The Search for a Practical Iterative Detector for Magnetic Recording

Rob Lynch, *Member, IEEE*, Erozan M. Kurtas, *Member, IEEE*, Alex Kuznetsov, Engling Yeo, and Borivoje Nikolić, *Member, IEEE*

Abstract—The striking benefits of iterative detection have generated strong interest in the disk drive signal processing area, but thus far application of this technology has been rather limited. We review the benefits that the most interesting iterative detectors have over the industry-standard partial-response maximum-likelihood (PRML) detectors, and examine the hardware complexity issues that have so far stood in the way of practical implementations.

Index Terms—Implementation complexity, iterative detection, magnetic recording, turbo codes.

I. INTRODUCTION

THE HISTORY of magnetic recording is largely a story of increasing capacity. Areal densities have increased at 60%/year for decades, with occasional spurts of over 100%/year when major technological advances are introduced. The continued increase in rotational speeds, which reduces the seek times and increases the throughput, coupled with the growth in areal density, requires a continual increase in signal processing data rates in the read/write channel. As the difficulty in increasing areal densities through improvements in heads and media rises, more attention is given to improvements available from advanced signal processing. For several generations of products the detection of data was accomplished with maximum-likelihood implemented with a Viterbi decoder, operating on a read signal that has been equalized to a specific partial response target. Over time, the partial response target has changed to more accurately match the signal spectrum, and the Viterbi decoder has been augmented by the addition of noise prediction to handle colored noise and postprocessing to account for signal-dependent noise. However, improvements in this partial-response maximum-likelihood (PRML) technology family are becoming more difficult to obtain. A promising technique to achieve significant further signal-to-noise ratio (SNR) improvement is the use of iterative detection and soft information. Initially proposed in 1993 for communications channels [1], this technique combines the concept of soft information (working with the probability of a bit's value rather than the value itself) and iteration (encoding data using two codes and then iterating back and forth between two decoders, each with knowledge of only one of the constituent codes). Initial investigations of iterative detectors for applications in magnetic recording channels created a great deal of interest among researchers in both academia and industry in recent years.

A large number of publications have appeared in the areas of code design and optimum code performance (see references in [2]), but somewhat less attention has been paid to decoder architectures or to implementation and system issues. Although iterative detectors promise large gains over conventional PRML systems, they have not been used in commercial applications so far.

We believe that this situation is due, at least in part, to the difficulty in finding and achieving the optimum tradeoff between performance and complexity/cost. Disk drive read/write systems have traditionally been very cost sensitive. The area of the silicon chip directly dictates its price, while power dissipation has to be low enough to allow for inexpensive packaging and system cooling. Historically, complementary metal–oxide–semiconductor (CMOS) scaling has been used to allow more advanced signal processing and higher speed, while the power and area stay roughly the same in each new technology generation. Thus, the challenge is to find the algorithm that achieves the best SNR performance at reasonably high speed while staying within these constraints. Of course, there are other hurdles to be overcome, not the least of which concern various system implications of using iterative detection. These important considerations, however, are beyond the scope of this paper.

II. POWER AND PERFORMANCE TRENDS IN DISK DRIVE READ-CHANNEL INTEGRATED CIRCUITS

Each silicon process technology generation has allowed integration of more complex signal processing schemes into an affordable chip size and has been characterized by exponential trends in data rates, as shown in Fig. 1. The annual increase of almost 40% in disk drive read electronics data rates has outpaced the performance boost provided by technology scaling, because of microarchitectural improvements that can be accommodated in succeeding generations. Even so, stringent cost and power requirements make the implementation of disk drive signal processing architectures challenging, particularly coupled with the demand for increased transfer rates. The need for low cost dictates small die sizes, traditionally in the 10–25 mm^2 range and low-cost plastic packaging [3]–[8]. The requirement for inexpensive packaging, in turn, limits the power consumption to less than 2 W [9].

Detectors have traditionally occupied only a small portion of the read/write channel chip. For example, a 16-state Viterbi decoder commonly used in 2.5 V, 0.25 μm CMOS technology occupies about 0.8 mm^2 and dissipates 300 mW running at 500 MHz, when implemented in full custom design methodology [3]. CMOS feature sizes have been reducing by 30% in every technology generation, and a new generation was introduced every 2–3 years in the past decade. When the feature
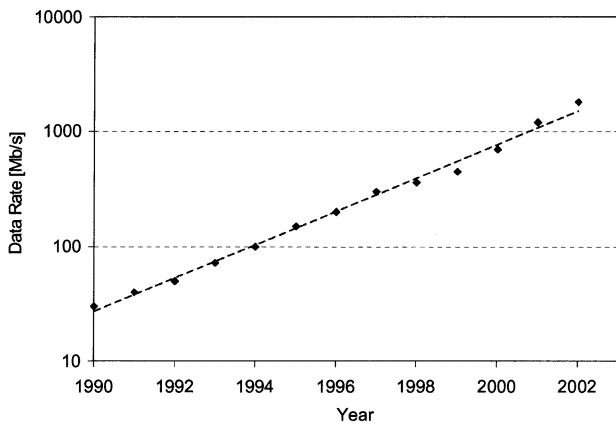
Fig. 1. Data rate trend in magnetic disk drive read channels (source: IEEE International Solid-State Circuits Conference and manufacturers' web sites).

sizes and the supply voltages are scaled by a factor of $S = 0.7$, the logic density approximately doubles (scales by a factor of $1/S^2$). Scaling of technology decreases the intrinsic transistor delay by a factor of S, thus increasing the frequency of the same implementation by a factor of $1/S$ [9]. Furthermore, scaling of the feature sizes and the voltages with the same factor preserves the active power density, and reduces the total active power proportionally to the area reduction. For example, when scaled to 1.8 V, $0.18 \mu m$, the 16-state Viterbi decoder in [3] would occupy 0.4 mm^2, run at 700 MHz and dissipate 150 mW.

On the other hand, analog front-ends do not scale at the same rate as the digital back-ends [10], so a potential size reduction of digital back-ends would not yield a proportional reduction in overall die size. As a result, the sizes of digital signal processing blocks have remained approximately constant. For example, detectors have generally occupied 7%–15% of the die area and dissipated 20%–40% of the total power [3]–[8]. Future read detectors, assuming similar power and area constraints, are likely to fit into similar proportions. Hence, instead of reducing the size of read-channel circuits, the technology scaling has been exploited to permit integration of more complex algorithms. The evolution of detectors illustrates this trend: 8-state conventional Viterbi decoders, common in $0.35 \mu m$ technology, were replaced with more complex 8-state noise-predictive Viterbi decoders or conventional 16-state decoders in $0.25 \mu m$. This was followed by 16-state noise-predictive decoders in $0.18 \mu m$. Currently, state-of-the-art $0.13 \mu m$ detectors incorporate 32-state noise-predictive decoders or 16-state decoders with postprocessing.

As the number of states increases, there is diminishing marginal improvement in bit-error-rate (BER) performance, despite an exponential growth in implementation complexity. The situation is thus ripe for more complex coding and signal processing techniques to challenge the PRML class of detectors. In the remainder of this paper, we attempt to identify the leading iterative detection candidates that can supply significant performance improvement at reasonable complexity and cost.

III. PERFORMANCE ANALYSIS

Ryan [11] published the first detailed investigation of applying iterative detectors to magnetic recording. Since then, a large number of investigators have examined a variety of iterative detection schemes. We screen this large number of choices based on some quick, coarse analysis. For example, both maximum *a posteriori* decoding based on the Bahl–Cocke–Jelinek–Raviv algorithm (MAP/BCJR) and the soft-output Viterbi algorithm (SOVA) have been considered for the inner (PRML) soft-input, soft-output decoder [12]. We have found in our analysis that the performance benefit that the BCJR possesses over SOVA is overcome by the large complexity penalty (the size and power dissipation of the BCJR decoder are approximately two times larger than for a SOVA decoder), so we consider here only the SOVA option. Iterative detectors based on convolutional codes are also recognized to be too complex to justify the modest extra performance they offer. For example, in order to meet the throughput requirements in disk drive applications, four iterations of turbo decoding with a convolutional code will require an unrolled architecture comprising eight 16-state SOVA decoders and seven interleavers [12]. This results in a more than $20\times$ increase in area and power over a conventional 16-state noise-predictive Viterbi decoder. Hence, we limit this discussion to iterative detectors based on parity checks, as we expect them to have lower complexity:

- random low-density-parity-check (LDPC) code with rate 8/9 and column weight 3 [13];
- structured LDPC code with rate 0.94 and column weight 4, designed from an integer lattice [14];
- turbo product code (TPC) based on single parity checks, with sixteen 16×16 matrices (rate 8/9) [15].

Coded bits of the TPC were randomly permuted (interleaved) before being precoded and sent to the channel. For both random and structured LDPC codes, we used a $1/(1 \oplus D)$ precoder, while for the TPC, a better BER was obtained using a $1/(1 \oplus D^2)$ precoder.

In order to place the results in context, we also include performance and complexity estimates for a simple 16-state Viterbi decoder and for an advanced PRML decoder with noise-prediction in its 16-state Viterbi trellis and parity-based post-processing, as described in [16]. Noise-predictive decoders with parity post-processing can be also used for complexity comparison: at the point of their introduction, they occupied 1–2 mm^2 of silicon area while dissipating less than 500 mW.

The iterative decoding algorithm is similar for the TPC and LDPC codes. The SOVA is used on the PRML trellis, and the message passing algorithm (MPA) is used for soft decoding of the TPC and LDPC codes. The decoding is established by iterating between the inner decoder and the outer decoder. That is, the soft information is extracted from a partial response channel using the SOVA operating on the channel trellis, and then used in the MPA for the LDPC or TPC decoding [15]. The bit "likelihoods" obtained from the MPA are passed back to the SOVA as *a priori* probabilities and so on. The BER results are given below for an architecture in which the compound iteration "SOVA $+I$ iterations of MPA" is performed G times before the final hard decision is taken ($I = 4$ for the LDPC and $I = 2$ for the TPC). In other words, the LDPC decoder performs four internal (bit-to-check plus check-to-bit) iterations, while the TPC decoder performs only two internal (rows followed by columns) iterations prior to looping back to supply the inner decoder with extrinsic information. Larger values of G improve the BER, but
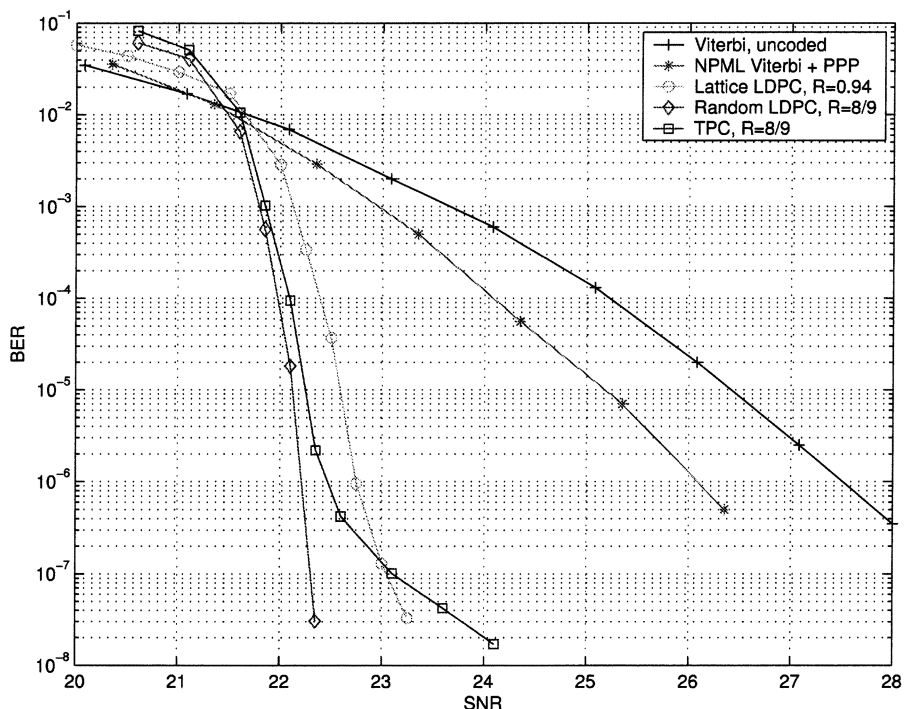
Fig. 2. BER versus input SNR for several detectors.

at a decreasing rate. Thus, a value of $G = 4$ is chosen as a compromise between poorer performance and increased delays and complexity.

The analog part of the magnetic recording channel is modeled using linear superposition of Lorentzian pulses. Formally, the readback signal $V(t)$ of the channel without jitter is defined as $V(t) = \Sigma_k b_k g(t - kT_b) + n(t)$, where $g(t)$ is the transition response of the channel, $b_k = \{0, 1, -1\}$ corresponding to no, positive, or negative transitions, and $n(t)$ is additive white Gaussian noise. The transition response of the longitudinal recording channel is modeled as a Lorentzian pulse $g(t) = V/[1 + (2t/\text{PW}_{50})^2]$, where PW_{50} is the pulsewidth at 50% of its peak value, and $V = \sqrt{2\text{PW}_{50}/\pi}$, so that the energy of an impulse response is equal to one. We define the SNR as $10 \log_{10}(E_i/N_0)$, where E_i is the energy of an impulse response of the channel and N_0 is the spectral density height of the additive white Gaussian noise, so that we can compare the efficiencies of different codes operating at different symbol densities. The code rate loss is reflected through the detection signal-to-noise ratio, which can be shown to be approximately proportional to R^2, where R is the code rate.

In the current generation of disk drives the user data are encoded by an ECC encoder, an RLL encoder, and can also be precoded before being recorded onto the disk. Therefore, the BER at the output of the channel detector and the BER after the ECC decoder should both be taken into consideration. In fact, the sector failure rate (SFR) is an even more appropriate metric for characterizing the read/write channel. In practice, it is difficult to measure the SFR directly in the operating range, and therefore it is usually estimated using various mathematical models. Since an accurate estimation technique for the iterative detection channels is still to be developed and thoroughly verified, in this paper we use the classical BER at the output of the iterative detectors.

The performance analysis results are summarized in Fig. 2 for the normalized user density of 2.5 and the PR target [5, 4, −3, −4, −2] [17]. Using BER = 10^5 as a reference, the iterative detectors (LDPC and TPC) give an SNR benefit of 4.1–4.2 dB over the simple Viterbi and 3.0 dB over the sophisticated NPML-PP detector. Clearly, it is not possible to operate a real system close to the very steep cliff observed for the iterative detectors, so we cannot avail ourselves of the full 3 dB. As we can see from this figure, all three iterative detectors exhibit a steep drop in BER at an SNR where the PRML detector BER is about 10^{-2}. For the random LDPC, this cliff continues falling at least to the 10^{-8} range. The structured LDPC drops at a somewhat higher SNR, and shows evidence of a change in slope at about 10^{-7}. The TPC drops along with the random LDPC, but has a significant slope change at even higher BER, about 10^{-6}.

This striking feature of iterative detectors—the steep drop in BER versus SNR—leads us to reconsider the requirements imposed on the signal processing function by the overall disk drive or even operating system. We note that the usual rule-of-thumb used for PRML detectors—the SNR required to achieve a specified BER (for example, 10^{-5})—is not useful for iterative detectors because this occurs on such a steep slope. Here, even a small degradation in SNR leads to a very large degradation in BER. In actual operation, an iterative detector will be run at an SNR some margin above the steep cliff. At this SNR, the nominal BER might be immeasurably small. This leads to another observation: variations which simplify the hardware complexity and result in BER versus SNR curves with shallower slope do not necessarily perform worse in practice than the best detectors. If the BER obtained at the "operating SNR" is low enough, even a large difference in BER does not have much effect, assuming the raw errors are cleaned up by an outer ECC. Indeed, a shallower slope allows the system to monitor performance over time and flag regions that appear to be degrading. This very useful

diagnostic function is available in commercial PRML systems today.

Another feature of interest is the distribution of error counts per sector for various detection schemes. For PRML detectors, the error counts per sector appear to be smoothly distributed. It has been generally feared that iterative detectors would produce very bursty error count statistics at the low BERs that would be used in practical systems. This results from the expectation that the MPA will decode to a valid codeword, and when this is not the correct codeword there will be at least as many errors as the $d_{\min}$ of the code. This is indeed the case for the TPC system. However, we have consistently observed that the failure mechanism for LDPC codes at low BER is not decoding to an incorrect codeword, but rather the MPA gets stuck at "near-codewords," as described by Gallager [18] and more recently by MacKay [19]. This may be due to a breakdown in the underlying assumption in the MPA of statistical independence of the data being passed around the graph. This is actually a desirable situation, since if the final result of the decoder is not a codeword, the system is alerted to the need for some kind of error retry procedure. For the random LDPC code used in our simulations, when a 4096-bit codeword has errors, the number is only on the order of 1 to 10 (over simulation runs totaling ∼100 000 codewords).

IV. IMPLEMENTATION ISSUES FOR ITERATIVE DETECTION

Implementation of iterative detectors in today's 0.13 μm CMOS technology requires a larger area and power budget than implementation of conventional detectors. Here, we investigate several architectures for decoding TPC and LDPC codes and evaluate their complexity. Building blocks of each of the decoders were synthesized in standard cell methodology to obtain accurate estimates of power and area.

We assume the use of a SOVA decoder as a soft-input soft-output PRML channel decoder. In CMOS technology, there is a strong relationship between the speed, power, and area for implementing a function. As in Viterbi decoders, SOVA decoder throughputs are limited by the single-cycle add-compare-select (ACS) recursion in the algorithm. The speed of a SOVA decoder can be traded off for power and area savings through microarchitectural transformations, logic transformations, and circuit sizing of the ACS [20]. Fig. 3 shows the area-delay and power-delay tradeoffs for a 16-state NPML decoder, obtained through synthesis in 0.13 μm CMOS technology. As the cycle time is shortened, the logic is resized to meet the required timing. The impact of sizing on speedup of a given topology is limited, and eventually the microarchitecture has to be changed to achieve faster decoders. The kinks in Fig. 3 illustrate this effect. Each microarchitecture is optimized to perform the single-cycle recursion for a range of throughputs. A conventional implementation of the ACS recursion is the smallest and dissipates the least amount of power, but has the longest critical path. Speed can be increased by retiming the ACS recursion [21] or loop unrolling [22], at the expense of increased area and power. Fig. 3 also illustrates the effects of technology scaling on power, performance, and area of this detector. The size of the 16-state SOVA decoder is comparable to the size of the 16-state NPML decoder.

Implementation of the MPA requires the computation of bit-to-check and check-to-bit messages (marginalized posterior probabilities), and passing them between the check nodes and
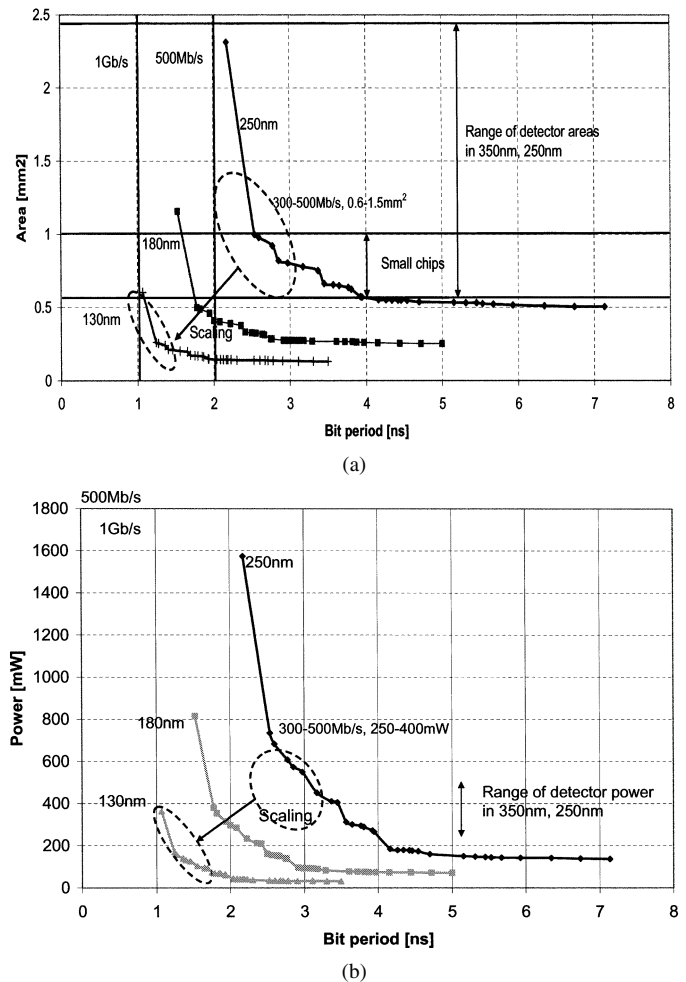

(a)

(b)

Fig. 3. (a) Area-delay tradeoffs and (b) power-delay tradeoffs in 16-state NPML decoders in 0.25 μm, 0.18 μm, 0.13 μm CMOS.

the bit nodes [23]. MPA-based decoding of LDPC and TPC codes allows for a variety of architectures with a wide range of throughputs and latencies. We examine the implementation challenges associated with fully parallel and fully serial LDPC decoder architectures, followed by a discussion of alternative architectures that fall between these two extremes.

A fully parallel implementation of the LDPC decoder directly maps the complete message passing graph onto the silicon [24]. A decoder for the random LDPC example analyzed in Section III would consist of 4608 bit nodes, 512 check nodes, and the interconnect network for passing the messages (Fig. 4). With the irregular structure typical of random LDPC codes, the area utilization of this implementation is expected to be about 50% because of the complexity of the interconnect network. The overall chip size is therefore estimated to be almost 160 mm^2. Though impractically large, this amount of parallelism would yield low clock frequencies and relatively low power; about 250 mW for four iterations at 1 Gb/s with less than one sector latency.

On the other hand, a fully serial LDPC decoder would use only one processing element for all message-passing calculations, and one large memory to store soft information for all the messages (13824 5-bit-wide words). The bottleneck in the serial architecture is the memory, which is required to provide three operands for each bit node or 27 operands for each check node
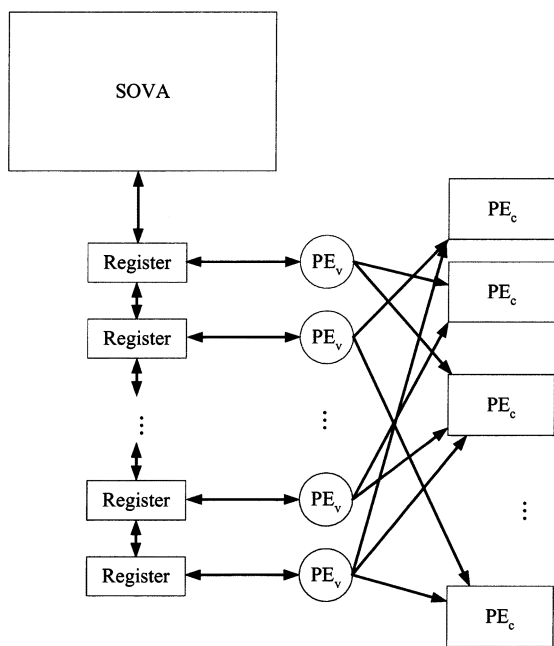
Fig. 4. Fully parallel LDPC decoder.

Fig. 5. Architecture of TPC decoder partitions the code into 17×17 blocks.

operation. This requirement implies that the memory has to operate at a multiple frequency of a bit node or check node. Adding more ports to the memory is impractical due to the quadratic increase in memory area. At best, a double-ported memory of this size operates up to 400 MHz, and completes the number of accesses required for four iterations of decoding in a latency equivalent to 15 sector delays. Such an architecture is dominated by the memory size, estimated to be 9 mm^2.

Alternative architectures that fall between a fully parallel or fully serial structure are impractical because of the irregular nature of random LDPC codes. This random connectivity prevents partitioning the memory in a decoder into sub-blocks that can be accessed independently. As such, increasing the number of processing elements will require additional memory ports, thus further slowing the memory access and increasing the size of the memory.

The desire to overcome the above impediments leads us to structured LDPC codes. These codes allow partitioning of the large message-passing graph into several smaller graphs, which allows for architectures with limited parallelism. Unfortunately, this results in somewhat poorer BER performance. TPC codes and LDPC codes based on lattices are examples of structured LDPC codes, where several smaller parity-check codes are combined to form a larger block. The TPC example divides the sector into sixteen 17×17 blocks, and the MPA can be implemented for each of these blocks using a single processing element (Fig. 5). To achieve a one-sector latency with minimum area, a total of eight parallel MPA are used with double-ported memories in the interleavers. The parallel TPC decoder logic takes only about 0.2 mm^2 and dissipates about 150 mW at 1 GHz, with four iterations. However, we observe that the performance of TPC codes is only acceptable if global iterations are performed between the inner SOVA and outer TPC decoders: the iterations within the TPC decoder provide insufficient BER improvement. Thus, the power and the area of the overall itera-
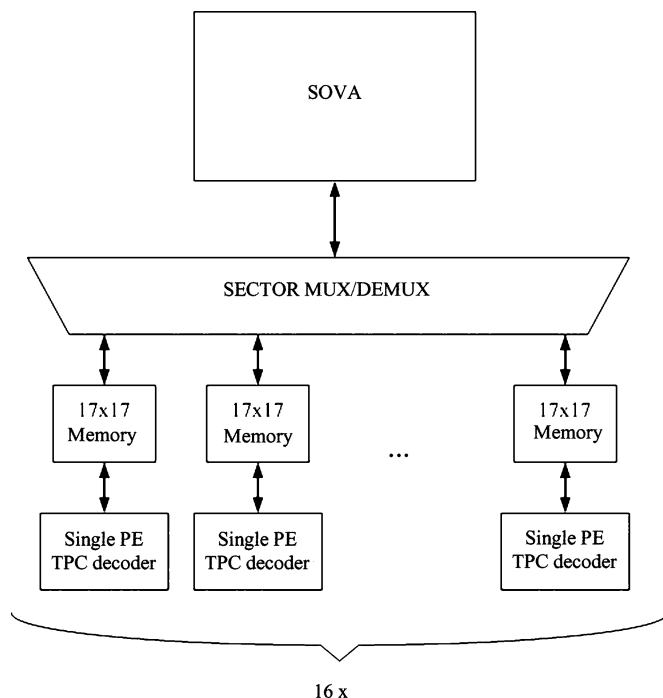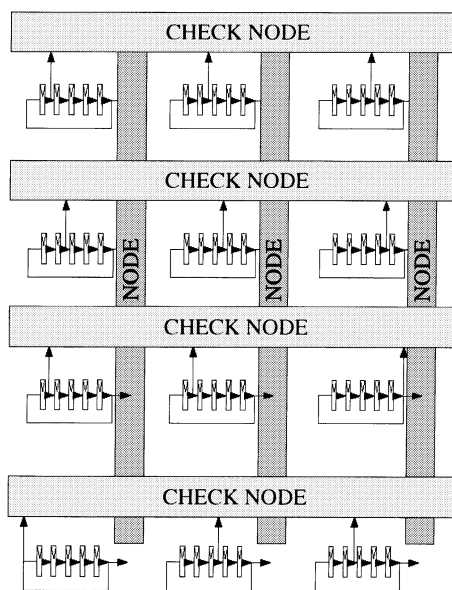

Fig. 6. Shift-register structure for LDPC decoders based on lattices.

tive detector, shown in Fig. 5, will be dominated by the SOVA decoders and the interleavers.

LDPC codes based on lattices display the unique property that consecutive rows in each of the sub-blocks are cyclic. Each row of the sub-block also has a weight of one. Fig. 6 shows a structure that exploits these properties with shift-registers and regular arrangement of the processing elements. The shift registers permit operation at clock rates close to 1 GHz. The regularity of the decoder enhances the logic density, resulting in a decoder that is approximately 3 mm^2 and dissipates about 2 W with four iterations of decoding at 1 GHz. Although this architecture does not require explicit interleavers, the area is large because it uses registers for storing messages.
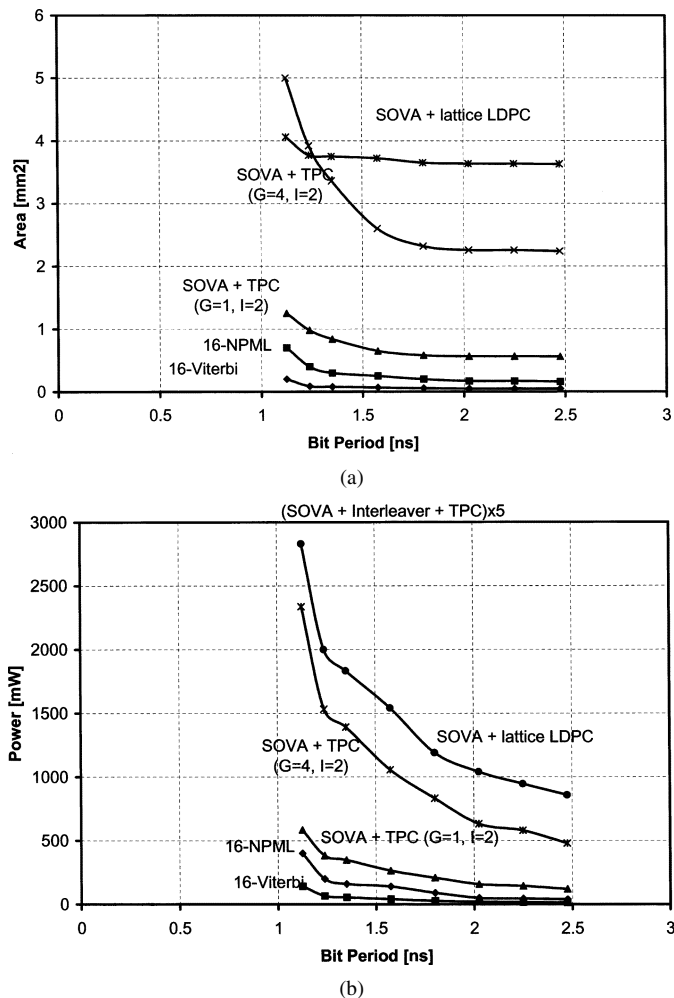
Fig. 7. (a) Area-delay and (b) power-delay tradeoff for iterative detector architectures in 0.13 μm.

Fig. 7 summarizes the area-delay and power-delay tradeoffs for the TPC decoder with one and four global iterations and the lattice LDPC decoder with one global iteration, and compares them to 16-state Viterbi and NPML decoders [3].

V. CONCLUSION

The curves in Fig. 7 show that the implementation of high-speed iterative detectors in 0.13 μm technology is not feasible in the historic cost and power read channel budgets. Even detectors running at lower speeds (below 500 Mb/s) would still be several times larger than NPML detectors at the technology point of their introduction. However, implementation in 90 nm CMOS technology would lower the area and power requirements of iterative detectors by a factor of two. This, in addition to the 40% improvement in speed, would make high-throughput iterative detectors feasible.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *IEEE Int. Conf. Communications, Conf. Record*, vol. 2, Geneva, Switzerland, May 23–26, 1993, pp. 1064–1070.

[2] Z. Wu, *Coding and Iterative Detection for Magnetic Recording Channels*. Norwell, MA: Kluwer, 2000.

[3] N. Nazari, "A 500 Mb/s disk drive read channel in 0.25 μm CMOS incorporating programmable noise predictive Viterbi detection and trellis coding," in *IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 2000, pp. 78–79.

[4] S. Sutardja, "360 Mb/s (400 MHz) 1.1 W 0.35 μm CMOS PRML read channels with 6 burst 8–20x over-sampling digital servo," in *IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 1999, pp. 40–41.

[5] T. Pan et al., "A trellis coded E^{2}PRML digital read/write channel IC," in *1999 Int. Solid-State Circuits Conf., ISSCC'99 Dig. Tech. Papers*, San Francisco, CA, Feb. 15–17, 1999, pp. 36–37.

[6] M. Leung et al., "A 300 Mb/s BiCMOS EPR4 read channel for magnetic hard disks," in *1998 Int. Solid-State Circuits Conf., Dig. Tech. Papers*, San Francisco, CA, Feb. 5–7, 1998, pp. 378–379.

[7] M. Demicheli et al., "A 450 Mbit/s EPR4 PRML read/write channel," in *Proc. 1999 IEEE Custom Integrated Circuits Conf.*, San Diego, CA, May 16–19, 1999, pp. 317–320.

[8] H. Thapar et al., "Hard disk drive read channels: technology and trends," in *Proc. 1999 IEEE Custom Integrated Circuits Conf.*, San Diego, CA, May 16–19, 1999, pp. 309–316.

[9] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[10] H. Samueli, "The broadband revolution," *IEEE Micro*, vol. 20, pp. 16–26, Mar.–Apr. 2000.

[11] W. E. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," in *IEEE Int. Conf. Communications, Conf. Record*, vol. 2, June 7–11, 1998, pp. 947–951.

[12] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "VLSI architectures for iterative decoders in magnetic recording channels," *IEEE Trans. Magn.*, vol. 37, pp. 748–755, Mar 2001.

[13] J. Fan, A. Friedmann, E. Kurtas, and S. W. McLaughlin, "Low density parity check codes for partial response channels," in *Allerton Conf. Communications, Control and Computing*, Urbana, IL, Oct. 1999.

[14] B. Vasic, E. Kurtas, and A. Kuznetsov, "LDPC codes based on mutually orthogonal Latin rectangles and their application in perpendicular magnetic recording," *IEEE Trans. Magn.*, pt. 1, vol. 38, pp. 2346–2348, Sept. 2002.

[15] J. Li, K. R. Narayanan, E. Kurtas, and C. N. Georghiades, "On the performance of the high-rate TPC/SPC and LDPC codes over partial response channels," *IEEE Trans. Commun.*, vol. 50, pp. 723–735, May 2002.

[16] R. D. Cideciyan, J. D. Coker, E. Eleftheriou, and R. L. Galbraith, "Noise predictive maximum likelihood detection combined with parity-based post-processing," *IEEE Trans. Magn.*, vol. 37, pp. 714–720, Mar. 2001.

[17] H. Sawaguchi, S. Mita, and J. Wolf, "A concatenated coding technique for partial response channels," *IEEE Trans. Magn.*, vol. 37, pp. 695–703, Mar. 2001.

[18] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[19] D. J. C. MacKay and M. S. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes," *Electronic Notes in Theoretical Computer Science*, vol. 74, 2003.

[20] E. Yeo, S. Augsburger, W. R. Davis, and B. Nikolic, "A 500 Mb/s soft output Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 38, pp. 1234–1243, July 2003.

[21] G. Fettweis, R. Karabed, P. H. Siegel, and H. K. Thapar, "Reduced-complexity Viterbi detector architectures for partial response signaling," in *Proc. IEEE Global Telecommunications Conf.*, Singapore, Nov. 13–17, 1995, pp. 559–563.

[22] P. Black and T. Meng, "A 140 MB/s 32-state radix-4 Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1877–1885, Dec. 1992.

[23] E. Yeo, B. Nikolic, and V. Anantharam, "Iterative decoder architectures," *IEEE Commun. Mag.*, vol. 41, pp. 132–140, Aug. 2003.

[24] A. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-$\frac{1}{2}$ low-density parity-check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, pp. 404–412, Mar. 2002.