electrically Programmable Fuse (eFUSE): From Memory Redundancy to Autonomic Chips," in *IEEE Custom Integrated Circuits Conference*, 2007, pp. 799–804.

[13] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA, 2002, p. 148.

[14] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th Annual Design Automation Conference (DAC)*, San Diego, California, 2007, pp. 9–14.

[15] Y. Wang, W. Yu, S. Wu, G. Malysa, G. E. Suh, and E. C. Kan, "Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints," in *IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, 2012, pp. 33–47.

[16] Y. Wang, W. Yu, S. Q. Xu, E. Kan, and G. E. Suh, "Hiding Information in Flash Memory," in *IEEE Symposium on Security and Privacy*, 2013, pp. 271–285.

[17] L. M. Grupp *et al.*, "Characterizing flash memory: anomalies, observations, and applications," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture - Micro-42*, New York, New York, 2009, p. 24.

[18] "MSP430F543x, MSP430F541x Mixed-Signal Microcontrollers datasheet (Rev. F)." Texas Instruments Incorporated, Aug-2009, [Online]. Available: http://www.ti.com/lit/ds/symlink/msp430f5438.pdf.