объединенный
институт
ядерных
исследований
дубна

E10-89-362

N.M.Nikityuk

SOME QUESTIONS OF USING THE ALGEBRAIC CODING THEORY FOR CONSTRUCTION OF SPECIAL-PURPOSE PROCESSORS IN HIGH ENERGY PHYSICS SPECTROMETERS

Submitted to International Conference AAECC 7,
Toulouse Cedex, France, 26.06.89

1989

Problem

Different types of coordinate detectors, which consist of a large number of hodoscopic planes, are widely used in high energy physics. These planes contain many position-sensitive detectors (sources). One plane is composed of several hundred or thousand sources and more. As a consequence, there are tens of thousands of registration channels in conventional spectrometers. A typical scheme of such a spectrometer is given in fig.1. A great volume of fast electronics, special-purpose processors (SP) and modern computers are required for event registration, the accumulation of statistics and the reconst ruction of particle interaction. It should be noted that the cost of modern high energy spectrometers is equal to several million dollars, and the cost of data processing electronics is about 40 %. This is the reason why the problem of optimum coding and information compression, registered in multichannel detectors of charged particles (MDCP), arises so sharply. The processing of physics events is hierarchical in nature. Each selection level is characterized by the dead time τ. At the first level the value of τ equals 30 - 50 ns. During this

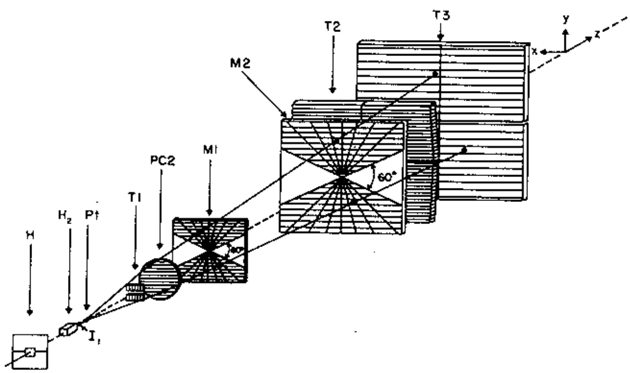

Fig.1. Block-diagram of a typical high energy physics spectrometer. T1 - T3 - scintillation hodoscopes; PC1 - PC6 and M1 - M2 - multiwire proportional chambers; H_2, P_t - targets.

time the spectrometer is put into operation, and the multiplicity of events is determined. At the second level the coordinates of events or other parameters, e.g. the scattering angle of particles, are determined. If the solution is positive, the data are registered on a tape.

The method of syndrome coding for the filtration of events at the first and second levels has been suggested by the author [2]. The use of this method allows one to apply a mathematical algorithm of algebraic coding theory for the creation of economical and fast devices for the registration and processing of useful events. New results are given.

II. Systolic method of signal processing

According to the syndrome method, a Galois field element is set for each source so that the first source corresponds to an a^0 element, the second source to an a^1 element and the n-th source to an a^{n-1} element. In other words, the positions of the sources are numbered by the degrees of the Galois field elements $GF(2^m)$. As data in MDCP are read out in a unitary position code, the next step after amplifying and shaping the signals is their transformation to a cyclic code (Galois field element) [2,3]. For simplicity a MDCP is assumed to have $n = 2^6 - 1 = 63$ sources ($m = 6$). This means that the Galois field elements are generated over an irreducible polynomial $X^6 + X + 1$, and the element $a^1 = 010000$ is the root of this polynomial. Then for multiplicity $t < 4$ a part of the coding matrix (parity check) $H_{63,24}$ takes the form

$$H^T_{63,2} = \begin{array}{r} *0 \\ I \\ 2 \\ *3 \\ 4 \\ *5 \\ \cdots \\ 62 \end{array} \left|\begin{array}{cccc} I & I & I & 1 \\ a^I & a^3 & a^5 & a^7 \\ a^2 & a^6 & a^{I0} & a^{I4} \\ a^3 & a^9 & a^{I5} & a^{2I} \\ a^4 & a^{I2} & a^{20} & a^{28} \\ a^5 & a^{I5} & a^{25} & a^{35} \\ \cdots & \cdots & \cdots & \cdots \\ a^{62} & a^{60} & a^{58} & a^{56} \end{array}\right| = \left|\begin{array}{cccc} I00000 & I00000 & I00000 & I00000 \\ 0I0000 & 000I00 & 00000I & 0II000 \\ 00I000 & II0000 & 0000II & 00I0I0 \\ 000I00 & 000II0 & 000I0I & II0III \\ 0000I0 & I0I000 & 00IIII & 00III0 \\ 00000I & 000I0I & 0I000I & II0I00 \\ \vdots & \vdots & \vdots & \vdots \\ I0000I & I00III & IIIIII & IIIII0 \end{array}\right|$$

The elements of $GF(2^6)$ are presented in Appendix. Let the sources operate simultaneously at $X_I = a^0 X_2 = a^3$ и $X_3 = a^5$. Then we have

$$S_I = a^0 + a^3 + a^5 = a^{23},\ S_3 = a^0 + a^9 + a^{I5} = a^{6I},\ S_5 = a^0 + a^{I5} + a^{25} = a^{35} \text{ и } S_7 = a^0 + a^{2I} + a^{35} = a^{6I}. \quad (I)$$

To obtain a fast speed, the syndrome is calculated with the aid of parallel parity checkers (fig. 2). The analysis of the $H^T_{63,18}$ matrix shows

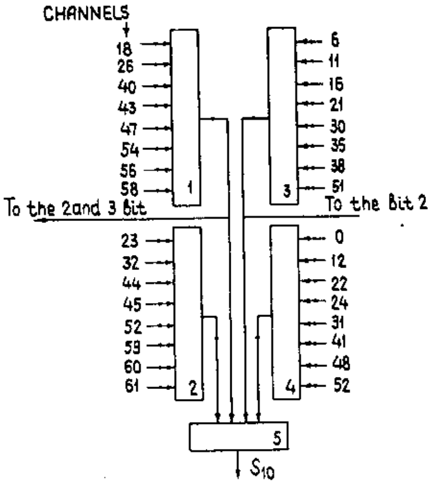

Fig.2. Principal scheme for the one-digit syndrome. 1 – 5 MC10160.

that the number of checkers can be decreased by a third if the inputs are grouped so that coincident units in the columns of the matrix enter into parity checking for a variety of syndrome digits. For example, there is a coincidence in the first column at positions 26, 40, 43, 47, 54, 56 and 58. The transformation of the unitary position code to the Galois field elements is executed rather fast by the parallel method as the delay of MC10160 is 6 ns. The number of syndrome bits, N, is 18 for n = 63 and t = 3. Thus, information from a 63-bit unitary code is compressed to a 18-bit cyclic code. The compression coefficient is 63/18.

III. Majority coincidence circuits with algebraic structure

The important property of the syndrome of the BCH-code is to carry information on the multipilicity and coordinates of particle interactions . The algoritm of a majority coincidence circuit is based on the property of the L_t matrix. The txt matrix [5]

$$L_t = \begin{vmatrix} S_1 & 1 & 0 & 0 & 0 & 0 \ldots\ldots 0 \\ S_3 & S_2 & S_1 & 1 & 0 & 0 \ldots\ldots 0 \\ S_5 & S_4 & S_3 & S_2 & S_1 & 0 \ldots\ldots 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ S_{2t-1} & S_{2t-2} & S_{2t-3} & S_{2t-4} & S_{2t-5} & S_{2t-6} \quad S_t \end{vmatrix}$$

is singular if the weight of t is j – 1 or less and nonsingular if the weight of t is j or j + 1. The expressions for a matrix determinant for t = 1 - 4 calculated by a computer take the form

t	Det L_t
1	S_1
2	$S_1^3 + S_3$

3	$S_1^6 + S_1^3S_3 + S_1S_5 + S_3^2$
4	$S^{10} + S_1^7S_3 + S_1^5S_5 + S_1^2S_3S_5 + S_1S_3^3 + S_3S_7 + S_5^2 + S_1^3S_7$

Each value of L_t calculated independently may be equal to 1 or zero. The logical expressions for a majority coincidence circuit (MCC) containing n inputs and t outputs are of the following form:

$$\begin{aligned}
&\text{Output } 1 = \det L_1 \vee \det L_2 \vee \det L_3 \ldots \det L_j \vee \ldots \vee \det L_t \geqslant 1\\
&\text{Output } 2 = \det L_2 \vee \det L_3 \ldots \det L_j \vee \ldots \vee \det L_t \geqslant 2\\
&\text{Output } 3 = \det L_3 \ldots \det L_j \vee \ldots \vee \det L_t \geqslant 3\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (2)\\
&\text{Output } j = \det L_j \vee \ldots \vee \det L_t \geqslant j\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\text{Output } t = \det L_t \geqslant t
\end{aligned}$$

[6].

A block-diagram of the MCC is described by the expressions [2]. If we add a scheme used to analyse the determinants on logical 1 or zero, we can get rigorous equalities as seen from fig.3, where a block-diagram for the calculation of the determinant for t = 1, t = 2, t = 3 and t > 4 is given. It should be stressed that the scheme in fig.3 is drawn so that PROMs or PLAs having 2m inputs for variables can be used to calculate the de terminants. This allows one to extend the range of values of n for which the table method of solution can be used. So, m = 10 the straight method of decoding can be used only for t = 2 because modern PROMs or PLAs have 18 - 20 inputs for variables. In addition, the method of executing combined operations in $GF(2^m)$ field is extensively used in such a scheme and below. The essence of the method lies in that the capacity of PROMs or PLAs used for the calculation of complicated expressions is independent of its complexity but determined by the number of variables [5]. For example, to calculate $S_1^3 + S_3$, S_2S_5 and so on, it is enough to use only one PROM containing 2m inputs for variables because raising to a power and addition can be taken into account by module programming.

Let us consider an example for t = 3. We have $S_1 = \det L_1 = a^{23} \neq 0$, $\det L_2 = a^{69} + a^{61} = a^6 + a^{61} = a^{46} \neq 0$, $\det L_3 = a^{138} + a^{69}a^{61} + a^{23}a^{35} + a^{122} = a^{12} + a^4 + a^{58} + a^{59} = a^{54} \neq 0$, $\det L_4 = a^{41} + a^{33} + a^{24} + a^{14} + a^{16} + a^{17} + a^{59} + a^7 = 0$.

2.3. Determination of the event coordinates

The next step after multiplicity calculation is to determine the coordinates X_i (i = 1,2,..,t) as fast as possible. For this purpose the author has suggested to use the decoding method of the equation of mistake location which is well-known from algebraic coding theory

$$X^t + \sigma_1 X^{t-1} + \sigma_2 X^{t-2} + \ldots\ldots \sigma_t. \qquad (3)$$

This equation is called a coordinate one [8]. As

$$S_i = \sum_{i=1}^{t} X_i^j , \qquad (4)$$

the calculation of the coordinates of events can be executed simultaneously with the determination of multiplicity t. As known, for $t < 5$ the roots of eq. 3 can be found by the table method [10 - 13]. For example, consider the cases $t = 2$ and $t = 3$ separately.

For $t = 2$ we have

$$X^2 + \sigma_1 X + \sigma_2 = 0, \qquad (5)$$

where $\sigma_2 = S_1^3 + S_3/S_1$ and $S_1 = \sigma_1$. Substituting $X = \sigma_1 y$, eq. (5) reduces to the form $Y^2 + Y = \gamma$ (6), where $\gamma = \sigma_2/\sigma_1^2$, $X_1 = \sigma_1 Y_1$, $X_2 = \sigma_1 Y_2$ and $Y_2 = Y_1 + 1$.

In the author's opinion, the algorithm described in [18] is most simply realized with the aid of combined operations. So, if

$T(r) =$ o, then

$$y_1 - \sum_{i=0}^{m-1} r_i y_i \qquad (6)$$

The values of y_i are determined from the relation

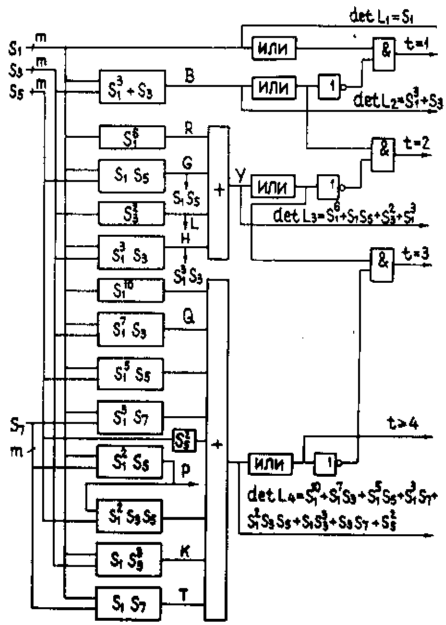

Fig.3. Block-diagram for calculating the determinants in $GF(2^m)$ for $t = 1 - 4$.

$$y_i^2 + y - \begin{cases} a^i & \text{for } Tr(a^i) = 0 \\ a^i + a^k & \text{for } Tr(a^i) = 0. \end{cases}$$

After not complicated calculations, we obtain the following values of $y_0 - y_5$ for $m = 6$:

$y_0 - a^0 -$ I00000 for $a^i - 0$, $y_I - a^{II} -$ II0000 for $a^i - a^{36}$,
$y_2 - a^{55} -$ 0IIII0I for $a^i - a^{32}$, $y_3 - a^0 -$ I00000 for $a^i - 0$,
$y_4 - a^{23} -$ I00I0I for $a^i - a^{38}$, $y_5 - a^{43} -$ III0II for $a^i - a^{I9}$ [8,9].

So, in GF(2^6) $y_I = y_{I0}\, a^0 + y_{II}\, a^I + y_{I2}\, a^2 + y_{I3}\, a^3 + y_{I4}\, a^4 + y^5_{I5}\, a^5$
и $\gamma = \gamma_0\, a^0 + \gamma_I\, a^I + \gamma_2\, a^2 + \gamma_3\, a^3 + \gamma_4\, a^4 + \gamma_5\, a^5$, then from (6) we have
$y_{I0} = \gamma_0,\ y_{II} = \gamma_0 + \gamma_I + \gamma_5,\ y_{I2} = \gamma_I + \gamma_2 + \gamma_3 + \gamma_5,\ y_{I3} = \gamma_0,$
$y_{I4} = \gamma_0 + \gamma_3 + \gamma_5$ и $y_{I5} = \gamma_0 + \gamma_I + \gamma_2 + \gamma_4 + \gamma_5.$ (7)

Fig.4 presents a block - diagram of solving eq. (5) where use is made of one PROM having 2m inputs for variables. The speed of the processor is calculated from the expression

$$T_{K2} = T_y + 2T_s + T_\gamma,$$

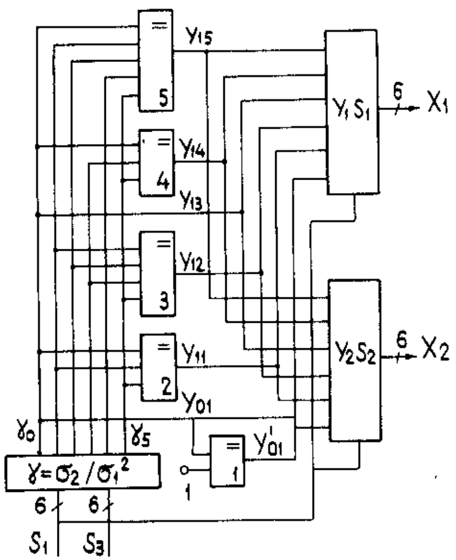

Fig.5. Block- diagram for the calculation of the 2nd degree coordinate equation in GF(2^m).

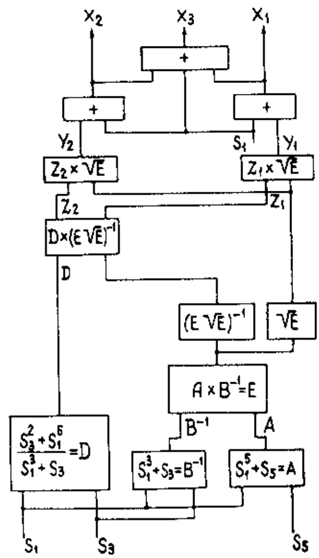

Fig.6. Block-diagramm for the calculation of the 3 d degree coordinate equation in GF(2^m).

where T_y is the time for multiplying two elements in GF(2^m), T_s the time of modulo 2 addition and T_γ the delay of one PROM. A fast solution (about 25 ns) can be obtained if logical elements MC10102 and parity checkers MC10160 are used instead of PROM. For instance, the sources are fired at $X_1 = a^0$ and $X_2 = a^2$. After simple calculations, we have $S_I = a^{I2}$, $S_3 = a^3$, $\sigma_I = a^{I2}$, $\sigma_2 = a^2$ and $\gamma = a^{4I} =$ IOIIIO. From eq. (7) we obtain

$$\sigma_3 = R_3 + R_I\sigma_2 \quad (10)$$

$$R_5 + R_3\sigma_2 + R_I\sigma_4 \quad (11)$$

$$R_7 + R_5\sigma_2 + R_3\sigma_4 \quad (12), \text{ where}$$

$$R_3 = S_3 + S_I^3 = B$$

$$R_5 = S_5 + S_I^2 S_3 = V$$

$$R_7 = S_7 + P + V + S_I^7$$

Equation (11) does not contain R_7 and thus it is calculated more easily

$$\sigma_4 = \frac{R_5}{R_I} + \frac{R_3\sigma_2}{R_I}$$

So, after a prelimanary calculation of σ_2, from (10) and (11) we can find σ_3 and σ_4. As PROMs with a smaller number of inputs for variables are required to calculate these valuas, such an algorithm for obtaining σ_3 and σ_4 has as a whole no influence on the speed.

V. Use of the theory of Reed-Solomon codes

If some cluster events are simultaneously registered in a MDCP as shown in fig.6 [17], it is worthwhile to use the theory and practice of nonbinary BCH-codes (Reed-Solomon (RS) codes [18]. The advantage of this approach can be explained as follows. Events with clusters are most often registered in real experiments. According to the decoding method of binary BCH-codes, a cluster b in length having t units is processed as if t independent sources be fired. From the physicists' viewpoint, a cluster is most commonly a one-particle event. Thus, to determine the number of clusters by the theory of RS-codes, it is necessary to solve the determinants of lesser orders. For example, if two clusters b = 4 in length are registered in a MDCP, in the first case the determinant of the 9-th order should be calculated whereas, according to the theory of RS-codes, it is enough to solve the determinant of the 3d order.

Using the syndrome method, $2^m - I - 2t$ information symbols are considered as zero ones, the signals registered in the MDCP are divided into groups with m bits in each group, and the maximal cluster length is m. Since n >>t under experimental conditions, information compression with the coefficient $K_c = n/2t$ takes place.

VI. Syndrome coding method for sequential systems

Two reasons make us use economical, sequential methods of event registration in MDCPs.

1. A great number of experiments is planned in which a large multiplici-

ty, t, of 15-30 is registered in one hodoscopic plane. Thus, to solve coordinate equations for $t > 5$, the author has suggested to use the sequential decoding methods which are well-known from the theory of error correcting codes. The most economical decoding method is described in paper [16]. The idea of creating coordinate processors for large multiplicity is taken from this article. Preliminary calculations show that for $t = 20$ and $n = 1000$ all 20 event coordinates can be found for 10 - 15 μs.

2. There are many experiments in high energy physics [19] and applied research, e.g. in medicine, where it is enough to register one cluster. A scheme of the two-coordinate position-sensitive detector is given in fig.7.

A series of pulses is generated from one charged particle in two planes. It is necessary to determine the centre coordinates of the cluster. The signals from the planes are read out with the help of a magnetostrictive delay line [25]. An economical coding scheme for cluster registration based on Fire coding devices is suggested [22]. The tables of the Fire codes for n = 15 - 1200, which can be used to create a coding

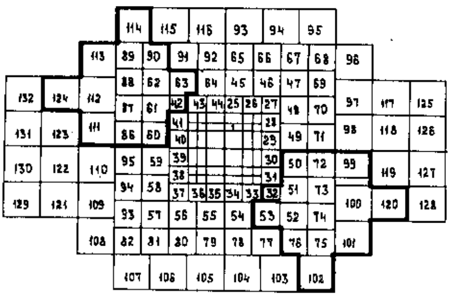

Fig.7. Example of the registration of two events with clusters in a hodoscopic calorimeter.

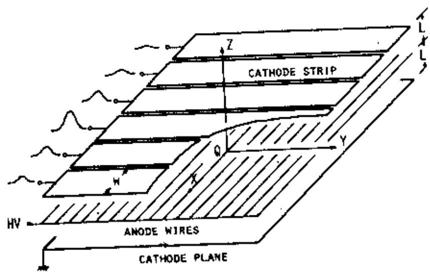

Fig.8. Diagram of the two-coordinate position-sensitive detector.

device, are presented in paper [23]. The number of bits in the coding device is equal to the degree of the generating polynomial g(X) because an information word equals zero. For $b = 3$ and $n = 15$ we have

$$g(X) = X^9 + X^6 + X^5 + X^4 + X + 1$$

The compression coefficient is characterized by the ratio n/r. Besides, the efficiency of compression grows with increasing n on condition that $b \ll n$. It is shown [26] that there are optimum Fire codes for some cases when b = 3 or 4. For example, as it follows from [23], a 6-bit register can be used instead of a 9-bit one for b = 3. So, K_C = 4096/14 for b = 4 and n = 4096, and a PROM can be used for parallel decoding.

VII. Fast algorithm for multlplication in Galois field

Multiplication in Galois field it is performed simultaneously over only two elements or their logarithms [14, 16; 23]. Below we give a description of the algorithm with the help of which multiplication over an arbitrary number of elemente can be performed [26]. Consider the essence of the algorithm illustrating the operation of multiplication in $GF(2^4)$. The base of the algorithm is the method of parallel data compression used in the schemes of fast multiplication of usual numbers. Fig.8 gives two examples which illustrate the proposed algorithm. Such a device is called a cyclic compressor because addition is performed modulo 2^m-1.

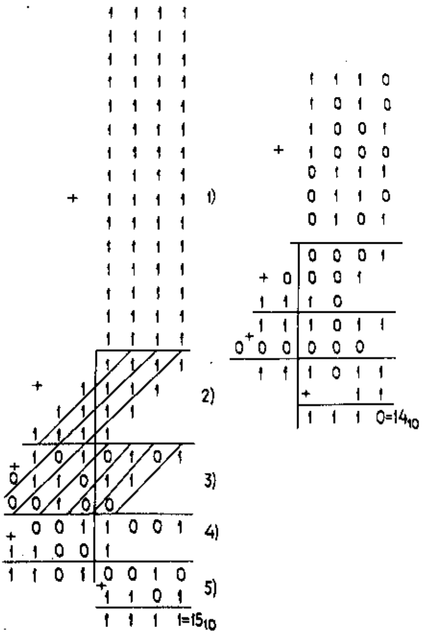

Fig.8. Two examples illustrating the operation of a cyclic compressor in $GF(2^4)$.

The first example on the left corresponds to a simultaneous multiplication of 15 elements $a^0 = a^{15}$, and cyclic compression is executed over the degrees of multiplicands. After three compression steps, we get two addends divided into 2 parts. Besides, the second part of the cyclic sum (left) corresponds to a maximum number (11010000) which equals the sum of cyclic carries arising from the compression of 15 addends 1111. At the last step high-order bits of 1101000 are added to 1101 modulo 15. The complete result is 1111. Fig.8 on the right shows an example of the sum of the multiplicand degrees a^{10} a^{14} a^9* a^8 a^7 a^6 a^5 = a^{45} a^{14} = a^{14} in $GF(2^4)$. This example can be used as a basis for the creation of a cyclic compressor (fig.9). The schemes for the calculation of the logarithms, which are in essence PROMs, are not given in this figure. Parallel (n,k)- counters can be used for creating cyclic compressors as well as usual PROM. [29]. As shown ih fig.8, the cyclic compressor is composed of a group of (15,4)-, (4,3)- and (3,2)- counters. The number of cascades of parallel counters, M, equals 3 for m = 3 - 7 and 4 for m = 8 -15. Fig. 10 shows the schemes with the help of which it is possible to determine the structure of the counters and to create the corresponding cyclic compres-sor. The time of multiplying $2^m - 1$ multiplicands can be calculated from the expression

$$T_m = 2T_p + 2T_s + (T_{c1} + T_{c2} + \ldots\ldots T_{cM}),$$

T_p is the delay in a PROM used to calculate algorithms and antilogarithms, T_s is the time of summation modulo $2^m - 1$ and the delay times in the corresponding parallel counters are given in brackets.

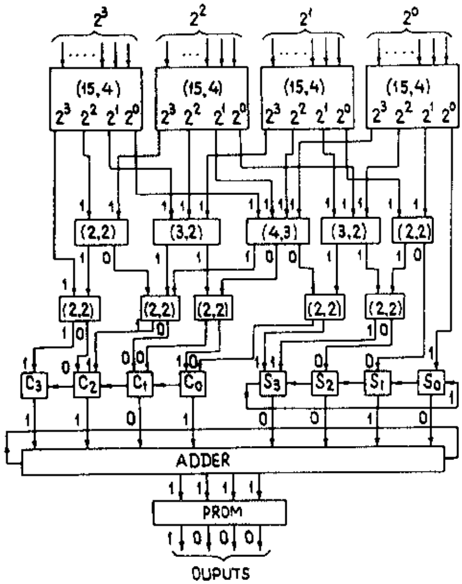

Fig.9. Scheme for simultaneous multiplication in $GF(2^4)$.

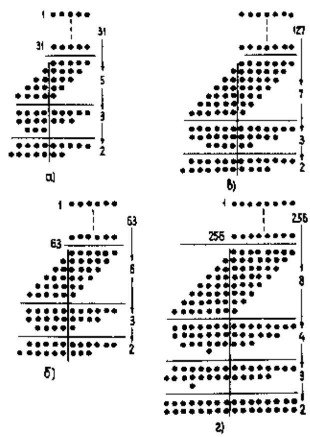

Fig.10. Diagram for the calculation of cyclic compressors at $m = 5 - 8$.

Coclusion

It is shown that the algebraic coding theory can be successfully used for data compression registered in MDCPs and for the creation of special-purpose coordinate processors. The economical algorithm of executing a simultaneous multiplication over a great number of elements has been suggested for fast speed operations in complex exprssions in Galois field. The method of syndrome coding can be used in other multichannel systems for data registration.

References

I. Badier J., et al., CERN/EP 80-36, Geneva, 1980.
2. Nikityuk N.M., JINR No. E10-88-28, Dubna, 1988. Submitted to the Int. Joint Conference of ISSAC-88 and AAECC- 6, Rome, 1988.
3. Nikityuk N.M., Radzhabov R.S., Shafranov M.D., Nucl. Instr. and Methods, 1978, vol. 155, No.3.
4. Nikityuk N.M., Radzhabov R.S., Shafranov M.D., Pribory i tekhnika eksperimenta, 1978, No.4, p. 95.
5. Messy J.L., IEEE Trans. on Inf. Theory, 1965, vol. IT-11, No.4, p.580.
6. Gaidamaka R.I., Kalinnikov V.A., Nikityuk N.M., Shirikov V.P., JINR, P13-82-628, Dubna, 1982.
7. Nikityuk N.M., JINR, P11-87-54, Dubna, 1987.
8. Nikityuk N.M., JINR, No. P11-80-484, Dubna, 1980.
9. Nikityuk N.M., JINR, No. P10-88-853, Dubna, 1989.
10. Polkinghorn F., IEEE Trans. on Inf. Theory, 1966, vol. IT-12, No.4.
11. Banerji R.B., Proc. IEE, 1961, vol.49, No.10, p.1585.
12. Chien R.T., Cunmingham B.D., IEEE Trans. on Inf. Theory, 1969, vol. IT-15, No.2.
13. Okano H., Imai H., IEEE Trans. on Comput., 1987, vol. C-36, No. 10.
14. Nikityuk N.M., JINR, No. P10-89-16, Dubna, 1989.
15. Berlekamp E.R., Rumsey H., Solomon G., Information and Control, vol.10, 1967, p.553-564.
16. Berlecamp E.R., IEEE Trans. on Information Theory, 1964, vol. IT-11, No.4, p.577.
17. Chien R.T., IEEE Trans. on Inf. Theory, 1964, vol. IT-10.
18. Nelson K.S., Erwin A.R., IEEE Trans. on Nucl. Science, 1983, vol. NS-30, No.4.
19. Nikityuk N.M., JINR, No. P10-88-854, Dubna, 1989.
20. Breskin A., Charpak G., Demierre C et al. Nucl. Instr. and Meth., 1977, vol. 143, No. 1.
21. Hendrix J., Furst H. IEEE Trans. on Nuclear Science, 1980, vol. NS-27, No.5.
22. Jeavons A.P., Fora N., Lindberg B. et al. IEEE Trans. on Nucl. Science, 1976, vol. NS-23, No. 1.
23. Nikityuk N.M., JINR, No. R10-88-742, Dubna, 1988.
24. Wagner W., IEEE Trans. on Inf. Theory, 1970, vol. IT-8, NO. 5.
25. Elspas B., IRE Trans. on Inf. Theory, 1962, vol. IT-8, No. 1.
26. Bartee T.C., Sneider P.I., Information and Control, 1963, vol.6, No.1
27. Nikityuk N.M., JINR, No. P11-88-852.
28. Ho I.T., Chen T.C., IEEE Trans. on Comput., 1973, vol. C-22, No.8.
29. Gajski D.D., IEEE Trans. on Comput., 1980, No.5, p. 393.

30. Swartzlander E.E. IEEE Trans. on Comput., 1973, vol. C-22, No.11.

Appendix

The elements of Galoius field modulo $X^6 + X + 1$

$a^0 = 100000$	$a^{21} = 11011$	$a^{42} = 010111$
$a^1 = 010000$	$a^{22} = 101011$	$a^{43} = 111011$
$a^2 = 001000$	$a^{23} = 100101$	$a^{44} = 101101$
$a^3 = 000100$	$a^{24} = 100010$	$a^{45} = 100110$
$a^4 = 000010$	$a^{25} = 010001$	$a^{46} = 010011$
$a^5 = 000001$	$a^{26} = 111000$	$a^{47} = 111001$
$a^6 = 110000$	$a^{27} = 011100$	$a^{48} = 101100$
$a^7 = 011000$	$a^{28} = 001110$	$a^{49} = 010110$
$a^8 = 001100$	$a^{29} = 000111$	$a^{50} = 001011$
$a^9 = 000110$	$a^{30} = 110011$	$a^{51} = 110101$
$a^{10} = 000011$	$a^{31} = 101001$	$a^{52} = 101010$
$a^{11} = 110001$	$a^{32} = 100100$	$a^{53} = 010101$
$a^{12} = 101000$	$a^{33} = 010010$	$a^{54} = 111010$
$a^{13} = 010100$	$a^{34} = 001001$	$a^{55} = 011101$
$a^{14} = 001010$	$a^{35} = 110100$	$a^{56} = 111110$
$a^{15} = 000101$	$a^{36} = 011010$	$a^{57} = 011111$
$a^{16} = 110010$	$a^{37} = 001101$	$a^{58} = 111111$
$a^{17} = 011001$	$a^{38} = 110110$	$a^{59} = 101111$
$a^{18} = 111100$	$a^{39} = 011011$	$a^{60} = 100111$
$a^{19} = 011110$	$a^{40} = 111101$	$a^{61} = 100011$
$a^{20} = 001111$	$a^{41} = 101110$	$a^{62} = 100001$
		$a^{63} = a^0 = 100000$

Received by Publishing Department
on May 24, 1989.