# Design and Performance of a VLSI 120 Mb/s Trellis-Coded Partial Response Channel

J. W. Rae and G. S. Christiansen
IBM Storage Systems Division, Rochester, MN 55901

P. Siegel and R. Karabed
IBM Research Division, Almaden Research Center, San Jose, CA 95120

H. Thapar and S. Shih
IBM Storage Systems Division, San Jose, CA 95193

***Abstract*** **A VLSI trellis-coded partial-response (TCPR) codec module incorporating circuitry for three exploratory, improved trellis codes is presented. The architecture, implementation, and performance evaluation of a "partitioned" rate 8/10 matched-spectral-null (MSN) code are described in detail. The detector implementation uses a programmable, systolic architecture to realize a time-varying Add-Compare-Select (ACS) topology that rotates state-ACS assignments and periodically superimposes "auxiliary" survivor paths. The survivor memory unit requires only 3 bytes per interleave and is implemented using a two-stage, time-varying architecture combining register-exchange and traceback methods. The module was fabricated in 0.45 micron CMOS technology requiring less than 300 mW at 12 MB/sec. Selected experimental results for a variety of magnetic recording components are shown.**

## I. Introduction

The potential performance advantages of a trellis-coded PRML (TCPR) channel have been confirmed experimentally using a prototype matched-spectral-null (MSN) trellis codec chip operating at 30 MHz [1], [2], [3]. Some of the characteristics of the exploratory trellis code embodied in that chip, however, made it unsuitable for use in a commercial magnetic recording channel. In particular, the Viterbi detector architecture, based upon a novel pipelined 2-state emulation of the 6-state, 2-step trellis structure, stressed area-efficiency at the expense of speed. Also, the length 64-bit path memory and register-exchange survivor management approach had negative implications for circuit size, error-correction code redundancy, and track format overhead requirements.

To improve in these areas, new trellis code design methodologies were developed, as reported in the companion paper [4]. In addition, improved methods of constructing the detector were discovered [5]. In this paper, we report on a second generation experimental TCPR module which incorporates the new circuit and code design methodologies. The purpose of the module was to provide a test vehicle to measure the performance and the feasibility of the new techniques, and to provide prototype hardware for further development.

The TCPR module was designed to be used with an existing Class-4 partial-resonse (PR4) channel to provide a "fully functional" trellis-coded PR channel at data rates up to 15 MB/s. This means that both encoded write and read paths are supported using 512 byte sector format. No additional external control signals are required. Encode, decode, detectors, and support circuitry were constructed for three new codes: an improved rate 8/10, MSN code with partitioned trellis structure, a rate 8/10 "permuted" trellis code, and a rate 16/18 code (detector only).

Because the performance of the "partitioned" MSN(8/10) code was the most thoroughly evaluated of the three codes, only the design and implementation of hardware for the MSN data path will be discussed [6]. The design of detectors for the other two codes is similar.

The discussion of the MSN(8/10) implementation is divided into three areas: circuit design architecture, VLSI design methodology, and selected measured performance results.

## II. Architecture and Circuit Design

### A. Detector Architecture

Like the code in the first generation TCPR system, the new codes are designed for a $(1-D)$ channel. When applied to PR4 $(1-D^2)$ channel, input data is encoded and bit-wise interleaved before being written to the media. In the readback and detection process, two parallel Viterbi detectors operate independently on alternate even and odd bits. Each detector produces a "most likely" estimate of the encoded input data sequence. Detector outputs are decoded into bytes and sequentially re-interleaved to produce estimated input data (customer data).

The overall architecture of the "partitioned" MSN(8/10) detector is shown in Fig. 1. A single detector could have been used for both interleaves by adding pipeline latches. This design would use an estimated 38% less logic at the cost of 25% lower maximum overall data rate. Only the encode, decode, and detector functions need to be changed from an existing PR4 channel to realize the SNR gain offered by the trellis code.

Manuscript received June 24, 1994.

G.S. Christiansen, e-mail grant@rchvmp3.ibm.vnet.com, fax 507-253-4146; P.H. Siegel, e-mail siegelph@almaden.ibm.com, fax 408-927-3030; R. Karabed, fax 408-927-3520; J. Rae, H. Thapar, and S. Shih, now at DataPath Systems, Inc., Santa Ana, CA 92707, fax 714-546-4245.

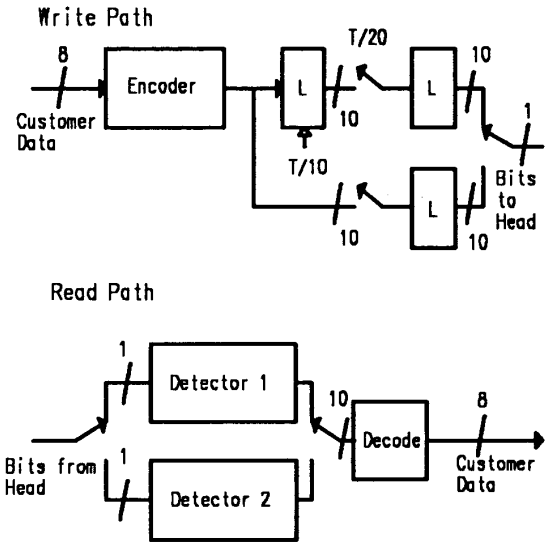


Figure 1. Interleaved detectors.

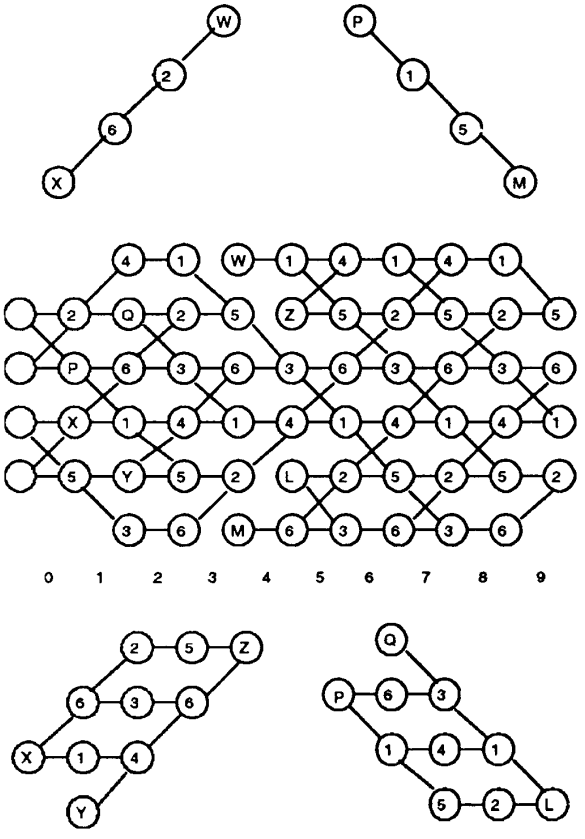


Figure 2. Partitioned MSN(8/10) detector trellis.

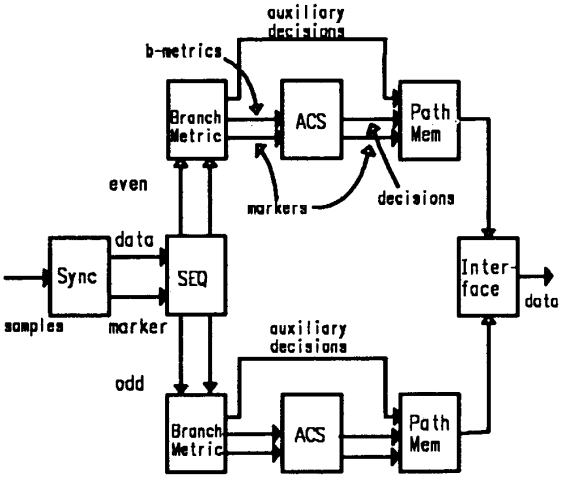


Figure 3. Detector block diagram.

The detector trellis structure for the MSN(8/10) code is shown in Fig. 2. It can be seen that the structure is defined by a time-varying "flat" 6-state trellis over which 4 auxiliary paths are superimposed. The auxiliary trellis paths join the flat trellis at bit times 3 and 4 (modulo 10). In this respect, the trellis is effectively three dimensional, varying in time, repeating every 10 bit periods. The state label (modulo 2) is the encoded input symbol leading to the state. The paths traced on the trellis diagram describe the set of sequences from which all codeword sequences are selected.

Because of the time-varying properties of the trellis, some new features are added to the detector as shown in Fig 3. The TCPR Viterbi detector now contains a synchronizer function, which is required to define codeword boundaries, vary the trellis as a function of time, and to initiate special start-up and shut-down sequences at the beginning and end of a read sequence.

Each node in the trellis at time *n* represents a 2-input add-compare-select (ACS) unit, which functions by adding a branch metric to the path metric connecting the node along the horizontal path. This sum is compared to the path metric connecting diagonally to the node (the diagonal branch metric is zero, so no addition is required). The smaller metric is selected and stored along with the decision (horizontal or diagonal).

The label contained within each node of the trellis represents the ACS unit assigned to that state for a given time n. States are numbered 1 through 6 from top to bottom and do not change with time. State assignments are different from ACS assignments. With the ACS-to-state-mapping indicated, the inputs to each ACS unit are preserved. That is, ACS 6 always chooses between ACS 3 and 4, ACS 3 always chooses between ACS 5 and 6 etc. If the trellis were completely homogeneous, the need for any input multiplexers would be eliminated. This circuit topology is called "rotate by three" [7]. The path metrics for a given

state are passed back and forth between alternate ACS units. The time-varying nature of the trellis is effected by adding an override feature to each ACS, controlled by programmable bits which force a predetermined decision and corresponding metric update.

### B. Design Equations

Given the "rotate by three" topology, the calculations needed to implement a Viterbi detector for the MSN(8/10) code are defined by the trellis diagram and are described as follows.

Define:

$M_n(k) \equiv$ path metric stored at time n in ACS unit k

$y_n \equiv$ input sample values at time n

$$b_n^{\pm} \equiv .5 \pm y_n. \tag{1}$$

Then, update the survivor metrics according to:

$$M_{n+1}(1) = \min\{M_n(4) + b_n^-, M_n(3)\} \tag{2}$$

$$M_{n+1}(2) = \min\{M_n(5) + b_n^+, M_n(6)\} \ \mathrm{mod}_{10}(n) \neq 4 \tag{3}$$

$$M_{n+1}(3) = \min\{M_n(6) + b_n^-, M_n(5)\} \ \mathrm{mod}_{10}(n) \neq 3 \tag{4}$$

$$M_{n+1}(4) = \min\{M_n(1) + b_n^+, M_n(2)\} \ \mathrm{mod}_{10}(n) \neq 3 \tag{5}$$

$$M_{n+1}(5) = \min\{M_n(2) + b_n^-, M_n(1)\} \ \mathrm{mod}_{10}(n) \neq 4 \tag{6}$$

$$M_{n+1}(6) = \min\{M_n(3) + b_n^+, M_n(4)\} \tag{7}$$

The auxiliary trellis computations are described as follows.

When $n = 4$ (modulo 10), define:

$$ab_n^+ \equiv 1 + \min\left\{\begin{matrix} y_n - y_{n-1}, \\ y_{n-1} - y_{n-2}, \\ y_{n-2} - y_{n-3} \end{matrix}\right\} \tag{8}$$

$$ab_n^- \equiv 1 + \min\left\{\begin{matrix} y_{n-1} - y_n, \\ y_{n-2} - y_{n-1}, \\ y_{n-3} - y_{n-2} \end{matrix}\right\}. \tag{9}$$

Then, set:

$$M_{n+1}(2) = \min\{M_{n-2}(2), M_{n-3}(4) + ab_n^+\} \tag{10}$$

$$M_{n+1}(5) = \min\{M_{n-2}(5), M_{n-3}(3) + ab_n^-\}. \tag{11}$$

When $n = 3$ (modulo 10), set:

$$M_{n+1}(4) = M_{n-2}(4) \tag{12}$$

$$M_{n+1}(3) = M_{n-2}(3). \tag{13}$$

A time-varying trellis topology is obtained by overriding the selections in equations (2) through (7) using programmed control bits. For clarity, the override conditions have been omitted from the equations, but can be easily deduced from the trellis diagram, Fig. 2. The capability to program ACS decisions allows several different trellis geometries to be implemented with the same hardware, and enables the trellis structure to be varied as a function of time, history, or sector position.

### C. Circuitry

This section deals with the hardware used to obtain a synthesis of the design equations (1) to (13).

The detector is functionally partitioned into four areas, as shown in Fig. 3: a synchronizer/sequencer function, a branch metric function, a 6-state Add-Compare-Select (ACS) processor, and a survivor path memory manager function, which in turn is subdivided into "local" and "global" sections.

The synchronizer/sequencer structure performs a number of different tasks which are vital to the operation of a time-varying trellis detector. The synchronizer partition implements a 2-state Viterbi detector for the $1 - D$ channel based on a dynamic threshold, as well as a sync mark correlator to find a unique 40 bit sync mark that is written at the beginning of a sector to indicate the start of read-back data. As soon as start of data is determined (n = 0), codeword boundaries are defined and detector circuitry is initialized.

The sequencer function generates control bits that are used to program the ACS units of each detector. Three different trellis topologies are used during a sector read operation. At the beginning of a read, a special start-up trellis is used; then is changed to the MSN(8/10) code topology, and at the end of a sector a special shut-down trellis is defined. Unique start-up and shut-down topologies are needed to minimize errors which would be generated when there is insufficient sample history to make low-error decisions.

The branch metric partition calculates branch-metrics according to equation (1), and is shown in Fig. 4. Input sample values, $y_n$, are converted from 6-bit wide offset binary to 2's-complement and translated to produce $b_n^+$ and $b_n^-$. An additional function of this partition is to calculate auxiliary branch metrics according to equations (8) and (9), using a 7-bit recursive minimum-finder circuit. Branch metrics, auxiliary branch metrics, and auxiliary trellis paths are passed on to the ACS and survivor memory manager for further processing.

The ACS processor consists of six programmable ACS units, and is shown in Fig. 5. As mentioned previously, each ACS unit is assigned to 2 states with a "rotate by 3" twist; that is, each ACS alternates between two states whose state labels are equal (modulo 3). In this implementation four multiplexers are used to merge the auxiliary trellis when $n = 4$ (modulo 10) according to equations (10) and (11). Ten-bit precision was chosen for the ACS units and path metrics. When 2's complement arithmetic is used, explicit re-normalization is not needed provided that the precision used (10 bits) allows a path metric value that is at least twice as large as the largest possible path metric difference [8]. Path metrics can be allowed to wrap, and the add and compare arithmetic proceeds without error.

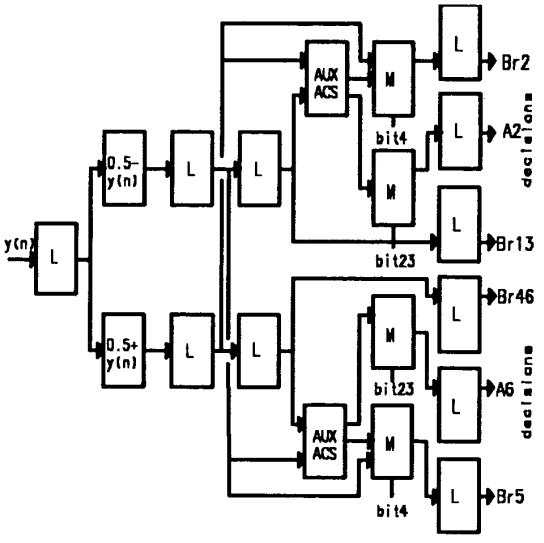


Figure 4. Branch metric calculations.

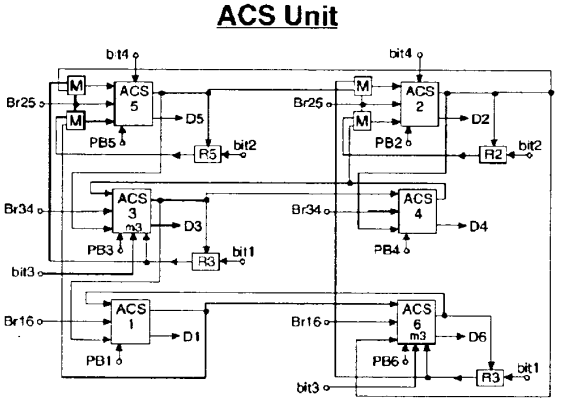


Figure 5. ACS function.

The 10-bit wide ACS unit circuits are constructed of cascaded 2-bit wide adder elements. The add and compare circuits are concatenated with element delays matched so that the add-carry and the compare-carry propagate in parallel. This means that the compare operation adds only one gate delay to the ACS operation. Worst-case delay for an ACS operation using 0.45u (Leff) technology was 8.4 ns (excluding latch set-up and delay).

The survivor path memory management section functions to determine one encoded data sequence to be generated as the "maximum likelihood estimate," based on a history of decisions supplied by the ACS partition. A block diagram of the path memory function is shown in Fig. 6 and Fig. 7.

The local path memory, which processes the first 10 bits, uses a modified register exchange algorithm to calculate a set of four probable codewords, and a set of pointers which represent the parent state of each codeword. Auxiliary trellis path sequences, obtained from the branch metric processor, are stored and mapped into the local memory, depending on ACS decisions made during bit-times $n = 3$ (modulo 10) and At the end of every 10 bit period, when the trellis narrows to 4 states, four surviving codewords and pointers are passed to the global memory for storage, as shown in Fig. 8. By starting with the state pointers of the most current codeword, the parent states can be traced back to find a common ancestor state. The corresponding codeword that ends in the common state at the traced-back time is selected and generated as the "most probable" encoded data estimate.

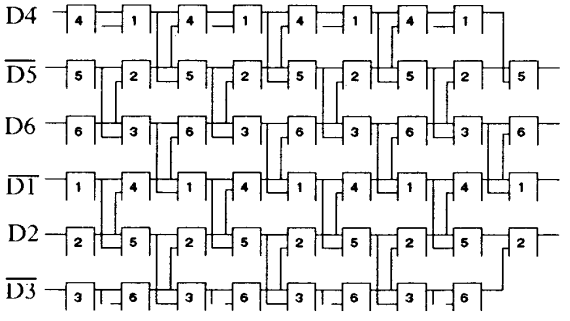


Figure 6. Register exchange local path memory.

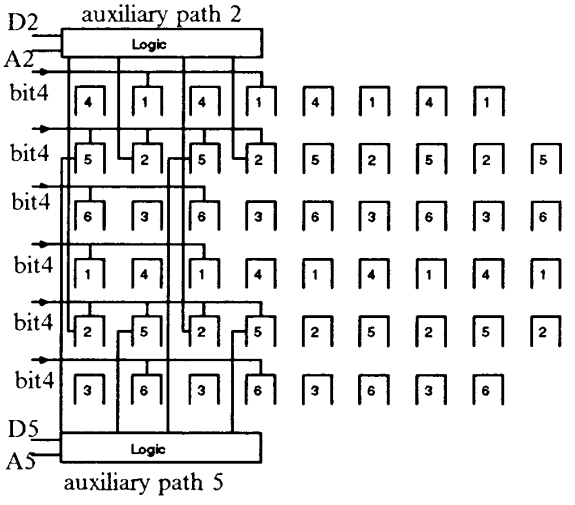


Figure 7. Local path memory with auxiliary path modification.

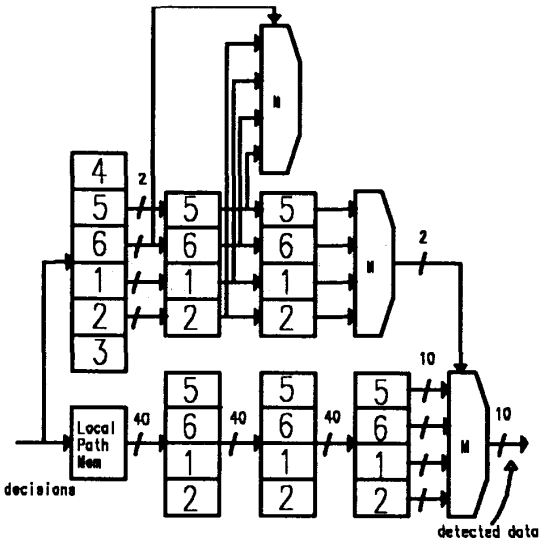


**Figure 8. Global path memory.**

*D. Encoder-Decoder Description*

The rate 8/10, partitioned MSN code is a dc-free code. It limits runlengths of like NRZ symbols to 5 within codewords, and to 6 across codeword boundaries. Each codeword contains at least 3 occurrences of 01 or 10, a property that improves timing recovery and gain control. The rate 8/10 encoder is a 2-state finite-state machine encoder. The code is designed to be invariant with respect to 180 degree phase shift of the readback signal. The decoder is a block decoder; i.e., it decodes each 10-bit codeword independently into a data byte, thereby limiting error propagation.

The assignment of 8-bit data bytes to codewords was selected in order to simplify the Boolean logic equations that implement the encoding and decoding functions. It exploits the natural specification of codewords as concatenations of 5-bit sequences subject to constraints involving their running-digital-sums (the number of 1's minus the number of 0's), as described in the companion paper [4]. The total gatecount for the logic was approximately 305 2-NOR equivalents.

## III. VLSI Design Methodology

The module was fabricated in 0.45u Leff CMOS standard cell technology available from IBM Microelectronics. The logic description was coded entirely in VHDL and VHDL macros. The macros expand into standard VHDL to instantiate commonly used logic functions. Speed-critical areas of the design were mapped directly using VHDL directives, while logic synthesis was used elsewhere. Completed VHDL was compiled and synthesized into a library of standard static CMOS logic functions. No custom logic was used.

High-level behavioral descriptions of the detectors were written in APL and in C language and were used to verify through simulation the expected coding gain. The VHDL compiler generated BDL/S, a netlist from a library of books, which served as input for automatic place-and-route and timing verification tools. A BDL/S netlist simulator was used to verify functionality and to close on higher level behavioral models. The design adhered to full LSSD design for testability scan guidelines. DC stuck-at fault test coverage was 99.7%

The specifications relating to circuit complexity, power, and speed are as follows:

| | |
|---|---|
| complexity = | 5028 2-NOR equivalents (parallel)<br>3116 2-NOR equivalents (pipelined) |
| power = | 279 mW @ 12MB/s nominal (parallel)<br>55.8 mW @ 20% duty cycle |
| speed = | 15 MB/s nominal<br>12 MB/s worst case (parallel) |

## IV. Experimental Performance

Four experiments were performed to verify the performance advantages of TCPR over PRML with the new module. In the first experiment, additive white noise of various power levels was added to the equalized signal before the signal was digitized. Fig. 9 shows the bit error rate as a function of the noise level with the channel data rate held constant. The results clearly demonstrate the 3dB coding gain of TCPR over PRML over a wide range of noise level.

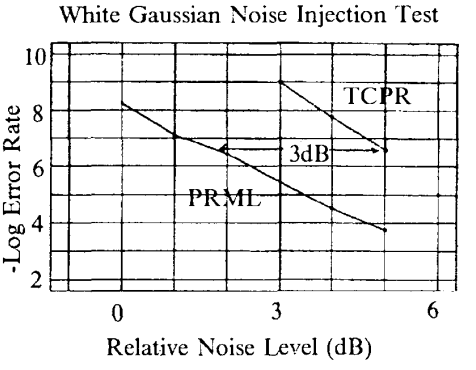


**Figure 9. Error rate as a function of injected white noise level.**

In the second experiment, three MR head/thin-film disk combinations were used to compare the error rate as a function of off-track position. Since the PRML system uses a rate 8/9 code and the TCPR system uses a rate 8/10 code, the channel clock speed was adjusted so that the user data rate for both systems was the same. The experimental results are shown in Figs. 10, 11, and 12. The normalized off-track posi-

tion is defined as the ratio of the measured off-track to the magnetic write width. The off-track enhancement offered by TCPR can be used to allow a more robust mechanical positioning system or higher track density. The head/disk combination that gave the largest TCPR advantage had the poorest overwrite performance of the three. One possible interpretation of the result is that TCPR channel is less sensitive to overwrite degradation than the baseline PRML channel.

In the third experiment, TCPR was compared to PRML over a wide range of normalized user densities using an MR head and two disks of different defect densities. Here the normalized off-track capability was again used as the performance measure. The results in Fig. 13 show that the off-track enhancement was maintained for both defect densities even at a $PW50/T$ of 2.5, and that TCPR showed a larger advantage at higher defect densities.

In the fourth experiment, using the same components as in the third, TCPR performance was compared to that of PRML with the same algebraic error correction code (ECC) applied to both channels, at a fixed user data rate of 5 MB/s. The ECC used was a single-burst correcting, triple-interleaved Reed-Solomon code. The results are shown in Fig. 14 and Fig. 15. TCPR still provides a significant performance improvement over PRML, particularly in the presence of high disk-defect density.

## V. Concluding Remarks

The architecture, VLSI design, and experimental evaluation of a trellis-coded, Class-4 partial-response channel has been presented. The implementation and performance details were provided for a novel rate 8/10, partitioned MSN trellis code. The chip circuit complexity, power, and speed specifications demonstrate the feasibility of a practical TCPR system for future disk drives. Experimental results confirm the performance advantages of TCPR over PRML, and the benefits of TCPR were shown to be complementary to those provided by an outer algebraic ECC.

## Acknowledgments

The authors wish to thank T. Dillinger of IBM AS400 Division and E. Senn of IBM Microelectronics for support and encouragement.

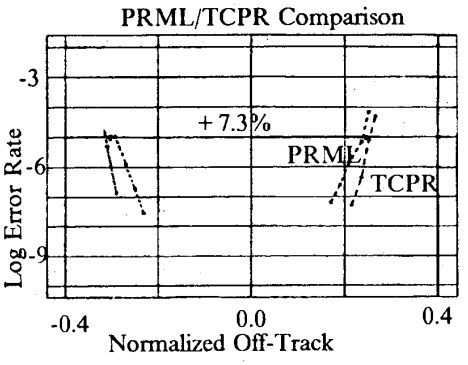


**Figure 10. Off-track error rate test results.**

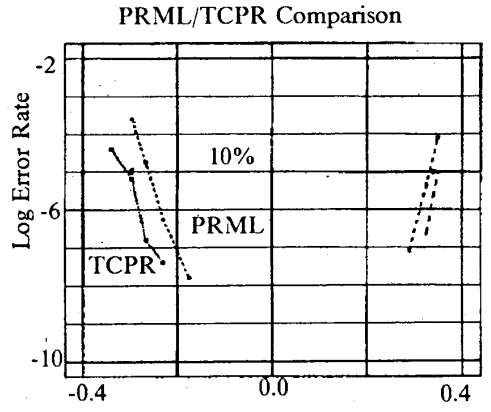


**Figure 11. Off-track error rate test results.**

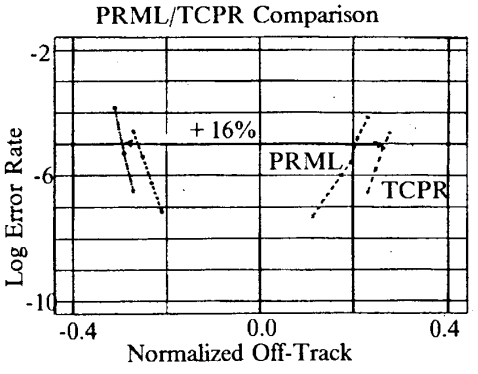


**Figure 12. Off-track error rate test results.**

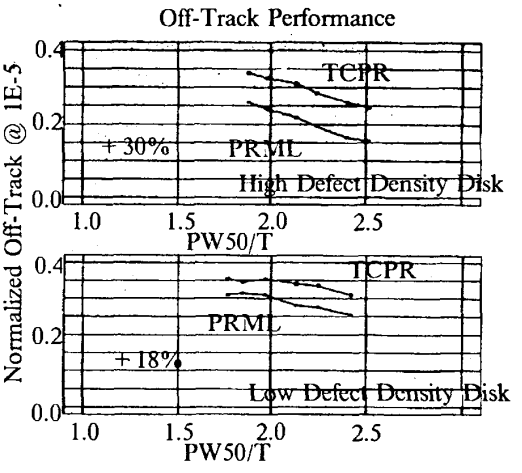


**Figure 13. Performance comparison over a range of densities.**

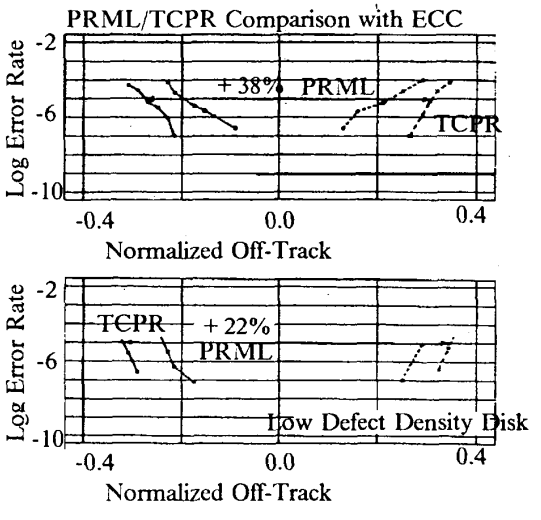


**Figure 14. Off-track performance comparison on low defect density disk, with ECC**

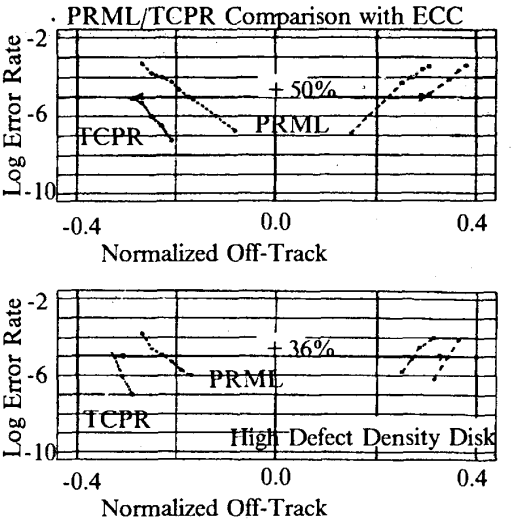


**Figure 15. Off-track performance comparison on high defect density disk, with ECC**

## REFERENCES

[1] H. Thapar, J. Rae, B. Shung, R. Karabed, and P. Siegel, "On the performance of a rate 8/10 matched spectral null code for Class-4 partial response," *IEEE Trans. Magn.*, vol. 28, no. 5, pp. 2884-2889, September 1992.

[2] H. Thapar, B. Shung, J. Rae, R. Karabed, and P. Siegel, "Real-time recording results for a trellis-coded partial response (TCPR) system," *IEEE Trans. Magn.*, vol. 29, no. 6, pp. 4009-4011, November 1993.

[3] B. Shung, P. Siegel, H. Thapar, and R. Karabed, "A 30-MHz trellis codec chip for partial-response channels," *IEEE J. Solid-State Circuits, Special Issue on Analog and Signal Processing Circuits*, vol. 26, no. 12, pp. 1981-1987, December 1991.

[4] L. Fredrickson, R. Karabed, J. Rae, P. Siegel, H. Thapar, and R. Wood, "Improved trellis-coding for partial response channels," *Proceedings of 1994 The Magnetic Recording Conference*, San Diego, California, August 1994.

[5] L. Fredrickson and J. Rae, "Systolic Array Architecture for Trellis-Code Viterbi Detectors," U.S. Patent No. 5,327,440.

[6] R. Karabed, J. Rae, P. Siegel, H. Thapar, "Matched Spectral Null Codes With Partitioned Systolic Trellis Structures," IBM patent application filed.

[7] G. Fettweis, R. Karabed, P. Siegel, and H. Thapar, "Reduced-Complexity Viterbi Detector Architectures for Partial-Response Signalling," IBM Research Report 9160, January 1993, submitted to *1995 IEEE Int. Conf. Commun.*.

[8] P. Siegel, C. Shung, T. Howell, and H. Thapar, "Exact Bounds for Viterbi Detector Path Metric Differences," *Proc. 1991 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Toronto, May 1991, pp. 1093-1096.