On the Performance of a Rate 8/10 Matched Spectral Null Code for Class-4 Partial Response

H. Thapar[a], J. Rae[b], C. Shung[c,d], R. Karabed[c], and P. Siegel[c]

[a]IBM Storage Systems Products Division, San Jose, CA 95193
[b]IBM Storage Systems Products Division, Rochester, MN 55901
[c]IBM Research Division, Almaden Research Center, San Jose, CA 95120
[d]National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Abstract-- **We present the application and performance of a rate 8/10 Matched-Spectral-Null (MSN) trellis code in conjunction with Class-4 partial response (PR4) signalling. The VLSI implementation of an MSN code is described. Results of experiments using synthesized PR4 waveforms are presented and it is shown that the rate 8/10 code can provide about a 3dB advantage relative to the Class-4 partial response maximum likelihood detection (PRML) in the presence of additive white Gaussian noise as well as additive synthesized adjacent track interference.**

I. INTRODUCTION

Class-4 partial response (PR4) signalling with maximum likelihood detection $[1-3]$ (PRML) has been experimentally shown [4] to provide a substantial performance improvement on magnetic recording channels relative to the peak detection method. On a given head-medium combination, PRML can provide acceptable performance over a range of recording density. Beyond that range, the performance degrades because of excessive noise enhancement in equalization, and other detection methods, such as decision-feedback equalization (DFE) [5] and extended PRML [6], or refinements to the PRML method are needed to maintain acceptable performance. In this paper, we focus on the latter approach, in particular, a trellis-coded PRML technique.

Specifically, we describe the application and performance of a rate 8/10 Matched Spectral Null (MSN) code [7] in conjunction with Class-4 partial response signalling. The MSN code imposes additional structure in the readback waveform to provide improved performance relative to PRML. Both theoretical and experimental results are given to show that, for a given symbol rate into the channel, the MSN/PR4 combination provides about a 3 dB advantage in signal-to-noise ratio relative to the PRML method in the presence of additive Gaussian noise. The same advantage is maintained when the Gaussian noise is replaced by synthesized adjacent track interference. Both of these measured results suggest that MSN coding represents a viable, evolutionary approach to enhancing the performance of the PRML method.

Section II gives an overview of MSN coding. Section III describes the specific MSN code used in a hardware prototype. Section IV examines some of the architectural considerations in the design of the experimental prototype. Section V describes the experimental results. Section VI is devoted to concluding remarks.

II. OVERVIEW OF MSN CODES

It has been long recognized [8] that the functions of coding and modulation must be combined to effectively utilize the available bandwidth and signal-to-noise ratio of a channel. Indeed, over the past decade, combined coding and modulation in the form of Trellis-coded Modulation (TCM) [9] has revolutionized the communications industry. Data rates within 15-20% of the Shannon Capacity are becoming commonplace over the ubiquitous voiceband channel. TCM codes are designed for transmitting multi-level symbols over memoryless channels. As such, they are not applicable to saturation recording, where only binary input symbols are allowed and the channel has memory which depends upon the linear density and the head/medium response.

Although fundamentally different in design, MSN codes capture the basic essence of TCM in that they provide a means of combining the functions of coding and modulation for partial response channels. They belong to the general class of trellis codes where the channel input-output is depicted by a trellis diagram. The detection requires the soft-decision Viterbi algorithm, which recursively traces the most likely recorded sequence through the trellis based on the observed noisy samples. Like other trellis codes for partial response channels [10,11], MSN codes are designed to enhance the minimum Euclidean distance between any pair of allowed sequences at the channel output relative to the uncoded system. Such a criterion reduces the error event probability, P_e, for the soft-decision Viterbi detector. In additive white Gaussian noise (AWGN), at moderate signal to noise ratios, this probability is well-approximated by the analytic expression

$$P_e = NQ\left(\frac{d_{\min}}{2\sigma}\right) \tag{1}$$

where σ^2 is the noise variance, $d_{\min}$ is the minimum distance between any pair of allowed sequences and N is the average number of trellis sequences at this distance "seen" by each code sequence at each point in time. The function $Q(x)$ is the well-known complementary error function. The relative increase in the minimum distance defines the coding gain, which, according to (1), is a measure of the extra immunity to additive white Gaussian noise (AWGN).

Unlike other trellis codes for partial response channels, however, MSN codes exploit, rather than nullify, the memory of the partial response channel. The key idea underlying the design of MSN codes is to match the nulls of the code power spectrum with those of the channel transfer function. Thus, for example, an MSN code for a Class-4 partial response channel would use sequences with two nulls-- one at dc and the other at the Nyquist frequency-- in the code power spectrum.

A code is said to have an order-K spectral null at a particular frequency if the value of the code power spectrum and its derivatives through order $2K-1$ are zero at that frequency. Similarly, a partial-response channel has an order-L null at a specified frequency if the squared-magnitude of the channel transfer function and its derivatives through order $2L-1$ are zero at that frequency. (In the case of a channel spectral null at dc or the Nyquist frequency, this condition on the channel transfer function translates into the condition that the channel system polynomial $h(D)$ is divisible by $(1-D)^L$ or $(1+D)^L$, respectively.) The coding gain in such an MSN-coded system is dependent upon the order of the spectral nulls in the code power spectrum and the order of the channel nulls. For example, if one applies a code with order-K spectral nulls at dc and the Nyquist frequency to the PR4 channel, the minimum distance at the output of the PR4 channel satisfies $d_{\min} \geq 2(K+1)$. This coding gain property of MSN codes extends to a broad class of partial response channels. Roughly speaking, the minimum distance at the channel output is proportional to the **sum** of the spectral null order of the code and the spectral null order of the partial-response channel. For the dicode, PR4, and EPR4 channels with system polynomials of the form $h(D) = (1-D)(1+D)^L$, for $0 \leq L \leq 2$, if the spectral null orders of the code and channel are equal, the minimum distance is doubled, and the system can tolerate approximately 3 dB more noise power in AWGN.

Several practical advantages arise from the use of the matched-spectral-null approach to create coding gain. The most important is the availability of a reduced-state Viterbi detector that achieves near maximum-likelihood performance with dramatically less complexity than would be required in a maximum-likelihood detector. The detector effectively tracks only the spectral content (or running digital sums) at the spectral null frequencies, rather than the more complicated constraints of the specific code sequences embodied in the encoder finite-state-machine. Since the design distance is guaranteed by the matched spectral null property, this "suboptimal" detector suffices to achieve the improved error-rate performance predicted by the MSN coding gain analysis.

From the signal modulation standpoint, one evident advantage is the spectral null property. In particular, a spectral null at dc has been found to be a desirable code feature for several reasons, as attested to by the past and present use of dc-free and dc-constrained codes in many digital recording

Manuscript received February 17, 1992.

applications [12]. In a PR4 system, a code null at the Nyquist frequency also provides potential benefits by mitigating the effects of distortion and aliasing near the channel band-edge.

Another benefit that derives from the spectral null properties of MSN codes is the implicit restriction on runlengths of consecutive zero samples in the noiseless channel output sequences, as well as in the interleaved subsequences at even/odd time instants. These constraints, sometimes referred to as $(0,G/I)$ constraints [13], are critical to reliable timing recovery and gain control, and they play a role in limiting the required path memory length in the Viterbi detector. (For the latter, the code must satisfy the broader constraint that it generates no quasi-catastrophic trellis sequences, as discussed in more detail in Section III.)

Finally, another feature of the MSN code approach is the flexibility it affords in the design of the encoder/decoder functions. The reduced-complexity detector can be applied to any code that satisfies the spectral null constraints reflected in the detector trellis structure. The code designer can therefore optimize the encoder/decoder functions, taking into account such issues as encoder complexity, decoder error propagation, and forbidden sequences reserved for synchronization or other purposes.

As mentioned in the Introduction, practical TCM schemes providing 3 dB to 4 dB of additional immunity to noise were critical in driving data rates of voiceband data modems toward their theoretical limits. Bandwidth-efficient (i.e., high rate) coded-modulation schemes, such as the rate 8/10 MSN trellis-coded PR4 technique presented in this paper, can play a similar role in optimizing recording channel performance. They provide an efficient way to generate structured waveforms in signal space with improved distance properties to enhance distinguishability in the presence of channel noise, as well as a practical demodulation technique for achieving the performance of a maximum-likelihood detector. The added tolerance to noise in signal space reduces the probability of error at the detector output and permits correct detection in the presence of noise waveforms that would overwhelm the uncoded channel. The errors that do occur tend to be bursty, reflecting the nature of error events in trellis-based detectors. Algebraic error-correction methods (such as Reed-Solomon codes) [14] are well suited to correction of such infrequent burst errors and can be incorporated into an error recovery procedure to take full advantage of the benefits of the coded-modulation channel, providing an overall system configuration that will better approach the performance, density, and data rate limits established by information theory.

III. RATE 8/10 MSN CODE DESIGN

This section describes the details of a specific rate 8/10 MSN code designed for a dicode $(h(D) = 1 - D)$ channel. By means of bit-wise interleaving of code sequences, the same code may be applied to interleaved-dicode channels, such as PR4 $(h(D) = 1 - D^2)$.

A. Overview of Code Properties

The rate 8/10 MSN code is "dc-free"; that is, its power spectrum has a null at $f = 0$, the frequency at which the dicode partial-response channel has a zero in its transfer function. Specifically, the code sequences are among those sequences having a spectral null at $f = 0$ and with digital sum variation (DSV) bounded by 6. The DSV of a sequence $a = a_1, \ldots, a_n$ is the maximum variation in the running digital sum values

$$\sum_{i=1}^{N} (-1)^{a_i}, \tag{2}$$

for $N = 1, \ldots, n$.

In addition to the DSV constraint, the state diagram also incorporates a constraint which limits to 5 the maximum run length of zero samples at the output of the dicode channel. The code is also designed to eliminate all quasi-catastrophic sequences; that is, each valid output sequence corresponds to a unique path in the trellis. These two constraints serve essentially the same function as the global constraint G and the interleaved constraint I in the $(0,G/I)$ codes for the baseline PRML system, namely, ensuring adequate non-zero samples for timing/gain control and limiting the required path memory in the Viterbi detector. In particular, when interleaved for use on the PR4 channel, the code satisfies the constraints $(0,G/I) = (0,10/5)$.

Finally, the code is designed to be invariant under 180 degree phase-shift, without requiring a precoder circuit. In other words, channel output sequences y and $-y$ both decode to the same data sequence x. This property is useful in magnetic recording systems where the state of the write driver is not known in advance, and where head leads may be interchanged.

The code is implemented as a sliding block code. The encoder requires 4 states, and the decoder is a sliding block decoder requiring one codeword look-ahead. Therefore, the maximum length of an error burst in the decoded data sequence resulting from a single error in the detected code sequence is limited to 2 bytes.

The Class-4 channel, which underlies the magnetic PRML system is an interleaved dicode $(1 - D)$ channel, so the trellis code for dicode is easily adapted to Class-4 by interleaving. When interleaved, the code produces spectral nulls at $f = 0$ and $f = 1/2T$, where T is the code symbol interval. These nulls correspond to the zeros in the Class-4 frequency response. Fig. 1 shows the average power spectrum of typical channel input waveforms generated by the interleaved code, as well as the average power spectrum of the waveforms generated by the ensemble of sequences with DSV no more than 6 on each interleave [15].

B. Encoder Description

Table I shows the structure of the finite-state-machine encoder. States are designated by decimal numbers (0-3), and data words are denoted by decimal numbers (0-255). Codewords are grouped into lists with similar properties, as will be described below, and designated by capital letters, with the size of the set indicated in parentheses. For example, $A(100)$ denotes a list of 100 codewords designated by A. The notation $\overline{A}$ denotes the list of codewords which are obtained from the words in set A by bit-wise complementing. The notation ϕA denotes the list of codewords obtained from the list A by reversing the order of appearance of codewords in the list.

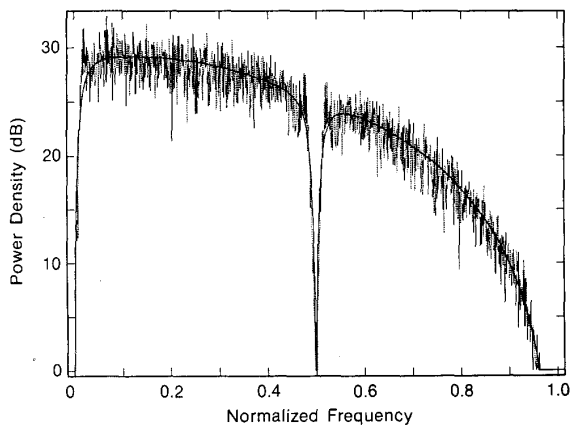

Fig. 1. Power spectrum of MSN-coded input signals and of maximum entropy signals with DSV ≤ 6.

TABLE I
FINITE-STATE-MACHINE ENCODER

Current state	Data	Next state	Codeword
0	0-99	3	$A(100)$
0	100-142	1	$B(43)$
0	143-255	2	$C(113)$
1	0-76	2	$D(77)$
1	77-119	0	$E(43)$
1	120-127	1	$F(8)$
1	128-135	2	$\phi F(8)$
1	136-255	0	$\overline{G}(120)$
2	0-76	1	$\overline{D}(77)$
2	77-119	3	$\overline{E}(43)$
2	120-127	2	$\overline{F}(8)$
2	128-135	1	$\phi\overline{F}(8)$
2	136-255	3	$G(120)$
3	0-99	0	$\overline{A}(100)$
3	100-142	2	$\overline{B}(43)$
3	143-255	1	$\overline{C}(113)$

Fig. 2 shows a diagram, with seven states, that generates all binary sequences with digital sum variation (DSV) bounded by 6, as defined earlier, including, in particular, all codewords in the code.

The codeword lists in the encoder are best described in terms of the related diagram in Fig. 3. In this diagram, the edges in the graph correspond to pairs of consecutive edges in Fig. 2 where the first edge emanates from state 2, 4, or 6. Each edge in Fig. 3 is labelled with the pair of symbols generated by the corresponding pair of edges in Fig. 2. (The Viterbi detector trellis structure will be derived from the diagram shown in Fig. 3 as well.)

List A: Consists of the 100 sequences generated by paths starting from state 2 and ending in state 6.

List B: Consists of the 43 sequences generated by paths from state 2 to state 4 which also pass through state 6.

List C: Consists of 113 of the 121 possible sequences generated by paths from state 2 to state 4 which do not pass through state 6.

List D: Consists of 77 of the 90 sequences generated by paths which start and end at state 4 which also pass through state 2 but do not pass through state 6. The 77 words begin with either 0 (65 total) or 1000 (12 of 13 total).

List E: Consists of the 43 sequences generated by paths from state 4 to state 2 which also pass through state 6.

List F: Consists of 8 of the 20 sequences which start and end at state 4 which also pass through both state 2 and state 6. The 8 sequences are precisely those which begin with 00.

List G: Consists of 120 of the 121 sequences generated by paths from state 2 to state 4 which do not pass through state 6, including the words in list C.

Table II gives a full description of the codeword lists, with the 10 bit words expressed in decimal. Codewords in the list are given in increasing numerical order, from left-to-right, row-by-row from top-to-bottom. List C (not shown) was chosen to be the last 113 words in list G.

C. Decoder Description

The rate 8/10 code is decoded with a sliding block decoder which requires one codeword look-ahead to decode the current codeword. The maximum error length due to a single bit error at the decoder input is therefore no more than 2 data bytes. The sliding block decoder algorithm may be described as follows. For the current 10-bit codeword $y = y_1, \ldots, y_{10}$, the decoder determines an 8-bit word $w = w_1, \ldots w_8$ according to Table III.

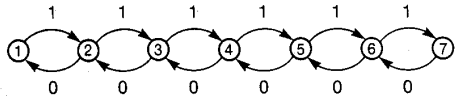

Fig. 2. State diagram for sequences with DSV ≤ 6.

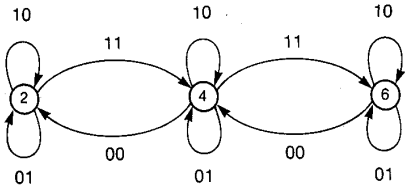

Fig. 3. Component of second power of diagram in Fig. 2.

TABLE II
CODEWORD LISTS

List A

351 367 375 379 381 382 415 431 439 443 445 446 463 471 475
477 478 487 491 493 494 499 501 502 505 506 607 623 631 635
637 638 671 687 695 699 701 702 719 727 731 733 734 743 747
749 750 755 757 758 761 762 799 815 823 827 829 830 847 855
859 861 862 871 875 877 878 883 885 886 889 890 911 919 923
925 926 935 939 941 942 947 949 950 953 954 967 971 973 974
979 981 982 985 986 995 997 998 1001 1002

List B

380 444 476 492 497 498 500 504 636 700 732 748 753 754 756
760 828 860 876 881 882 884 888 924 940 945 946 948 952 963
965 966 969 970 972 977 978 980 984 993 994 996 1000

List D

454 457 458 460 465 466 468 472 481 482 484 488 604 620 625
626 628 632 668 684 689 690 692 696 709 710 713 714 716 721
722 724 728 737 738 740 744 789 790 793 794 796 805 806 809
810 812 817 818 820 824 837 838 841 842 844 849 850 852 856
865 866 868 872 901 902 905 906 908 913 914 916 920 929 930
932 936

List E

240 368 432 449 450 452 456 464 480 624 688 705 706 708 712
720 736 773 774 777 778 780 785 786 788 792 801 802 804 808
816 833 834 836 840 848 864 897 898 900 904 912 928

List F

124 188 220 236 241 242 244 248

List G

347 349 350 359 363 365 366 371 373 374 377 378 407 411 413
414 423 427 429 430 435 437 438 441 442 455 459 461 462 467
469 470 473 474 483 485 486 489 490 599 603 605 606 615 619
621 622 627 629 630 633 634 663 667 669 670 679 683 685 686
691 693 694 697 698 711 715 717 718 723 725 726 729 730 739
741 742 745 746 791 795 797 798 807 811 813 814 819 821 822
825 826 839 843 845 846 851 853 854 857 858 867 869 870 873
874 903 907 909 910 915 917 918 921 922 931 933 934 937 938

TABLE III
DECODER INITIAL LOOK-UP TABLE

y	w
A	0 - 99
$\overline{A}$	0 - 99
B	100 - 142
$\overline{B}$	100 - 142
G (note $C \subset G$)	136-255
$\overline{G}$	136-255
D	0 - 76
$\overline{D}$	0 - 76
E	77 - 119
$\overline{E}$	77 - 119
ϕF	128 - 135
$\phi \overline{F}$	128 - 135
Other	0 (or erasure indicator)

For all but 16 of the codewords, namely the 16 words in lists F and $\overline{F}$, the decoder actually requires no look-ahead, and the decoder output is w. For the remaining 16 words, the decoder output is either w or the bit-wise complement of w, as determined by a simple binary-valued function U of the bits in the current and look-ahead codewords.

To define the binary-valued function U, it is convenient to introduce certain variables related to the current and look-ahead codewords. Let $z = z_1, \ldots, z_{10}$ denote the look-ahead codeword. Let $V(y)$ denote the Hamming weight (the number of 1's) in the current codeword, and $V(z)$ denote the same for the look-ahead codeword. The decoder uses a look-up table to determine the 8-bit pattern w. The decoded data word $x = x_1, \ldots, x_8$ is given by $x_i = w_i + U(y,z)$, where U is defined by:

$$U(y,z) = \begin{cases} 1 & \text{if } V(y) = 5 \text{ and } y_1y_2 = 11 \text{ and} \\ & ((V(z) = 6) \text{ or } (V(z) = 5 \text{ and} \\ & (z_1z_2z_3z_4 = 0111 \text{ or } 1\overline{000}))) \\ \text{or} & \text{if } V(y) = 5 \text{ and } y_1y_2 = 00 \text{ and} \\ & ((V(z) = 4) \text{ or } (V(z) = 5 \text{ and} \\ & (z_1z_2z_3z_4 = 1000 \text{ or } 0\overline{111}))) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The notation $\overline{000}$ refers to the set of binary 3-tuples excluding 000, and similarly for $\overline{111}$.

D. Viterbi Detection

A maximum-likelihood detector for the rate 8/10 MSN-coded dicode channel would use a trellis structure derived from the finite-state encoder. However, this trellis would be far too complex for practical implementation. A significant reduction in complexity and near-maximum-likelihood performance was achieved by using a trellis structure derived from the spectral null diagram in Fig. 3. Fig. 4 shows the reduced complexity Viterbi detector trellis structure for the rate 8/10 MSN code.

The edges are labelled by u_1u_2/v_1v_2 where u_1u_2 are code symbols and v_1v_2 are channel output symbols. The free Euclidean distance of the trellis, d^2_{free}, is the minimum squared Euclidean distance between trellis-coded channel output sequences which start from the same state and remerge after a finite number of steps. It can be checked that $d^2_{free} = 4$, indicating a potential 3 dB gain over the uncoded dicode channel. Fig. 4 shows an example of a minimum distance error event.

The maximum possible length of a minimum distance error event is no more than 42 bits. This is important for limiting the required path memory in the Viterbi detector. Standard interleaving techniques can be utilized to apply this detector structure to the Class-4 channel.

The Viterbi algorithm for the trellis in the figure is derived according to conventional methods. The survivor metric update rules are given in Table IV. Survivor path extensions are determined on the basis of the metric update computations in the usual manner by means of an Add-Compare- Select (ACS) processor. Issues related to the implementation of this algorithm are addressed in detail in Section IV.

Fig. 5 shows the results of a bit-by-bit computer simulation of the rate 8/10 MSN-coded dicode channel. The noise was assumed to be additive, Gaussian with zero mean and standard deviation σ, independent from sample to sample. The probability of error event at the detector is plotted as a function of signal-noise-ratio. Also shown are two analytically calculated curves: the performance of the uncoded dicode with maximum-likelihood detection (given by $P_{event} = 2Q(\sqrt{2}/2\sigma)$), and the performance corresponding to an exact 3 dB coding gain. The rate 8/10 MSN code achieves approximately 2.8 dB coding gain in this range of SNR. The deviation from 3 dB can be attributed to the "error coefficient", which reflects the average number of trellis sequences at minimum distance from a typical code sequence. In this SNR range, all observed error events were minimum distance.

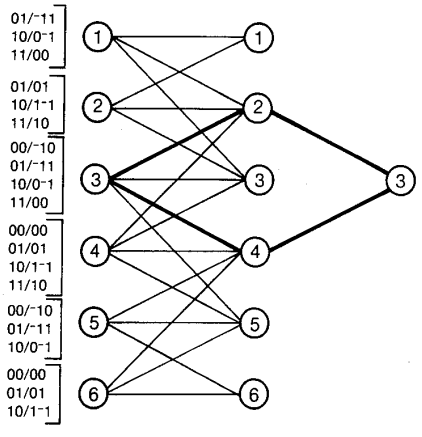

Fig. 4. Trellis structure for rate 8/10 MSN Viterbi detector.

TABLE IV
VITERBI ALGORITHM METRIC UPDATE EQUATIONS

$$M_{n+1}(1) = \min\{M_n(1) + 2 + 2z_1 - 2z_2, M_n(2) + 1 - 2z_2\}$$

$$M_{n+1}(2) = \min\{M_n(1) + 1 + 2z_2, M_n(2) + 2 - 2z_1 + 2z_2, M_n(3) + 1 + 2z_1, M_n(4)\}$$

$$M_{n+1}(3) = \min\{M_n(1), M_n(2) + 1 - 2z_1, M_n(3) + 2 + 2z_1 - 2z_2, M_n(4) + 1 - 2z_2\}$$

$$M_{n+1}(4) = \min\{M_n(3) + 1 + 2z_2, M_n(4) + 2 - 2z_1 + 2z_2, M_n(5) + 1 + 2z_1, M_n(6)\}$$

$$M_{n+1}(5) = \min\{M_n(3), M_n(4) + 1 - 2z_1, M_n(5) + 2 + 2z_1 - 2z_2, M_n(6) + 1 - 2z_2\}$$

$$M_{n+1}(6) = \min\{M_n(5) + 1 + 2z_2, M_n(6) + 2 - 2z_1 + 2z_2\}$$

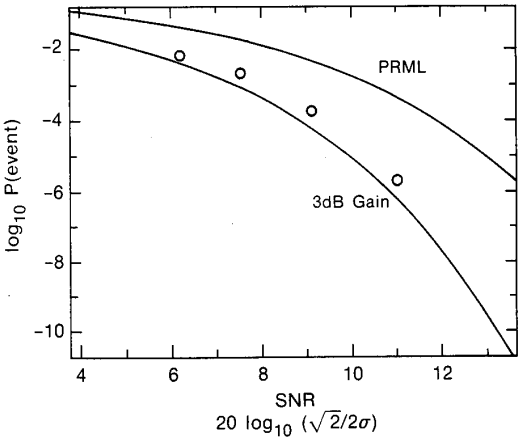

Fig. 5. Computer simulation of rate 8/10 MSN-coded dicode channel.

IV. EXPERIMENTAL PROTOTYPE

In order to experimentally evaluate the performance of the rate 8/10 MSN code described in Section III, an experimental prototype was built using a custom VLSI chip that incorporated the encoder, Viterbi detector, and decoder functions. The details of the chip implementation are given in [16]. Here we will describe the salient features of the prototype as well as some of the novel architectures used in the chip design.

The block diagram of the chip is shown in Fig. 6. The encoder is a 4-state machine that encodes an 8-bit data input into a 10-bit codeword output. Its design was based on the table look-up method. The decoder is a sliding block decoder that maps suitably synchronized two adjacent 10-bit sequences from the Viterbi detector to an 8-bit decoded data. It was also implemented using table look-up along with combinatorial logic to perform the look-ahead function.

The Viterbi detector is based on the six state trellis diagram given in Fig. 4 and the associated recursive equations in Table IV. It was implemented using a novel pipelined approach that saved about 50% in the chip area relative to the conventional approach.

Most high-speed hardware realizations of the Viterbi detector rely on a state-parallel approach wherein one add-compare-select unit is devoted to each state (recursion) in the trellis diagram. A conventional Viterbi detector for the six state trellis in Fig. 4 would therefore require four 4-input ACS's (for states 2, 3, 4 and 5) and two 2-input ACS's (for states 1 and 6). Our implementation required two 4-input ACS units. The ACS units were configured to operate in a carefully organized pipeline that allowed the servicing of all six states, but with minimum impact on the speed. The scheduling and partitioning of the states involved in the pipeline are described in [16]. Theoretical extensions and further applications of this pipelined architecture may be found in [17].

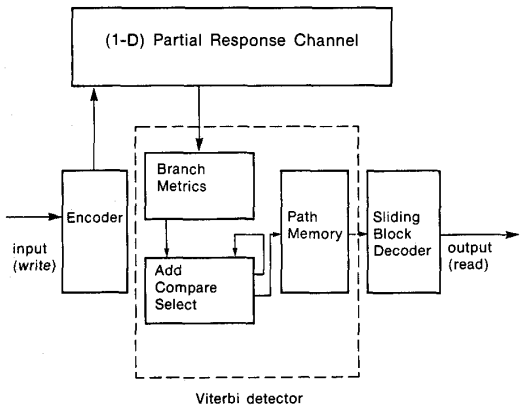

Fig. 6. VLSI chip block diagram.

The overall area-time tradeoff for the Viterbi detector can be summarized as follows: The speed is 33% slower than a 6-state, non-pipelined implementation, mainly due to the use of two additional dummy states. On the other hand, the area is substantially less because of the reduction in: (1) the number of ACS's; (2) the number of feedback connections between ACS's; and (3) the number of interconnections between the branch metric calculators and the ACS's, stemming from the fact that each branch metric is used in one ACS only. Although these area improvements are slightly offset by the incorporation of pipeline latches in the ACS units, the net area reduction is measured to be roughly 50%, thereby providing a favorable overall area-speed tradeoff.

Another important consideration in the design of the Viterbi detector is path metric normalization. The normalization is required to prevent errors due to overflow or underflow during the updating and storage of the path metrics. In the PRML method, this normalization is implicitly performed in the commonly-used difference metric algorithm. In our implementation, modulo normalization [18], which relies on the properties of two's-complement arithmetic, was used. This technique builds the normalization directly into the ACS operation and thereby provides the most local and uniform design from the VLSI standpoint. It is especially well-suited to the pipelined ACS architecture mentioned above.

The remaining function in the Viterbi detector-- namely, the path memory-- was implemented using the conventional register-exchange scheme requiring the survivor sequences to be stored in 6 shift registers, each of length 32 stages (64 bits). Each shift register is updated during every symbol interval.

Two of the VLSI modules shown in Fig. 6 were configured according to Fig. 7 to build the complete experimental prototype, capable of operating up to 3 MB/sec (30 MHz). The incoming data is serial-to-parallel converted to produce 8-bit words, which are alternately applied to the two encoders. The encoded output is serialized and interleaved. The resulting sequence has a finite running digital sum as well as a finite alternate running digital sum, thereby satisfying the necessary and sufficient conditions for producing nulls in the code power spectrum at dc and the Nyquist frequency. These nulls match the nulls of the Class-4 partial response spectrum.

On the readback side, suitably equalized samples are de-interleaved and applied to the soft-decision Viterbi detectors. The detected output is decoded and suitably reframed to produce the readback data sequence. The timing and gain control functions were derived from an experimental PRML receiver.

V. EXPERIMENTAL RESULTS

The performance of the rate 8/10 MSN code was experimentally evaluated and compared with the PRML method. The comparison was based on the experimental configuration shown in Fig. 8. The analog signal generator was used to synthesize Class-4 partial response waveforms for the rate 8/10 MSN code or the rate 8/9 (0,4/4) code based on a given random data input. Noise was added to the waveform, which was then applied to the PRML receiver. For the rate 8/9 coded system, the decoded output from the PRML receiver was used to determine the error rate as a function of signal-to-noise ratio (SNR). For evaluating the MSN code performance, the PRML receiver was used only for the purpose of performing A/D conversion, and for timing and gain recovery. The sample sequence and the recovered clock from the receiver were applied to the trellis codec chips. The

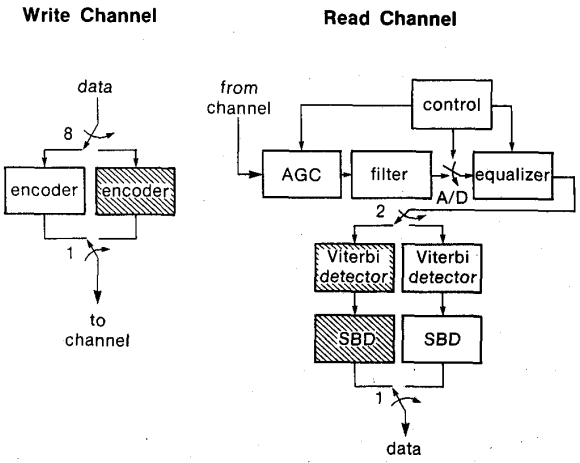

Fig. 7. Configuration of simulated MSN-coded PR4 system.

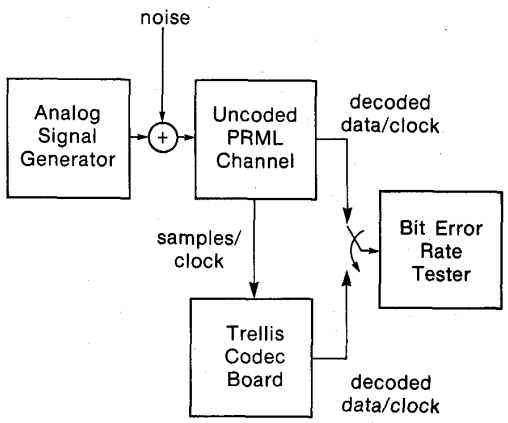

Fig. 8. Experimental set-up.

decoded data was used to measure the error rate. All performance comparisons were made at a channel data rate of 30 Mbits/sec.

Fig. 9 shows measured performance of the baseline PRML and the rate 8/10 MSN-coded systems in the presence of additive noise. As shown, the trellis-coded system tolerates approximately 2.8 dB more noise than the uncoded system at an error rate of 10^{-7}. The measured improvement is in good agreement with the analytic and simulated results given in Section III.

In magnetic recording applications, the effect of correlated interference is of more significance than additive white Gaussian noise. A second experiment was conducted to evaluate the effect of correlated interference (resembling adjacent track data) on the MSN-coded and the PRML systems. White Gaussian noise was added to the data waveforms until the error rate for both systems was 10^{-7}. Based on the first experiment, this amounted to setting the SNR for the MSN-coded system to be 2.8 dB lower than that for PRML. Suitably encoded waveforms, resembling adjacent track interference (ATI), were then added asynchronously to the noisy waveforms to simulate the offtrack condition. The amplitude of the interfering waveform was adjusted to obtain the bathtub-like plot shown in Fig. 10. As shown, the MSN-coded system with 2.8 dB lower SNR provides the same bathtub performance as PRML, thereby demonstrating comparable immunity to correlated interference as PRML. Both of these experimental results suggest that the MSN coded system provides added immunity to noise or interference. This immunity may be used to improve performance or increase the recording density, or both, in magnetic recording systems.

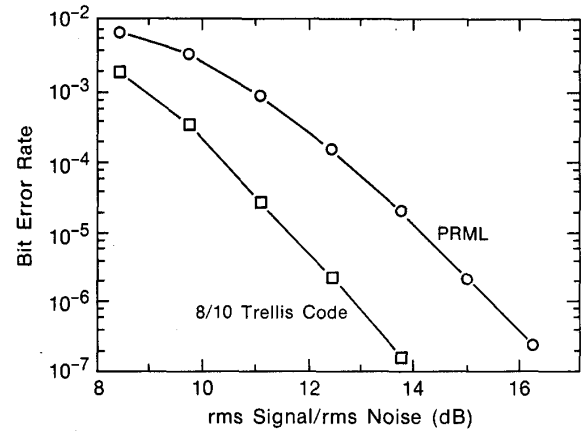

Fig. 9. Measured performance in AWGN.

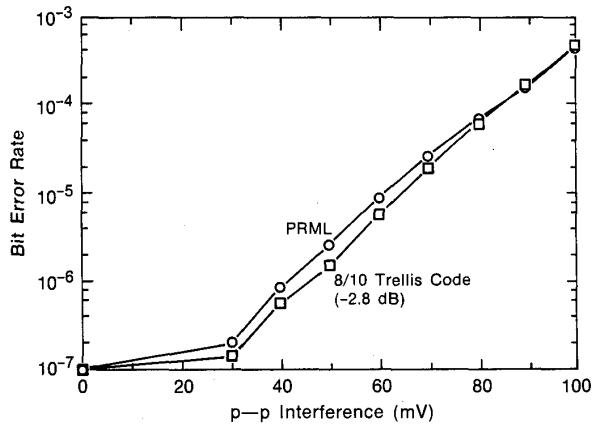

Fig. 10. Measured performance in AWGN and synthesized ATI.

VI. CONCLUDING REMARKS

This paper has described the application and performance evaluation of a rate 8/10, matched-spectral-null (MSN) trellis code in a Class-4 partial response system. Details of the encoder and decoder functions were presented, as well as the reduced-complexity trellis structure underlying the Viterbi detector for the coded system. Implementation issues involved in the design of a VLSI chip embodying the encoder, decoder, and Viterbi detector for the trellis-coded system were discussed. Experimental results confirm theoretical predictions that the rate 8/10 code provides approximately 3 dB advantage over the baseline Class-4 partial-response, maximum-likelihood (PRML) system in the presence of additive, white, Gaussian noise as well as additive adjacent track interference.

REFERENCES

[1] H. Kobayashi and D.T. Tang, "Application of partial-response channel coding to magnetic recording systems," *IBM J. Res. Dev.*, vol. 14, pp. 368-375, July 1970.

[2] H. Kobayashi, "Application of probabilistic decoding to digital magnetic recording systems," *IBM J. Res. Dev.*, vol. 15, no. 1, pp. 64-74, January 1971.

[3] G.D. Forney, Jr., "Maximum likelihood sequence detection in the presence of intersymbol interference," *IEEE Trans. Info. Th.*, vol. IT-18, no. 3, pp. 363-378, May 1972.

[4] H. Thapar and T. Howell, "On the performance of partial response maximum likelihood and peak detection methods in digital magnetic recording," *Digests of the 1991 IEEE Magn. Rec. Conf.*, Hidden Valley, Pennsylvania, June 12-15, 1991, paper D1.

[5] K. Fisher, J. Cioffi, C. Melas, "An adaptive DFE for storage channels suffering from nonlinear ISI," *Proc. 1989 IEEE Int. Conf. Comm.*, Boston, Massachusetts, June 1989.

[6] H. Thapar and A. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magn.*, vol. MAG-23, no. 5, pp. 3666-3668, September 1987.

[7] R. Karabed and P. Siegel, "Matched-spectral-null codes for partial response channels," *IEEE Trans. Info. Th.*, vol. 37, no. 3, pt. II, pp. 818-855, May 1991.

[8] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, 623-656, October 1948.

[9] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets," *IEEE Commun. Mag.*, vol. 25, no. 2, pp. 5-21, February 1987.

[10] J. Wolf and G. Ungerboeck, "Trellis coding for partial-response channels," *IEEE Trans. Commun.*, vol. COM-34, no. 8, pp. 765-773, August 1986.

[11] A.R. Calderbank, C. Heegard, and T.A. Lee, "Binary convolutional codes with application to magnetic recording," *IEEE Trans. Info. Th.*, vol. IT-32, no. 6, pp. 797-815, November 1986.

[12] K. A. Schouhamer Immink, *Coding Techniques for Digital Recorders*, Englewood Cliffs, New Jersey: Prentice-Hall, 1991.

[13] B. Marcus, P. Siegel, and J. Wolf, "Finite-state modulation codes for data storage," *IEEE J. Select. Areas Commun*, vol. 10, no. 1, pp. 5-37, January 1992.

[14] E. Berlekamp, "The technology of error-correcting codes," *Proc. IEEE*, vol. 68, no. 5, pp. 564-593, May 1980.

[15] K.J. Kerpez, "The power spectral density of maximum entropy charge constrained sequences," *IEEE Trans. Info. Th.*, vol. 35, no. 3, pp. 692-695, May 1989.

[16] C. 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.

[17] C. Shung, H-D. Lin, R. Cypher, P. Siegel, and H. Thapar, "Area-efficient architectures for the Viterbi algorithm, Part I: Theory and Part II: Applications," *IEEE Trans. Commun.*, in press.

[18] C. Shung, G. Ungerboeck, P. Siegel, H. Thapar, "VLSI architectures for metric normalization in the Viterbi algorithm," *Proc. 1990 Int. Conf.Commun.*, pp. 1723-1728, Atlanta, Georgia, April 1990.