tial to increase the sample rates possible with a given circuit family, and to allow a particular rate to be achieved with a lower cost circuit technology, because it has the ability to sample data at frequencies that significantly exceed the maximum switching speed of the circuit elements.

20 January 1994
Electronics Letters Online No: 19940339

S. M. Clements, W. Liu, J. Kang and R. K. Cavin III (*Department of Electrical and Computer Engineering, North Carolina State Univesrity Raleigh, NC 27695-7911, USA*)

References

1 RAHKONEN, T., and KOSTAMOVAARA, J.: 'The use of stabilized CMOS delay lines for the digitisation of short time intervals', *IEEE J. Solid-State Circuits*, **SC-28**, (8), pp. 887–894, 1993

2 VAN NOIJE, W., GRAY, C.T., LIU, W., HUGHES, T.A., CAVIN, R.K., and FARLOW, W.J.: 'CMOS sampler with 1Gbit/s bandiwdth and 25ps resolution". Proc. IEEE Custom Integrated Circuits Conf., pp. 27.5.1–27.5.4, San Diefo, CA, May 1993

3 KANG, J., LIU, W., and CAVIN III, R.K.: 'A monolithic 625Mb/s data recovery circuit in 1.2μm CMOS'. Technical Report NCSU-VLSI-93-12, North Carolina State Univesrity, 1993

4 CLEMENTS, M., LIU, W., KANG, J., and CAVIN III, R.K.: 'Continuous sampling and storage using matched delays'. Technical Report NCSU-VLSI-93-13, North Carolina State Univesrity, 1993

New filling procedure to reduce delay of burst-error correcting array codes

W. Henkel and H.Y. Chung

Indexing terms: Error correction codes, Codes and coding

A special procedure to fill the array of an array code (product code) is proposed that reduces the delay by about 35%, depending on the size of the array code. The method is based on an array code with diagonal readout, prefilling a region of ~30% of the array and then in parallel to the readout using diagonal, horizontal and vertical writing modes interchangeably. The mode changes are determined by a computer search, thereby minimising the number of prefilled components.

Introduction: Array codes with diagonal readout have been proposed by Blaum, Farrell and van Tilborg in [1, 2]. Let the parameters be $(n_1, n_2; s)$, where n_1 is the number of rows, n_2 the number of columns, and s the step size when moving from one diagonal to another. We shall focus on the case $s = 1$, described by

$$f(i,j) - (j-i)n_1 + i \quad \text{modulo}(n_1 n_2) \qquad i = 0,1,\ldots,n_1-1 \quad j = 0,1,\ldots,n_2-1 \tag{1}$$

In [1, 2] the authors show that, if we choose n_2 to be greater than or equal to $2n_1 - 3$, then these array codes using a simple parity check in rows and columns are capable of correcting all single bursts of length $n_1 - 1$. Regarding the following example with $n_1 = 5$, $n_2 = 2n_1 - 3 = 7$, it is obvious that an error burst of length $n_1 - 1 = 4$ results in a parity violation in $n_1 - 1$ consecutive positions of rows and columns starting from row and column locations that correspond to the first error in the burst, whereas in the case of a burst length of n_1 two errors may cancel each other when computing the parity (when the burst does not begin in the first row). The starting position of the burst can only be detected by the column parities, if a possible error-free range G_{in} within one burst cannot be as long as the remaining error-free range G_{out} outside the burst. Formally, this means

$$G_{out} = n_2 - (n_1 - 1) > (n_1 - 1) - 2 = G_{in} \qquad \Longrightarrow n_2 > 2n_1 - 4 \Longrightarrow n_2 \geq 2n_1 - 3 \tag{2}$$

The idea of the decoding algorithm is to search for the longest error-free range of columns. The burst starts cyclically after this range. The burst location is now given by taking the cyclic column positions starting after the determined error gap [Note 1] and combining it with the corresponding row locations, beginning at the top of the scheme and proceeding downwards.

Example of diagonal readout: The parameters are (5, 7; 1). The numbers in the array in Table 1 below correspond to the order of readout.

Table 1: Array for diagonal readout example

0	5	10	15	20	25	30
31	1	6	11	16	21	26
27	32	2	7	12	17	22
23	28	33	3	8	13	18
19	24	29	34	4	9	14

As for block interleavers, the main drawback of such a scheme is the delay of nearly twice the array size. The following Section describes a procedure that reduces the delay caused by the transmitter-side array significantly. Nothing can, of course, be done to reduce the delay at the receiver matrix.

Writing procedure: First note that, if we start writing and reading in a diagonal manner, the first parity symbol that would be transmitted is $(n_1 - 1, n_1 - 1)$, where counting starts from zero. Therefore, we should prefill the corresponding column and compute the parity symbol. Even the next column parity positions might be needed for the following diagonal readouts. For this reason, we prefill a certain number of columns starting from column $n_1 - 1$. How many are needed to be prefilled has to be determined by a computer search, but is always about 30% of the whole array.

After the prefilling, writing and reading starts from position (0, 0) in a diagonal fashion. Then it is evaluated, if, when proceeding with the diagonal readout, a vertical or horizontal parity symbol would be needed next. The writing then switches to vertical or horizontal mode, filling the corresponding columns or rows, respectively. The mode changes are determined by the search program minimising the number of prefilled components. One parity position, the parity of the parities $(n_1 - 1, n_2 - 1)$, must be left empty or filled with some unprotected information. The real parity symbol can only be computed after the whole matrix is filled. Unfortunately, this parity symbol is needed for the above-mentioned decoding algorithm. Hence, it has to be transmitted afterwards. This should be done in triplicate, featuring a repetition code of length 3, where each repeated symbol has to be transmitted at the 1st, n_1th and $(2n_1 - 1)$th position after the whole matrix is read out (inserted into the data of the next array). The delay between the repeated symbols takes into account the maximum burst length of $(n_1 - 1)$. Thus, the repetition code can correct an error caused by such a burst.

Table 2: Array for writing procedure example

0	5	9	17	19	*	*	12	@
24	1	6	10	18	*	*	13	@
25	29	2	7	11	*	*	14	@
26	27	28	3	8	*	*	15	@
20	21	22	23	4	*	*	16	@
@	@	@	@	@	@	@	@	—

where * = prefilled positions, @ = parity symbols, — = parity-of-parities (left empty)

Example for writing procedure: The parameters are (6, 9; 1). The numbers in the array in Table 2 correspond to the order of writing:

Note 1: Gap = error-free range between error bursts

The changes of mode take place at the positions given in Table 3.

Table 3: Positions of mode changes

Mode	(*i*, *j*)	Symbol number
diagonal	(0, 0)	0
vertical	(0, 7)	12
diagonal	(0, 3)	17
horizontal	(4, 0)	20
vertical	(1, 0)	24
horizontal	(3, 1)	27
vertical	(2, 1)	29

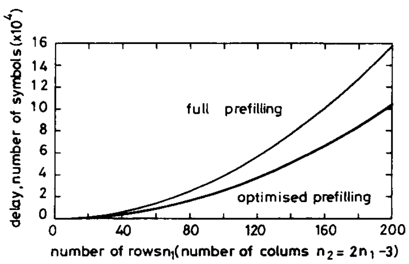

Fig. 1 *Comparison of delays in the case of optimised prefilling and the usual full prefilling*

Fig. 1 shows the difference in delay between the proposed optimised prefilling scheme and the normal full prefilling. Fig. 2 gives the percentage of redundancy of the array code.

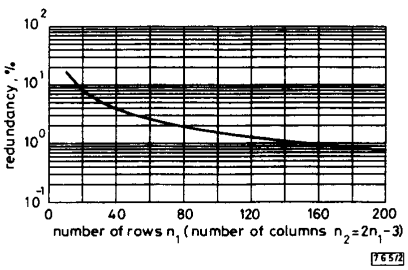

Fig. 2 *Percentage of redundancy of an array code (including repetition code for parity-of-parities symbol)*

Curves are drawn dependent on the number of rows n_1 with the smallest number of columns $n_2 = 2n_1 - 3$. The effect of the repetition code for the parity-of-parities symbol has been incorporated.

Finally, we consider an example for a 6.864Mbit/s-ADSL (asymmetrical digital subscriber line) [3] which uses an additional rate-11/12 trellis-coded modulation. The scheme should be capable of correcting an error burst of length up to 200µs (impulsive noise on subscriber lines) and the introduced delay should be less than 40ms. An array code based on 16bit components with $n_1 = 85$ leads to a delay of 39.1ms with a burst-error correcting capability of 84 symbols = 176µs. The percentage of redundancy in this case is only 1.79%. If we choose the symbol size to be 24 bits and $n_1 = 70$, we obtain a delay of 39.5ms and a maximum burst length of 219.5µs (redundancy = 2.18%).

One advantage of using an array code for ADSL instead of the more popular burst-error-correcting Reed-Solomon codes is the much lower decoding complexity.

Conclusions: A new prefilling and array writing procedure has been proposed that interchanges different writing modes (diagonal, vertical, horizontal). This reduces the delay significantly, so that it can be used as an alternative to convolutional interleaving together with Reed-Solomon codes. An example for a 6.864Mbit/s-ADSL has been given.

A 'C' program for determining the writing procedure for a certain array code can be obtained from the authors.

17 January 1994
Electronics Letters Online No: 19940325

W. Henkel (*Research Center of Deutsche Bundespost Telekom, PO Box 10 00 03, D-64276 Darmstadt, Germany on leave at AT&T Bell Laboratories, 200 Laurel Ave, Middletown, NJ 07748, USA*)

H. Y. Chung (*AT&T Bell Laboratories, 200 Laurel Ave, Middletown, NJ 07748, USA*)

References

1 BLAUM, M., FARRELL, P.G., and VAN TILBORG, H.C.A.: 'A class of burst error-correcting array codes', *IEEE Trans.*, 1986, **IT-32**, (6), pp. 836–839

2 VAN TILBORG, H.C.A.: 'An overview of recent results in the theory of burst-correcting codes' (Springer Lecture Notes in Computer Science, 388, 1988), pp. 164–184

3 SISTANIZADEH, K.: '*Ad Hoc* meeting report on ADSL loop reach, channel rates, & PSD'. *T1E1.4 Ad Hoc Meeting Contribution T1E1.4/93–150*, 11 May, 1993

Ring-TCM codes for QAM on fading channels

F.J. López, R.A. Carrasco and P.G. Farrell

Indexing terms: Trellis codes, Amplitude modulation, Fading convolutional codes

The application of new ring-TCM codes for QAM on the land mobile satellite channel is investigated. It is demonstrated that their intrinsic trellis structure makes them especially suitable for application on fading channels. Furthermore, a comparison between ring-TCM codes and the best known TCM codes for fading channels has revealed that they offer a significantly better tradeoff between actual coding gain and decoder complexity.

Introduction: In recent years, the land mobile satellite channel has become one of the most important channels for the transmission of digital information. The fact that this channel is severely limited in both power and bandwidth resources has been primarily overcome with the application of trellis coded modulation (TCM) [1]. However, there is now a growing demand for higher spectrum efficiencies. Therefore, constant envelope modulation formats, such as *q*-ary phase shift keying (*q*-PSK), which have been imposed for a long time due to amplitude distortions introduced by this channel on the transmitted signal, must now be replaced by more power efficient non-constant envelope *q*-ary quadrature amplitude modulation (*q*-QAM) techniques. Fortunately, advances in the present state of the art will enable the use of *q*-QAM modulation schemes on the land mobile satellite channel in the near future (amplitude distortions caused by the nonlinearities of the high power amplifiers (HPAs) can be resolved by using predistortion techniques [2], and channel sounding techniques, which allow the estimation of the magnitude of the fades in the channel, can be used to counteract the fading distortion [3]).

In [4], a new 4-D TCM technique based on rings of integers modulo-4 and suitable for rectangular *q*-QAM signal sets was presented. A range of non-rotationally invariant (NRI) and rotationally invariant (RI) ring-TCM codes were designed and evaluated on the Gaussian channel. In the present study, the application of ring-TCM codes to the land mobile satellite channel is investigated. Following the well known criteria to design optimum TCM codes for fading channels described in [5], a range of new NRI and RI ring-TCM codes for 16-QAM and optimised for fading channels is presented. It is demonstrated that the intrinsic trellis structure of these codes is especially suitable for application on fading channels. Furthermore, their performance on different fading channels has been analysed, and a comparison between these