Abstract—The reliability of non-volatile NAND flash memories is reaching critical levels for traditional error detection and correction. Therefore, to ensure data trustworthiness in nowadays NAND flash-based Solid State Drives, it is essential to exploit powerful correction algorithms such as the Low Density Parity Check. However, the burdens of this approach materialize in a disk performance reduction. In this work a standard decoding approach is compared with an optimized solution exploiting hardware resources available in NAND flash chips. The simulation results on 2X, 1X and mid-1X MLC and TLC NAND flash-based Solid State Drives in terms of disk bandwidth, average latency, and Quality of Service favor the adoption of the presented solution in different host scenarios and realistic workloads. The proposed solution is particularly effective when high error correction interventions and read- or write-intensive workloads are considered.

Index Terms—Solid-State Drive, SSD, ECC, Low Density Parity Check, LDPC, Endurance, NAND Flash, MLC, TLC

I. INTRODUCTION

Solid State Drives (SSDs) are now the most effective solution for fast mass storage systems in cloud services and high performance computing [1]. One main SSDs’ limitation is their reliability, which is dependent on the non-volatile NAND flash memories used as storage medium. These components, in fact, are subject to a progressive wear-out whose physical roots reside in the tunnel oxide degradation related to the mechanisms exploited for their program/erase. The aggressive technology scaling and the need for increasing memory capacity by storing more bits in a single cell (two bits Multi-Level Cells -MLC- or three bits Three-Level Cells -TLC-architectures) amplify the memories’ wear-out impact on the SSD reliability [2], [3]. In fact, as long as the number of bits stored in a single cell increases, the width of the threshold voltage distribution associated to a logical stored content decreases. As a consequence, the control of the entire set of voltage distributions, which drift with the endurance (i.e., number of program/erase -P/E- cycles) and retention time, is becoming more and more complex. A direct indication of this phenomenon is an increase in the Raw Bit Error Rate (RBER) in a NAND flash memory, that is the probability of having bits in error after a single read operation [4]. Such an increase translates into the inability to read correct data after a number of P/E operations or after long retention times. Fig. 1 shows the measured average RBER as a function of endurance for three MLC and one TLC NAND flash memories manufactured in 2X, 1X, and mid-1X technology nodes as described in Table. I. As it can be seen, as the number of P/E cycles increases, the error rate quickly grows up. In addition, either by scaling from a 2X to a mid-1X node or switching from a MLC to a TLC storage paradigm, the RBER increases significantly.

To broaden NAND flash reliability figures and, consequently, data trustworthiness over the whole SSDs’ lifetime, the use of sophisticated Error Correction Codes (ECC) is essential. This requirement is tightly coupled with the percentage of uncorrectable pages in a NAND flash memory that are pages which, if read, return a number of errors greater than the ECC’s correction limit. This latter value represents a quality metric of the whole SSD’s reliability because as soon as it is reached, NAND flash memories and therefore the disk are considered as failed [5]. Table. II shows the endurance measured for the 4 considered memories when a multi-threaded BCH decoder able to correct up to 100 errors in a 4320 Bytes codeword is used [6]. To overcome these limitations with the aim of moving the disk failure point as far as possible, powerful correction algorithms such as the Low Density Parity Check (LDPC) are considered.
TABLE I

<table>
<thead>
<tr>
<th>Sample</th>
<th>A-MLC</th>
<th>B-MLC</th>
<th>C-MLC</th>
<th>D-TLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory type</td>
<td>Consumer</td>
<td>Enterprise</td>
<td>Enterprise*</td>
<td>Enterprise*</td>
</tr>
<tr>
<td>Rated endurance</td>
<td>9 k P/E</td>
<td>12 k P/E</td>
<td>4 k P/E</td>
<td>0.9 k P/E</td>
</tr>
<tr>
<td>Measured Average Read Time [µs]</td>
<td>68</td>
<td>40</td>
<td>70</td>
<td>86</td>
</tr>
<tr>
<td>Measured Average Program Time [µs]</td>
<td>1400</td>
<td>2000</td>
<td>2500</td>
<td>2300</td>
</tr>
<tr>
<td>Program mode</td>
<td>dual plane</td>
<td>dual plane</td>
<td>dual plane</td>
<td>dual plane</td>
</tr>
<tr>
<td>Page size [Bytes]**</td>
<td>16384</td>
<td>16384</td>
<td>16384</td>
<td>16384</td>
</tr>
<tr>
<td>Technology node</td>
<td>2X</td>
<td>1X</td>
<td>Mid-1X</td>
<td>Mid-1X</td>
</tr>
</tbody>
</table>

*Early samples  
**w/o spare area

Due to their superior error correction capabilities, Low Density Parity Check (LDPC) codes now represent a forced choice for SSDs [7], [8]. Conventional LDPC decoders, if properly designed, can sustain a NAND flash RBER up to 10^-2 [8], [9], [10], [11]. The LDPC correction engine usually leverages on two sequential correction approaches: i) the hard decision (HD) which corrects errors by means of a single read operation of the selected memory page; ii) a sequence of soft-level decisions (SD) which perform, with considerably higher latency, a fine-grained multiple-read sensing operation that allows error correction by combining the multiple-read data with the original HD.

As summarized in Fig. 2, besides the HD whose data are stored in a buffer inside the LDPC decoder as a reference, each soft-level requires two page read operations with two different read references and two data transfers to the ECC engine. The algorithm continues this process until the page is correctly read or the maximum number n of soft-levels is reached and the page is marked as uncorrectable. The overall n-level SD algorithm requires 2n page reads and 2n data transfers operations. This serial approach is used mainly because high code-rates [12] are adopted to exploit the full SSD capacity and hence HD has the same limitations of BCH codes in terms of RBER [11]. Therefore as soon as this strategy fails to correct data, it is requested the intervention of the SD, with a higher correction range. However, in [11] it has been shown that, as soon as the HD approach starts to fail, there is an overhead both in terms of increased SSD power consumption and overall SSD latency since additional read operations are requested on NAND flash with respect to the HD approach.

An alternative LDPC correction approach that limits the drawbacks of the SD has been presented in [13]. The assumption of this methodology, named NAND-Assisted Soft Decision (NASD), is that data for ECC engine are produced by the NAND flash memory itself, which internally reads the target page twice for each soft-level. Then, read data are opportunely combined and only one transfer to the ECC is performed for each soft-level, as shown in Fig. 3, thus reducing the NAND flash I/O bus use. The NASD advantages become more pronounced when the impact on the command scheduling by the halved number of data transfers is taken into account.

In this paper we apply the NASD technique on 2X, 1X, and mid-1X MLC and TLC NAND flash-based SSD architectures to:

- show how NASD, thanks to a reduced number of data transfers and to the consequent impact on command scheduling, modifies significantly the SSD figures of merit: bandwidth, average latency, NAND flash I/O bus use, and Quality of Service (QoS) that is the ability of keeping a sustained performance over time within a defined threshold [14], [15], [16];
- compare the SSD performance at system level obtained exploiting the standard HD+SD and the HD+NASD LDPC. The analysis have been performed on two different host architectures: a consumer PC and an enterprise workstation;
- show, for the two host architectures, how NASD outperforms the traditional SD approach when synthetic 100% read and different realistic workloads such as MSN,
Financial, and Exchange [17] are considered;
The system performance have been evaluated by using the
SSDExplorer co-simulation framework [18], [19].

II. SOFT DECISION VS NAND-ASSISTED SOFT DECISION
NAND flash memories are read page-wise by using a
defined read reference, hereafter denoted as HD0. Cells are
read as 1 or 0 depending on their threshold voltage VT with
respect to HD0 (see Fig. 4a). If during the ECC decoding phase
the page is evaluated as uncorrectable, the LDPC decoding
algorithm can be retried with the SD. To accomplish this
second step, more information about the actual position of the
NAND flash threshold voltage distributions must be collected.
Basically, the algorithm moves sequentially the internal read
references to SD10 and SD11 (Fig. 4b) thus reading the page
twice and storing the two data content in two page registers
inside a page buffer. Data from the page buffer are transferred
byte-wise from the flash memory to the LDPC decoder and
then are analyzed with those previously read with HD0. This
step is possible because during the whole SD process the data
read with the HD0 are buffered inside the LDPC decoder and
are used as a reference. If the decoding process still fails, a
second iteration is performed by moving the read references to
SD20 and SD21 and comparing the new read data with the
HD as shown in Fig. 4c. The algorithm continues this process
until the page is correctly read or the maximum number of
soft-levels is reached and the page is marked as uncorrectable.

Table III summarizes the number of operations performed
by both algorithms. As it can be seen, NASD is able to halve
the number of page transferred from the NAND flash memory
to the ECC. As a consequence, the overall soft decision
process is shortened and hence, the SSD performance are
improved. To understand the effective NASD efficiency, it
must be taken into account that read operations are temporally
separated from the successive data transfer operations. Fig. 5
sketches the commands queue for NAND flash dies sharing
the same I/O bus, the corresponding data bus allocation, and
the ECC engine activity. After a HD0 read, the SSD controller
can send other read or write commands to the same NAND
flash die or to other dies. When the ECC engine communicates
the read failure to the controller, this latter stores the data
related to the HD and schedules the additional SD10 and SD11
reads. In the SD approach the two read data are transferred
separately when the I/O bus is available, with the risk that
between the SD10 and SD11 transfer the bus is contended
by other data transfers to/from other NAND flash dies (see
Fig. 5a). In the NASD approach, on the contrary, since SD10
and SD11 read data are combined in a single data transfer,
the consequent soft decision operation can start in advance
with respect to the SD case (see Fig. 5b). The advantages,
that become more pronounced when additional soft-levels are
considered, depend on the considered workload, as shows in
Section III-B. Moreover, since the number of data transfers
between the memory and the ECC are reduced, NAND flash
memory I/O bus accesses are reduced as well. This I/O bus
use reduction impacts the SSD dynamic power consumption.
The main component exploited by NASD is the NAND
flash page buffer which is used to store data for each soft-
level operation. In present NAND flash chips, this buffer is
composed by two registers used especially for read cache
and read retry operations [20], [21], [22], [23]. It becomes
clear that the NASD implementation does not require any
other register inside the memory and it can be performed by
a simple 8-Bit combinational logic placed between the
internal NAND flash page buffer and the I/O interface. In
fact, the two read operations performed by NASD can be
easily stored into the existing registers of the page buffer and
a simple block composed by 8 XORs (or 8 XNORs) acting as
a combinational circuitry is sufficient. Since the I/O interface
limits the parallelism to 8-bits, the logic combination between
the pages stored inside the two registers can be performed on-

<table>
<thead>
<tr>
<th>Command</th>
<th>Bus allocation</th>
<th>ECC activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HD0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>HD0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SD10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SD11</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. Time sketch, for a cluster of NAND flash dies sharing the same
data bus, of the command queue, of the data bus allocation, and of the ECC
engine activity. Numbers enlighten the events sequence during a single soft-
decision operation. Case a) and b) refer to the SD and NASD approach,
respectively.

### TABLE III

**Read and data transfer operations in SD and NASD approaches.**

<table>
<thead>
<tr>
<th>LDPC</th>
<th>One soft-level</th>
<th>Two Soft-levels</th>
<th>#n soft-levels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HD +</td>
<td>HD +</td>
</tr>
<tr>
<td>SD</td>
<td>2 page read +</td>
<td>4 page read +</td>
<td>#2n page read +</td>
</tr>
<tr>
<td></td>
<td>2 data transfer</td>
<td>4 data transfer</td>
<td>#2n page transfer</td>
</tr>
<tr>
<td>NASD</td>
<td>2 page read +</td>
<td>4 page read +</td>
<td>#2n page read +</td>
</tr>
<tr>
<td></td>
<td>1 data transfer</td>
<td>2 data transfer</td>
<td>#n page transfer</td>
</tr>
</tbody>
</table>
Fig. 6. NASD combinational circuitry. Just 8 logic XOR gates (or XNOR) have to be added before the 8-bit I/O interface. Data read from the NAND flash array are temporarily stored in Register #1 and Register #2. After that they are byte-wise combined by the NASD circuit and transferred to the 8-bits I/O bus.

Fig. 7. Characterization board used to stress the tested NAND flash chips.

the-fly in a byte-wise fashion during the data transfer phase (see Fig. 6). Regardless of the internal NAND architecture, just a single combinational logic can be integrated in a single NAND chip. As a consequence, the NASD implementation inside a NAND flash memory becomes a easy task which does not impact neither chip area nor power consumption.

III. EXPERIMENTAL SETUP AND RESULTS

Results have been collected by means of: i) a dedicated NAND flash memory characterization system which collects RBERs and statistics on uncorrectable pages; ii) a hardware implementation of the LDPC code which computes real decoding latencies; iii) a co-simulation SSD framework able to produce bandwidth, latency of a target disk architecture starting from previously collected reliability data and ECC statistics [18], [19].

Fig. 7 shows the test equipment exploited for memories characterization. It is composed by a programmable FPGA, a DRAM buffer for temporary data storage and a dedicated socket for NAND flash memory interfacing. Each tested device has been sequentially stressed with random data patterns. For testing purposes, each NAND flash memory has been stressed with a number of P/E cycles higher than their rated endurance (Table. I).

Fig. 8 shows the LDPC characterization setup. The board has been configured to generate random data patterns emulating different RBER values from a NAND flash. The codeword is computed and decoded by the LDPC board, whereas an external PC gathers encoding and decoding latencies to be further exploited by the co-simulation SSD framework. The HD correction capability of the LDPC engine has been set with the same correction strength used by the BCH code described in the Introduction (i.e., up to 100 errors in a 4320 Bytes codeword).

Fig. 9 shows the percentage of uncorrectable pages when only a HD-LDPC approach is used. As it can be seen, mid-1X MLC and mid-1X TLC memories show a high HD fail rate so that SD would be constantly required. As a consequence, NASD technique advantages are evident resulting in a higher SSD read bandwidth, an improved QoS and a lower average read latency. On the contrary, 1X-MLC and 2X-
MLC memories show a low percentage of uncorrectable pages which grows up only in proximity of the rated endurance. In these cases, error correction is less required and hence NASD advantages are present yet barely perceptible. It must be highlighted that two soft-levels were sufficient to correctly read all the tested memories up to twice the rated endurance, for both SD and NASD approaches.

The simulated SSD architecture is summarized in Table IV. Fig. 10 shows the main building blocks of the SSD. Besides the standard I/O processor exploited for the host-interface address fetch phase and the many-core processor on which the operations’ scheduler is executed, there is also an I/O processor acting as a read/write dies sequencer. In order to fully exploit the internal parallelism offered by the SSD, host random addresses which could cause die collisions (i.e., requests for a die already scheduled) are parsed and sequentially issued to NAND flash chips. In such a way, even if random commands are sent by the host, only sequential patterns are processed by NAND flash memories hence maximizing the throughput. To achieve accurate simulation results, command scheduling phenomena such as queuing and pipelining have been considered.

All data have been collected simulating two different host platforms (Table V). The first one is a consumer system which does not exploit the full SSD architecture (able to sustain 450 kIOPS) since I/O requests settle around 200 kIOPS. As a consequence, all internal error recovery techniques which exploit additional read operations produced by the ECC for the soft decoding step are partially hidden by the SSD’s architecture which masks all non-user reads. The second one is an enterprise workstation designed to serve hundreds of parallel processes which requests up to 600 kIOPS. In this case the disk performance cannot match this specification so that any further read produced by any error recovery technique will burden on the final SSD’s performance. Thanks to these two different test-cases it has been possible to test the NASD effectiveness over standard SD when disk resources such as NAND-flash I/O buses are partially or completely allocated for user operations.

Results presented in Section III-A refer to an enterprise host and a 100% 4 kB random read workload which represents the most challenging situation for the SSD performance characterization. In fact, when mixed read/write workloads are considered, since the DRAM chip in the SSD caches all the write operations, the measured average latency and bandwidth figures of the disk do not reflect the actual SSD behavior. Section III-B will extend the discussion to realistic workloads for both hosts.

A. 100% random read workload - Enterprise host

Fig. 11 shows the SSD’s read bandwidth gains achieved by the NASD approach with respect to two soft-levels SD as a function of the memory endurance for the 4 considered memory types and the Enterprise host.
Fig. 12. SSD average read latency gain achieved by NASD with respect to two soft-levels SD as a function of the memory endurance for the 4 considered memory types and the Enterprise host.

Fig. 13. Cumulative percentage on a normal probability paper of the SSD latency calculated at twice the rated endurance when a D-TLC sample is used and both SD and NASD are considered. The QoS threshold is calculated as the 99.99 percentile of the cumulative distribution [14].

Fig. 14. Calculated QoS at twice the rated endurance for the 4 considered memory types and the Enterprise host.

The predictability of low latency and consistency of high plausible to extract the SSD’s QoS defined as the 99.99 percentile calculated at twice the rated endurance for the D-TLC sample. Latency has been calculated as the average time elapsed the 99.99 percentile of the cumulative distribution [14].

Fig. 12 shows the average read latency gains achieved by NASD with respect to SD as a function of memory endurance. Latency has been calculated as the average time elapsed between a read command submission and its completion. All results concerning average latency reflect those obtained for bandwidth (Fig. 11).

Fig. 13 shows the SSD’s cumulative latency distributions calculated at twice the rated endurance for the D-TLC sample and both SD and NASD approaches. From these data it is possible to extract the SSD’s QoS defined as the 99.99 percentile of the cumulative latency distribution [14]. QoS represents the predictability of low latency and consistency of high bandwidth while servicing a defined workload and it can be considered as the key metric to assess the SSD’s performance in a worst-case scenario. Fig. 14 shows the calculated QoS at twice the rated endurance for all the considered memories and for both the SD and NASD approaches.

B. Realistic workloads - Enterprise and Consumer hosts

Since the NASD advantages are tightly coupled to the RBER showed by the NAND flash memories and to the command pattern, simulations have been also performed con-considering three realistic workloads [17], as detailed in Table VI. Write ratio represents the percentage of write commands in the command sequence, whereas write amplification factor denotes the number of additional writes produced by the SSD firmware for each single host write [28].

In the MLC and TLC architectures the write throughput is smaller than the read throughput (see Table I). In fact, to lower the RBER retrieved during read operations, sophisticated but long program algorithms are used [2], [29]. To deal with this bandwidth mismatch, it is usual to leverage multi-plane program commands which allow writing, on the same memory die, two or more pages in the time-frame of a single page program. This approach, on the one hand allows maximizing the program throughput towards the NAND flash dies, on the other hand, however, it severely impacts the I/O bus transfer time. In fact, for each program operation two or more 16 kBytes pages have to be transferred from the controller to the target memory die thus making the I/O bus busy for long times. In the NAND flash memories considered in this work write operations are performed in a dual-plane mode (see Table I). Therefore, before scheduling the actual program operation on a memory die, a chunk of 32 kBytes has to be moved from the SSD controller to the NAND flash die. As a consequence, since 4 kBytes chunks are read by the host during a read operation, it is clear that when programs are scheduled, the I/O bus is busy for a time which is 8x longer than a read. In light of these considerations and taking into account the scheduling effects shown in Fig. 5, it is clear that NASD will show better results either in extremely write intensive workloads (i.e., MSN) or in read intensive workloads. In the former case the probability of having a long write transfer between the two read operations required by the standard SD technique is high, whereas in the latter case other read operations can

<table>
<thead>
<tr>
<th>Workload</th>
<th>Write ratio [%]</th>
<th>Write amplification factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSN</td>
<td>96</td>
<td>1</td>
</tr>
<tr>
<td>Financial</td>
<td>81</td>
<td>1.32</td>
</tr>
<tr>
<td>Exchange</td>
<td>46</td>
<td>1.94</td>
</tr>
</tbody>
</table>
be scheduled on different dies belonging to the same channel between the two reads required by standard SD.

Tables VII - X show the bandwidth, the average latency, the QoS, and the NAND flash I/O bus use at twice the rated endurance for the 4 tested NAND flash memories and for the two host architectures. NAND flash I/O bus use, sampled with a 1 µs period, is representative of the dynamic power consumption of the whole internal I/O bus. As expected, simulations show that NASD outperforms SD when other commands are scheduled between the two data transfers required by the SD technique, thus temporally separating the data transfer operations and introducing a performance degradation.

NAND-assisted soft decision (NASD) is evaluated by comparing its performance with standard LDPC decoding approach. NASD advantages are highlighted when QoS is concerned since QoS is a metric for worst-case latency conditions rather than an average behavior such as bandwidth and average latency. As it can be observed, the QoS improvements for the MSN workload are in a 20% - 40% range. When looking at the NAND flash I/O bus use, (see Table X), advantages are materialized only when a 100% random read workload is considered. In fact, when write intensive workloads are devised, the I/O bus transfer time taken by program operations overshadows that of read operations, therefore the reduction in the number of read transfers materialized by NASD is blurred.

### IV. Conclusions

In this paper the potential of a LDPC technique called NAND-assisted soft decision (NASD) is evaluated by comparing its performance with standard LDPC decoding approach. The effectiveness of NASD has been proven through simulations of a 2X MLC, a 1X MLC, a mid-1X MLC, and a mid-1X TLC NAND flash-based SSDs running on a consumer and on an enterprise host system. The results, gathered for synthetic and realistic workloads, show the significant advantages of NASD with respect to the standard approach in particular when the Quality of Service is considered.
REFERENCES


**Alessia Marelli** is Senior Design Engineer at Microsemi. Before Microsemi, she joined Integrated Device Technology (IDT) in 2009 as senior digital designer, where she took care of ECC applied to SSD. In 2007, she joined Qimonda as senior digital designer. From 2003 to 2007 she joined STMicroelectronics, Agrate B., Italy where she was involved in digital design of Multilevel NAND Memories, especially redundancy, ECC and algorithms. She is co-author of Memories in Wireless Systems (Springer, 2008), Error Correction Codes for Non-Volatile Memories (Springer, 2008), Inside NAND Flash Memories (Springer, 2010) and Inside Solid State Drives (Springer, 2013).

**Rino Micheloni** is Fellow at Microsemi. Before Microsemi, he was Lead Flash Technologist at IDT (Integrated Device Technology), Senior Principal for Flash and Director of Qimonda’s design center in Italy, developing 36 nm and 48 nm NAND memories. From 2001 to 2006 he managed the Napoli design center of STMicroelectronics focusing on the development of 90 nm and 60 nm MLC NAND Flash. Before that, he led the development of MLC NOR Flash.

**Piero Olivo** received the Ph.D. in electronic engineering from the University of Bologna, Bologna, Italy, in 1987. He has been a Full Professor of Electronics with the University of Ferrara, Ferrara, Italy, since 1994. His research interests include the physics, the reliability and the experimental characterization of innovative non-volatile memory cells and architectures.