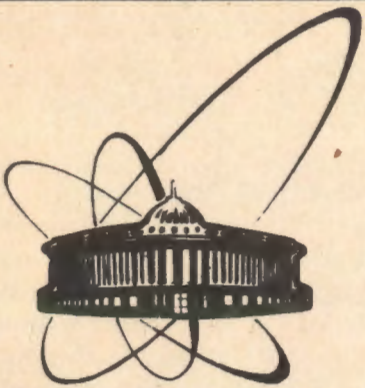


объединенный
институт
ядерных
исследований
дубна

E10-91-567

N.M.Nikityuk

USE OF PARALLEL COUNTERS FOR TRIGGERING

Submitted to "NIM"

1991

I. Classical parallel counters

1. *Discrete parallel counters.* A parallel counter is a device that determines how many its inputs are active (for example, in the logic one state). Parallel counters are used to construct fast multipliers, multiinput adders, associative processors [1], majority coincidence circuits for counting the number of hits registered in MPCs [2,3] and drift chambers [4] for triggering purposes. Parallel counters are effectively applicable to construction of special-purpose processors for event selection according to the number difference of hits registered in neighbouring hodoscopic planes [5]. To implement parallel counters, several methods can be used. Some of them are the following: 1)two-level gate network; 2)ROMs, PROMs and PLAs; 3) full adder array; 4) quasi-digital counters [1,9]. To make a n-input counter using a two-level gate, it is necessary a large number of gates so they are impractical for large values of n. ROMs and PROMs require a memory with 2^n words of k bits (k – is the number of outputs from the counter). This total storage equals $k2^n$ bits, and so such counters are impractical for large of n because modern detectors have thousands and more registration channels. PLAs are impractical for this reason, too.

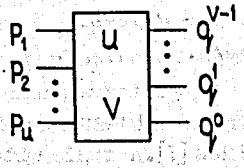


Fig. 1. Scheme of the (u,v) counter.

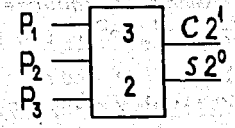


Fig. 2. Principal scheme of the (3,2)-counter.

Full-adder networks are of practical interest because they can be constructed for a large number of inputs n and a relatively small delay τ_c. Such a counter having n inputs, k outputs is named a

(n,k)-counter. Connections between inputs and outputs obey the relation

$$\sum_{i=1}^{n} a_i = \sum_{i=0}^{k-1} q_i 2^i, \quad a_i = 0,1; \; q = 0,1,$$

where $n = 2^k - 1$, $k = 2,3,4\ldots$.

A simple method for constructing parallel counters with network full--adders is described [1;6,7]. The essence of the method for construction of (n,k)-counters which have a large number of inputs is to use (u,v)-counters having a smaller number of inputs as shown in Fig.1. The outputs of a (u,v)-counter have binary weights $2^0, 2^1, 2^2, \ldots 2^{v-1}$. A full one-bit adder is a (3,2)-counter (Fig.2). A MC10180 microcircuit having τ_c 4,5 and 2,2 ns from the $p_1 - p_3$ inputs to the S (sum) and C (carry) outputs is used. This is the reason why addition of delays (for example a MC10101 microcircuit is requered in the full-adder network to get a high time-resolutin of parallel a counter). Fig 3 a), b) and c) presents (4,3)-, (5,3)- and (7,3)-co-

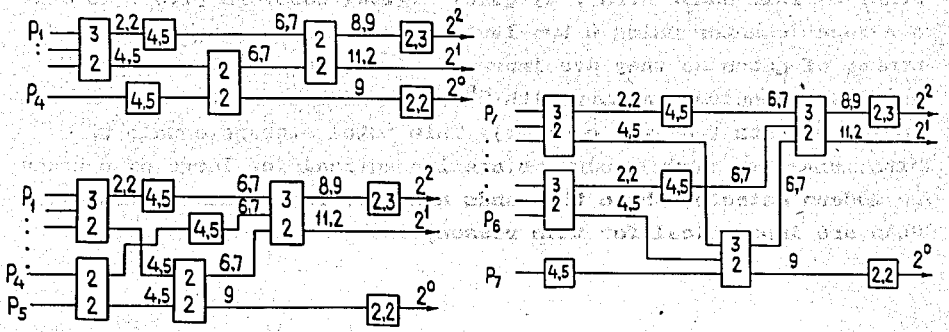


Fig. 3. Principal schemes: a) (4,3)-; b) (5,3)- and c) (7,3)-counters. Number in squares and top on the lines - delays. 2^0, 2^1, 2^2 - the wieghts of the outputs.

unters. Basically, the design of (n,k)-counters includes the grouping of counter inputs into sets of three lines [7]. A full adder reduces each set of three lines to a line with weight 1 (the sum output of the adder) and a line with weight 2 (the carry output of the adder). This results in approximately $n/3$ lines of weight 1 and n/3 lines of weight 2. Each of these sets of lines is separated into groups of three, and reduction continues until only one line of each weight remains. The carry shower process for $n = 31$ is given in Fig. 4. A

principal scheme of (31,5)-counters is shown in Fig.5. The number of full-adders N_{add} required for (n,k)-counters can be calculated from a simple relation

$$N_{add} = n - k.$$

Since the delay of a three-input counter is a one-adder delay, an n-input counter has a delay limited by [1,7]

$$[log_3(n - 1 + log_2 n] < \tau_c < 2(log_2(n)] - 1.$$

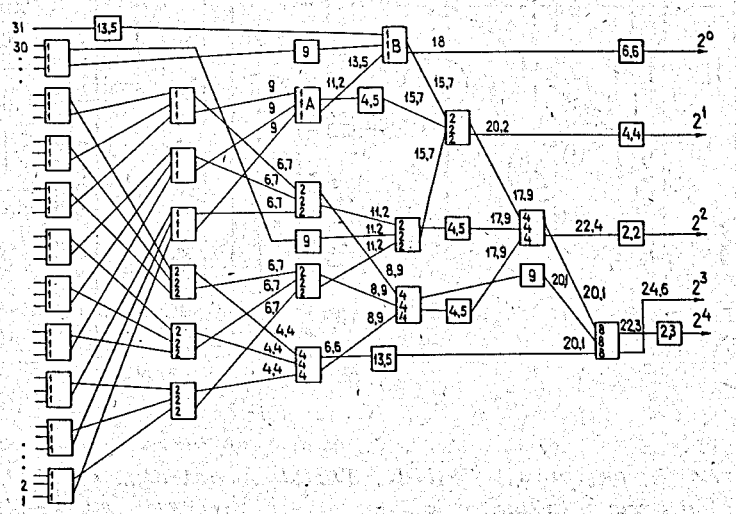
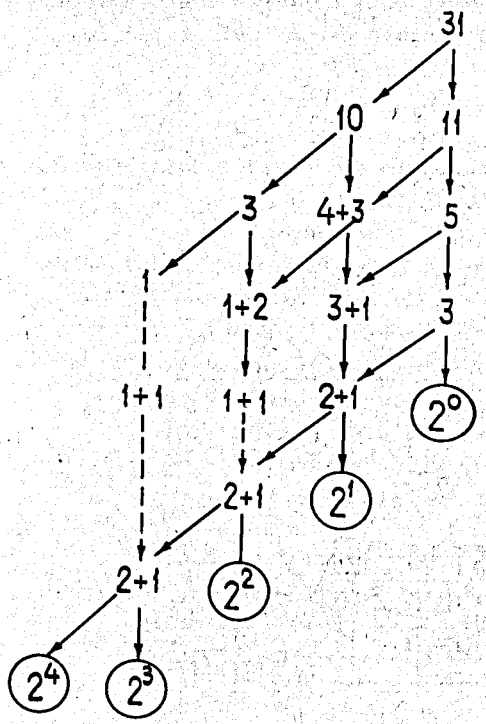


Fig. 4. Principal scheme of (31,5)-counter. The numbers in squares delays; the numbers on the top of the lines - total delays; the numbers in rectangles - the wieghts of the inputs lines.

The charateristics of some parallel counters is given in Table. The power consumption is given by condition that the loading resistor is 620 Om. As it follows from Table the *FASTBUS* is usefull for $n > 31$.

Quasi-digital counters. It is clear that delay τ_c is the most essential characteristic of a parallel counter. The fastest parallel



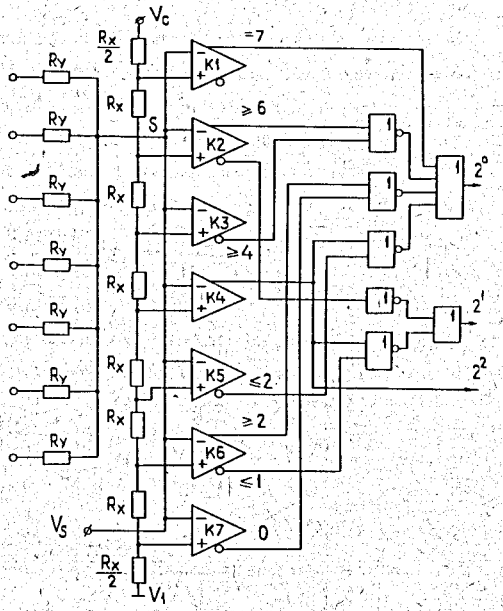


Fig. 5. Carry shower for n = 31 [7]. Dotted arrows indicate no processing.

Fig. 6. Quasi-digital (7,3)-counter. K1 – K7-comparators.

counter can be constructed using quasi-digital techniques [1,8]. A circuit for a seven-inputs quasi-digital conter is shown in Fig. 6. The total delay of this counter is

$$\tau_c = T_s + 5 \text{ ns},$$

where T_s is the delay added by a resistive network ($\tau_c = 5$ nc for $n = 31$) [1]. It is supposed that *ECL*-logic is used for making a quasi – digital counter. The disadvantage of this method is that analog networks are used and so it is difficult to suppres noises. That is

why quasi-digital counters have an input limitation of 50 [1]. It is clear that large counters can be built using quisi-digital counters

Table

Characteristics of some parallel counters

Counter Parameter	(3,2)	(4,3)	(5,3)	(7,3)	(15,4)	(31,5)	(63,6)	(127,7)
The number MC10180	0,5	1,5	2	2	5,5	13	28,5	60
Delay, ns	4,5	11,2	11,2	11,2	17,9	24,6	31,3	40
The number MC10101	0,25	1,5	1,5	2	5	8,75	16	32
Power dissipate, Wt	0,25	0,92	1,12	1,22	3,3	7,08	14,9	30

at the first stage and ripple-carry adders at the subsequent stages as described above.

2. Parallel compressors for fast triggering in calorimeters. Parallel counters are widely used for the construction of parallel compressors of fast multi-operand adder devices [9,10]. Tree adders are in use for the triggering in the calorimeters [11]. But it takes much time and requires many adders for such discrete triggers. Analog adders have number of lacks: unstability of characteristics and restricted functional capabilities. As shown in [12], parallel compressors can be used for fast triggering in the calorimeters. We designat this device as (M,m)-compressor where M is the number of addends and m the number of bits in a summand. Parallel counters are the essential part of parallel compressors. For clearness let us consider an example. Assume that it is necessary to add fast 15 15-bit digits as shown in Fig. 7. Summation for n = 15 is divided in four stages. The number of bits of equal weights is counted at the first

stage with the aid of (15,4)-counters. For example, the 1-st column (right) has 11 ones of weight 2^0 or 1011_2, the 2-nd column has 8 ones of weight 2^1 and so on. The 15-th column has 5 ones of weight 2^{14}. Four words are obtained after the first stage of compression and after the second stage number of addendums is compressed to 3 ones. A (4,3)- counters are used at the second stage. At least two words are summed with the aid of a 15-bit adder. The scheme in Fig.7 can be used to construct a (15,15)-compressor. Logic connection in Fig 8,a is performed with the help of the scheme presented in Fig.

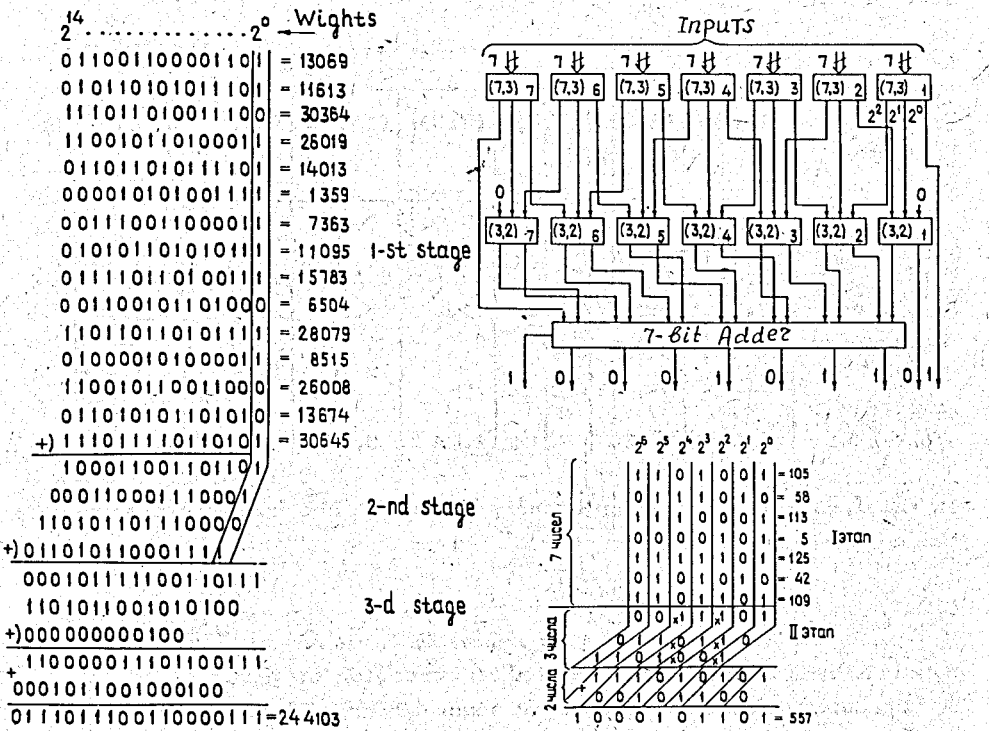


Fig. 7. Example of column compression of 15 15-binary numbers.

Fig. 8,a) Scheme of (7,7)-compressor; b) column compression of 7 7-binary numbers.

8,b. The diagram for the construction (15,7)-, (31,7)-, (63,7)- and (127,7)-compressors is given in Fig. 9. Binary numbers 1 and 0 are marked by points [12]. From Fig.9 it is seen that the (15,7)-compressor consists of seven (15,7)- counters at the first stage, four (4,3)- and (3,2)-counters ((2,2)- is the specific case of a (3,2)-

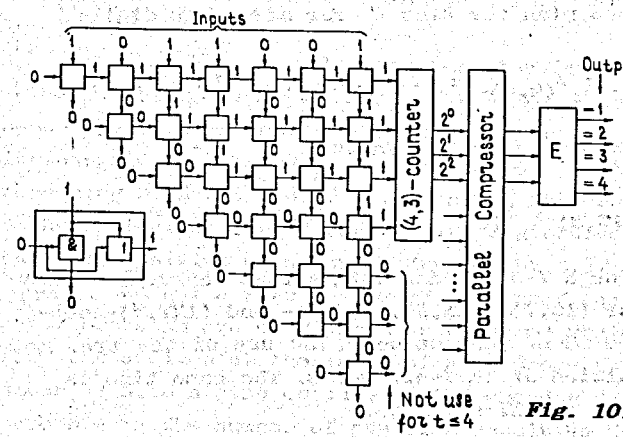


Fig. 10. Scheme of sequential-parallel compressor. 1 - OR-gate; E - encoder; & - AND-gate.

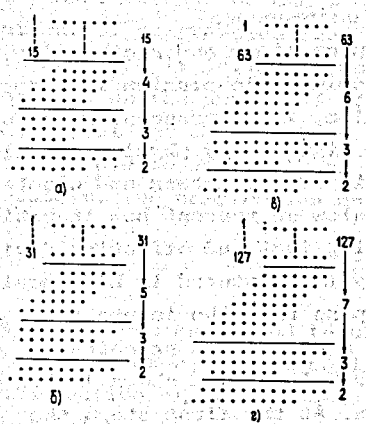


Fig.9. Diagrams for the construction (15,7)-; (31,7)-; (63,7)-; and (127,7)-compressors.

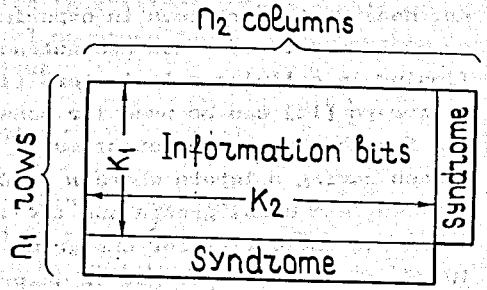


Fig. 11. Structure scheme of the iteration code.

counter) and so on. The last stage of compressors is always a binary adder. By analogy, parallel compressors can be constructed for arbitrary M and m. The number of stages L is equal to that of binary digits necessary to present M. It should be marked that a parallel compressor for large M can be made using a compressor with smaller M. The time of compression T_c equals the sum of delays of (n,k)-(k,q)-; ... (3,2)-counters plus the time T_s for add m-bit digits

$$T_c = T_{(n,k)-} + T_{(k,q)-} + \ldots\ldots T_{(3,2)-} + T_s.$$

For a (7,7)-compressor we have (see the Table)

$$T_c = T_{(7,3)} + T_{(3,2} + T_S = 28 \text{ ns}.$$

Assume that T_s = 12 ns for a 7-bit adder with a fast carry. The calculations show that T_c for (15,7), (31,5), (63,6)- and (127,7)-compressors are 48, 56,64 and 68 ns respectively. The use of the tree method takes 60 ns for addition of 15 7-bit words. The same time is needed for the compression of 127 summands.

II. Sequential-parallel compressors

During the last few years cellular automations and rectangular universal iterative arrays have shown considerable promise in the solution of problems of pattern recognition, image processing and so on. Many of the problems in experimental high energy physics are also of this nature: track reconstruction in wire chambers and cluster finding in cellular calorimeters [13]. Below we present how iterative arrays [14] can be used for constructing fast and effective devices for the determination of multiplicity t registered in hodoscopic planes having n inputs where $n >> t$. Suppose that the inputs n are divided into equal groups and $t < 4$. For simplicity a principal scheme of the device for one group having 7 inputs is shown in Fig. 10. There are three stages of compression. At the first stage the ones from the array inputs are shifted and compressed at their outputs with the aid of the itterative array. Then they are counted with the help of (4,3)-counters. The outputs of the counters are grouped and connected to the inputs of the parallel compressor.Binary digits are

encoded to unitary position codes by the encoder E. The delay T_C of such device is

$$T_C = 2T_G\, n/k + T_{PC} + T_{comp} + T_E,$$

where T_G is the delay of a gate, k the number of inputs in one group, T_{comp} the time of compression and E the delay of the encoder E.

III. Parallel counters for large inputs

Use of the *syndrome coding method* [15]. A great number of multichannel position-sensitive detectors is in use in high energy physics experiments. There is a small number (10-15%) of signals for processing with the aid of electronics. The essense of the method is the following. The number of hits registered simultaneously in a position-sensitive detector is denoted by t. If the detector does not operate, we get a zero code word, and ones occurring when the detector operates are considered as an error vector to the code word. This word arrives at the input of a syndrome shaper (encoder). The number of bits at the outputs of the encoder decreases to $N = log_2 n$, where n - is the number of position-sensitive detectors in a hodoscopic plane. For n = 63 and t = 3, N = 18. In other words, we have the effect of compression from a 63-bit unitary position code to a 18-bit code. As it follows from the coding theory, N carries information both on the number of $2t$ hits and their t coordinates. The efficiency of the method increases with increasing n. As a result, it is possible to use *PROMs* or parallel counters for a lesser number of inputs n. The compression coefficient

$$K_c = n/N.$$

The author has suggested to use iteration codes for the determination of the number of hits registered in pixel detectors. The method of construction of iteration codes can be explained using the following examples. At first information X_{ij} = 0,1 present as a matrix [16]

	Information symbols					Row parity check
	X_{11}	X_{12}	X_{13}	X_{ij}	X_{1m}	$\sum_{j=1}^{m} X_{1j}$
	X_{21}	X_{22}	X_{23}	X_{2j}	X_{2m}	$\sum_{j=1}^{m} X_{2j}$
	.....					
	$X_{\alpha 1}$	$X_{\alpha 2}$	$X_{\alpha 3}$	$X_{\alpha j}$	$X_{\alpha m}$	$\sum_{j=1}^{m} X_{\alpha m}$
Column parity check	$\sum_{i=1}^{\alpha} X_{i1}$	$\sum_{i=1}^{\alpha} X_{i2}$	$\sum_{i=1}^{\alpha} X_{i3}$	$\sum_{i=1}^{\alpha} X_{ij}$ ...	$\sum_{i=1}^{\alpha} X_{im}$	

(1)

Then parity symbols are added to each row and column. Figure. 11 gives a scheme of the iteration code. We suppose that each symbol in the pixel detector corresponds to a matrix element. As information symbols are zero according to the syndrome method, the parity bit carries data on hits registered in the pixel detector. It should be noted the class of iteration codes is very wide because combinations of different codes having an algebraic structure can be used for iteration. As known from the coding theory, there is a simple relation for usual correcting codes $t = d - 1$, where d is the coding distance. For iteration codes $d = d_1 d_2 \ldots d_\gamma$, where γ is the number of codes used for iteration. The number of codes used for iteration is called code dimensionality. The simplest two-dimension iteration codes are *OR-OR* ($d = d_1 d_2 = 1{,}5 \times 1{,}5 = 2$) and *PARITY-PARITY* ($d = 2 \times 2 = 4$). Then the coding distance of a four-dimensional *OR-OR - PARITI-*

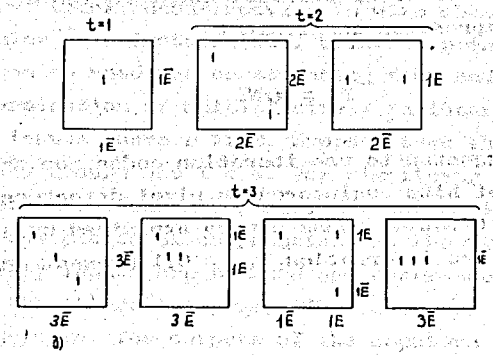


Fig. 12. View of events in two-dimensional *plane; t=1-4. E - even; Ē-odd.*

-PARITY iteration code is 8 and thus $t = 7$. Below we show how the simple 4-dimensional iteration code can be used for data compression registered in a multichannel pixel detector and for the consruction of a parallel counter for large $n > 500$ [17]. Figure. 12 depicts the views of events in two-dimensional planes for $t = 1 - 4$. The positions of the events are not taken into accaunt. We determine the

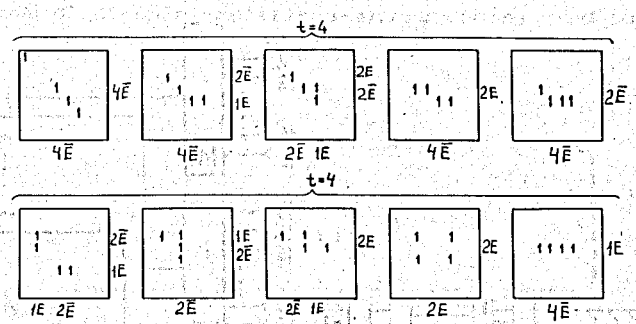


Fig. 13. Meaning of parity at t = 5 and 6.

number of events by counting the number of parities (odd and even) in rows (X-coordinate) and columns (Y - coordinate). As seen from Fig. 12, for t = 1 we have one *ODD* for the *X*-coordinate and one *ODD for* the *Y*-coordinate only. For $t = 2$, 3 and 4 we have two, four and 10 different, pictures respectively. Figure 13 presents similar pictures for $t = 5$ and $t = 6$ (E - even and $\bar{E}$ - *ODD*). Symmetrical pictures which give an identical result are not shown. For example, let us assume that a pixel detector has 31x31 pixels arranging in a matrix with 31 rows and 31 columns. For simplicity Fig. 14 gives an essential part of the scheme for one row . As zero is an even digit, some *OR*-gates are required to check if a row or a column has at least one hit. Couning the number of parities and combining the encoder outputs by connecting them to the corresponding inputs of the *AND-gates*, we can determine the number of hits registered in a pixel detector fastly and economically. It is not difficult to cal-

culate the delay of pulses T_C,

$$T_C = T_P + 2T_G + T_{PC} + T_E,$$

where T_P – is the delay of a parity checker, T_G the delay of an *AND* gate, T_{PC} – the delay of a (31,5)-counter and T_E delay of an encoder. If *ECL* logic is used then T_P = 12 ns (for 31 inputs), T_G = 2 ns, T_{PC} = 25 ns and T_E = 4 ns. Then T_C = 43 ns ! A scheme of the parity checker for 144 inputs having 12 ns delay is shown in Fig. 15. It should be noted that for the determination multiplicity $t > 7$ it is necessary to use iterative codes having a greater coding distance d. There are two methods to increase the value of a d. 1. Use two additional diagonal parity checks. 2. The use of codes having greater coding distance. It is known that the modified Hamming code has a coding distance, d, of 4. Then the iteration Hamming-Hamming

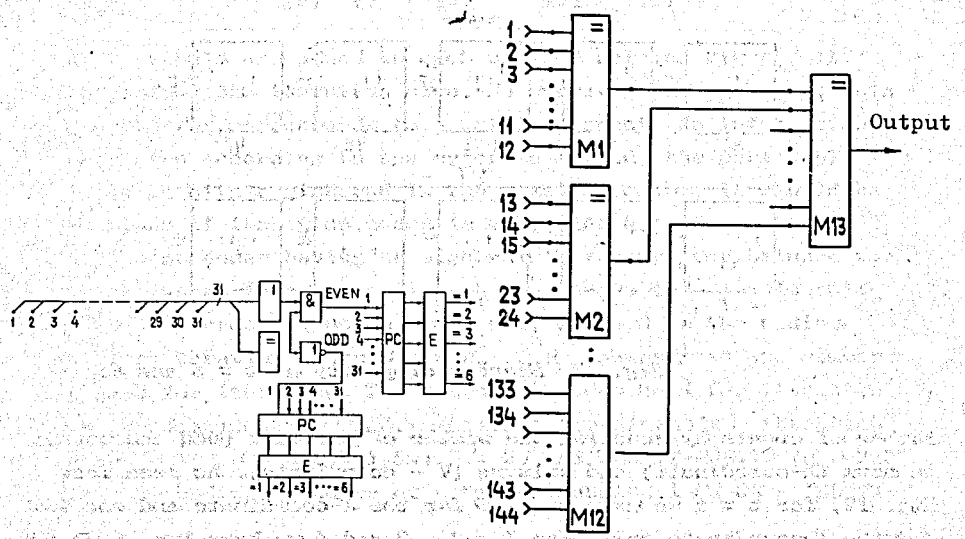


Fig. 14. Essential part of the device parallel counter for the pixel detector having 961 pixels. PC – parallel counter; 1 – OR-gate; & – AND.

Fig. 15. Scheme for calculating parity checks for n = = 144.

code has d = 16 and t = 15. In addition to this, $\nu = t/2$ coordinates can be determined with the aid of such code according to the relation [18]: $\nu = (d - 1)/2$.

IV. Use of superimposed codes

It is convenient to use them in light coding systems and when signals have a small amplitude (analog signals). It is important that light and electronic amplifiers-mixers can be used to calculate

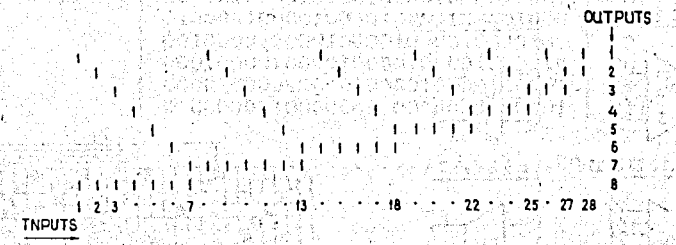


Fig. 16. Coding $H_{28,7}$ matrix.

the syndrome and it is unnecessary to use a modulo two-adder havig inverters. However, the compression coefficient K_c of such codes is smaller than that of the well-known *BCH*-codes [18] used for data processing in pixel detectors which syndrome is calculated not by modulo-2 but by the Boolean sum rules. Nevertheless, such codes are important in high energy physics experiments for data compression in scintillation hodoscopes and calorimeters [15,21-23]. If light signals are coded, photomultipliers can be used to calculate a syndrome. Below we show how a superimposed codes can be used for event selection. In accordance with the syndrome coding method, before the determination of myltiplicity t, information from the hodoscope plane with registration channels n is compressed to N with the aid of a coding matrix. This allows fast *PROMs* to be used for further analysis. In the general case the coding matrix $H_{n,M}$ consists of different n columns and N rows. Each column is connected to a registration channel and every row to the inputs of a signal mixer-amplifi-

er. The author has suggested an effective coding matrix having the following characteristics [22]: 1. There are $n = C_N^2$ columns and N rows. 2. Only two ones are in each row (the branching coefficient of signals equals 2) 3. The compression coefficient $K_c = C_N^2/N$. For example, let n be 28 and then $N = 7$. Figure 16 shows schemes of the coding matrix $H_{28,7}$ and Fig.17 shows scheme for calculation of the 1-st 3-td, 4-th and 8-th syndrome bits. Optical fibers and mixers can be used for coding.

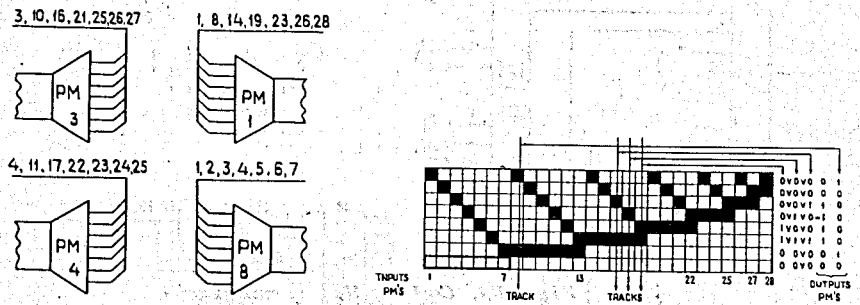


Fig. 17. Scheme for calculating 1-st; 3-rd; 4-th and 8-th bit syndrome.

Fig. 18. Coding mask-detector for t = 1, n = 28 and N = 7.

Using such coding matrix, one event with a tripple cluster as $3 \geqslant d \leqslant 4$ can be selected. The coding matrix can be created as a mask, where black squares are scintillators; and white squares, usual glass (Fig. 18). The coding device algorithm is the following [20]. Let us consider some examples. 1. The only one particle is registered in the plane, i.e. $t = 1$ and there is no cluster. In this case the syndrome weight can be equal to two. Thus, if all matrix columns are different, the 7-bit code carries information on the coordinate of a hit. 2. If a double cluster is generated, then the syndrome weight w is equalto 3.3.A tripple cluster was generated from a single particle in the plane. Thus, the syndrome weight is equal to 4.

The method of coding for $t > 1$ is given in [21]. For example it is necessary to create a scintillation hodoscope with N = 30 scintillators and multiplicity $t \leqslant 2$. What minimum number of multipliers is needed?Let us construct matrix $H_{30,11}$.

$H_{31,11} =$

	MPs
10101010101010101010101010101010	1
01010101010101010101010101010101	2
10010010010010010010010010010010	3
01001001001001001001001001001001	4
00100100100100100100100100100100	5
10000100001000010000100001000010	6
01000010000100001000010000100001	7
00100001000010000100001000010000	8
00010000100001000010000100001000	9
00001000010000100001000010000100	10
01000000100000010000001000000001	11

Scintillators ⟶ 1.....30

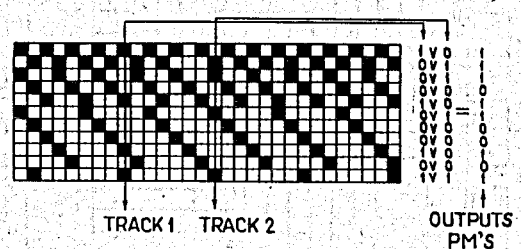


Fig. 19. Coding mask-detector for $t \leqslant 2$, $n = 30$ and $N = 11$.

Only 11 PMs need for multiplicity $t \leqslant 2$. Optical fibers can be used fo coding. The coding matrix can be created as mask also (Fig. 19).

Cluster counters. The coding matrix $H_{64,8}$ can be used for counting a singl cluster with dimensions $1 \leqslant b \leqslant 8$.

The matrix $H_{64,8}$ is composed of unitary matrices I_8. Fig. 20 gives a scheme of the counter. The *PROMs* programmed so that the syndrome

$$H_{64,8} = \left|\begin{array}{c} 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00000100 \\ 00000010 \\ 00000010 \\ 00000001 \end{array}\right| \left|\begin{array}{c} \\ \\ \\ \\ I \\ \\ \\ 8 \end{array}\right| \left|\begin{array}{c} \\ \\ \\ \\ \cdots\cdots\cdots \\ \\ \\ \end{array}\right| \left|\begin{array}{c} \\ \\ \\ \\ I \\ \\ \\ 8 \end{array}\right| \left|\begin{array}{c} N \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}\right|$$

$$n \longrightarrow \quad 1 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots 57 \cdots 64$$

weight $w = 1$, then $b = 1$ and so on. The efficiency of the coding scheme increases with encreasing number n. In general, for $b = z$ we must use a unitary matrix of the order z.

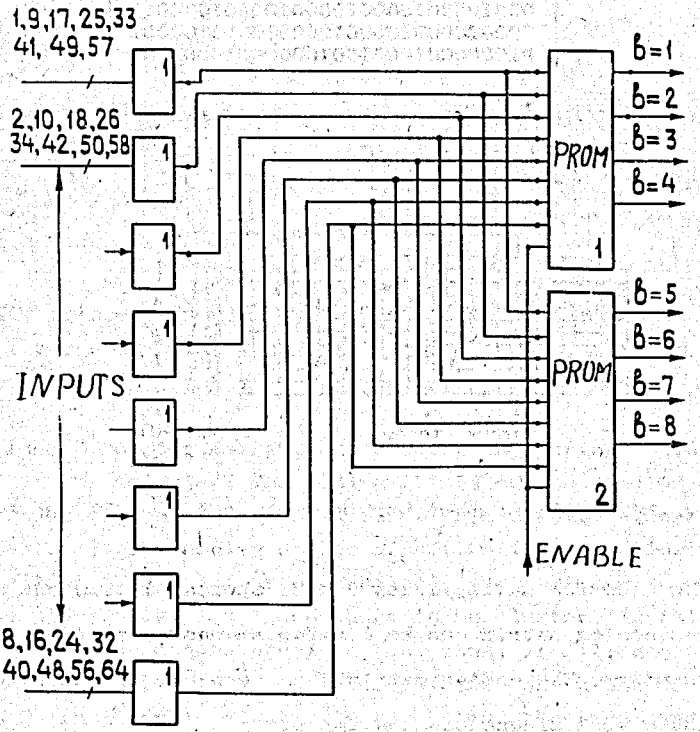


Fig. 20. Scheme of the cluster counter for n = 64, N = 8 and b = 8.

Superimposed iteration codes. The usual Hamming ($d = 3$) and the Gray ($3 \leqslant d \leqslant 4$) codes are shown to be superimposed ones [22]. That is way the itaration *OR*-Hamming, *OR*-Gray and Hamming-Hamming codes are interesting for the construction of coding schemes for scintillation hodoscopes [23], hodoscopic calorimeters and pixel detectors. For example let n be 225 pixels or scintillation counters which arrange the matrix having $k = 15$ rows and 15 columns. Let us calculate the syndrome for rows using the (15,4)-Gray code and for columns by the Boolean sum rules. As all the rows are coded equally, the coding scheme is identical for all 15 rows. The coding matrix $H_{15,4}$ represents the 4-bit Gray code [16]

$$H_{15,4} = \left|\begin{array}{c} 110011001100110 \\ 011110000111100 \\ 000111111110000 \\ 000000011111111 \end{array}\right| \left|\begin{array}{c} N \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\right|$$

$$n \longrightarrow \quad 1 \qquad\qquad 15$$

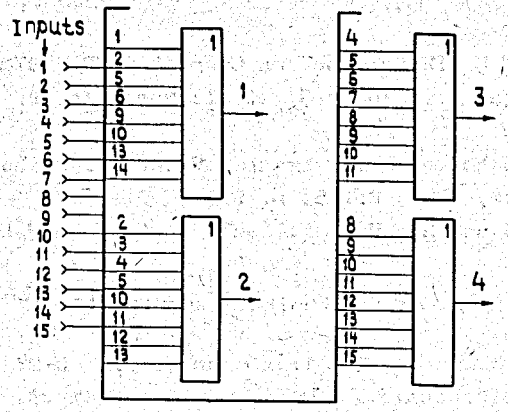


Fig. 21. Scheme for calculating the syndrome for one row. 1 – OR-gates, 1 – 4 – outputs.

Fig. 21 presents a principal scheme used to calculate the syndrome for one row. To calculate the syndrome for all the schemes, 15 *OR*-gates having 15 inputs and 4x15 *OR*-gates having 8 inputs are nee-

ded. It should be stressed that photomultipliers can be used as an *OR*- gate. The coding distance of such an iteration code is $d = 1.5 \times 3.5 = 7$. The determined multiplicity t is equal to 6. By analogy, more powerful superimposed iteration codes can be constructed.

Conclusion

Several methods of the construction of parallel counters used for fast signal processing registered in multichannel charged particle detectors are described. The author has suggested several types of the parallel counters which can be used to implementation of majority coincidence circuits having large number of inputs $n > 900$ and small delay. It is shown that for the construction of such devices preliminary data compression based on syndrome coding method should be used. Broad prospects of application of parallel compressors for obtaining the sum of a large number of signals registered in calorimeters should be also noted.

References

1. Swartzlander E.E. IEEE Trans. on Comput. C-22 (1973) 1021.
2. Brandt A., Dibon H., Flugge G. et al. Nucl. Instr. and Meth. 126 (1975) 519.
3. Guskov B.N., Maksimov A.I., Krastev V. et al. Pribory i Tekhnika Eksperimenta, 1984, No. 6, p. 91.
4. Imanishi A., Okuno H., Ukai K. Nucl. Instr. and Meth. A267 (1988) 168.
5. Nikityuk N.M. Pribory i Tekhnika Eksperimenta. 1991, No.1, 16-36.
6. Kobayashi H., Ohara H. IEEE Trans. on Comput. C-27 (1978) 753.7. Foster C.C., Stockton F.D. IEEE Trans. on Comput. C-20 (1971) 1580.
8. Riordan R.H.and Morton R.R.A. IEEE Trans. on Electr. Comput. EC-14 (1965) 153.
9. Ho. I.T., Chen T.C. IEEE Trans. on Comput. C-22 (1973) 762.
10. Gaiski D.D. IEEE Trans. on Comput. C-29 (1980) 393.

11. Schuller G.A. CERN, No. 82-09 (1982).
12. N.M. Nikityuk. JINR, P10-88-241 Dubna (1988).
13. Denby B. Computer Physics Communications. 49 (1988) 429.
14. Petrov E I. Problemy Peredachi Informatsii. IX (1973) p. 92.
15. Nikityuk N.M. In 6th Int. Conf. AAECC-6, Rome, Italy, July, 1988, Mora T. (Ed.), B. 357, Springer-Verlag, Berlin, 1989. (JINR, E10-88-28 (1988)) in English.
16. Nikityuk N.M. JINR, P10-87-266 Dubna (1987).
17. Nikityuk N.M. Byuleten Otkrytiy i Izobreteniy, No.8 (1990) 245. Avtorskoe swidetelstvo No. 1546992.
18. Peterson W.W. Error-Correcting Codes. New York, London 1961.
19. Nikityuk N.M. JINR, E10-91-161 (1991) Dubna (in English).
20. Nikityuk N.M., Rukoyatkin P., Swetov A.L. N. Pribori i Tekhnika Eksperimenta, 1991, No. 1 (1990) p.
21. Nikityuk N.M. JINR, E10-90-184 (1991) Dubna (in English). In Proc. Int. Conf. AAECC-8, Tokio, Aug.1990. Lecture Notes in Computer Science. 508 (1991)144. Ed. T. Sakata. Springer Verlag.
22. Nikityuk N.M. Pribory i Tekhnika Eksperimenta. No. 6 (1986) 77.
23. Gustaffson L., Hagberg E. Nucl. Instr. and Meth. A265 (1988) 521.

Received by Publishing Department
on December 26, 1991.