

# Synthesis and comparative analysis of multiple inputs parallel adders

Dimitar S. Tyanev, Nedyalko N. Nikolov, Stefka I. Popova

**Abstract:** This article consider/explore simultaneously addition of more than two integer numbers problem through (3:1) concentrators. Presented results are extension of previously published research of the problem, solved about horizontal organized addition. In this paper we discuss second possible organization, referred as vertical addition. Logical structure, which implements such organization, has been synthesized and analyzed. The analysis, as well as obtained quantitative estimations of machine costs and switching time, is presented. Conclusions based on the comparison of the two structures were made. Recommendations for design of solutions with different parameters have been formulated. Particular examples have been molded and implemented experimentally through Xilinx tools.

**Keywords:** parallel adders, concentrators, machine cost estimation, switching time estimation.

## 1) Problem formulation

Fast processing of big amounts of data is actual problem in most present applications, for example image processing systems. In these systems at the primary steps of data processing emerge the task of computing sums as:

$$y = \sum_{i=1}^r x_i, \quad (1)$$

where  $x_i$ ,  $i = \overline{1, r}$  are  $n$ -bits integer numbers. Requested high speed of the computing could be obtained by machinery implemented logical structures with high degree of parallelism [3÷14]. There are two possible computing organizations at these conditions, which differ at initial order of the operands and are named “horizontal” and “vertical” respectively. Synthesis of the horizontal structure, as well as the results from its exploration has been published in [1]. This paper presents the synthesis of the vertical structure.

### Vertical organized sum computation

Binary arithmetic sum (1) could be presented in polynomial form:

$$y = \sum_{i=1}^r (P_x)_i = \sum_{i=1}^r \left( (b_{n-1}^{(i)}) \cdot 2^{n-1} + (b_{n-2}^{(i)}) \cdot 2^{n-2} + \dots + (b_j^{(i)}) \cdot 2^j + \dots + (b_0^{(i)}) \cdot 2^0 \right), \quad (2)$$

where  $b_j^{(i)}$  stands for bit  $j$  of the addend  $x_i$  with number  $i$ .

Expression (2) could be presented as:

$$y = \sum_{j=0}^{n-1} \left( 2^j \cdot \sum_{i=1}^r b_j^{(i)} \right). \quad (3)$$

The description (3) of the task shows that it could be decomposed in two parts – computing of the bit sums and computing of the weight bit sums. Bit sums will be represented as:

$$BS_j = \sum_{i=1}^r b_j^{(i)}, \quad j = \overline{0, n-1}. \quad (4)$$

Then the final sum will be:

$$y = \sum_{j=0}^{n-1} 2^j \cdot BS_j. \quad (5)$$

The analysis and formulated two problems are significant about the logical synthesis. At the time of the logical synthesis two different structures have been created, utilizing the natural parallelism of the tasks.

## 2) Bit computing logical structure

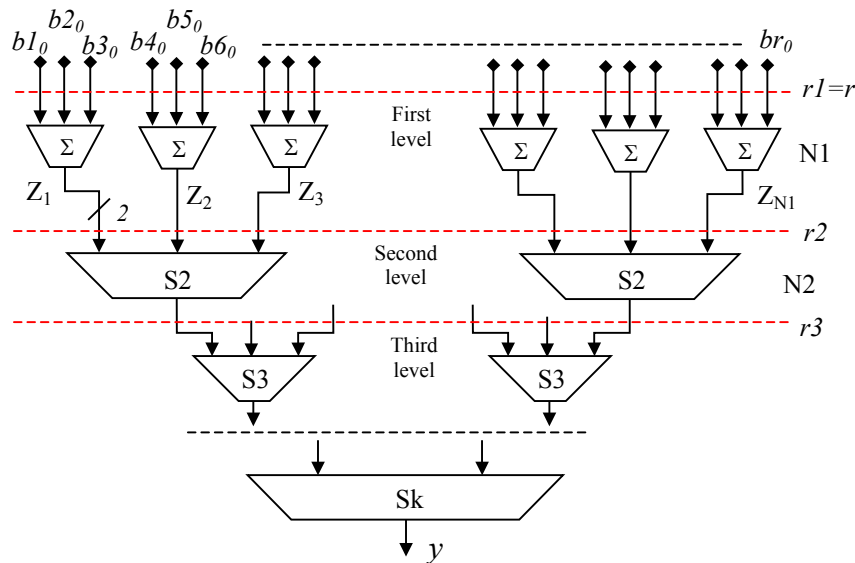
All  $n$  bit sums  $BS_j$  have to be computed at the same time, so  $n$  adding schemes are needed. These are schemes for addition of  $n$  1-bit numbers. Generally this problem is analogical to the explored in [1]. The difference is that the entry appends are 1-bit numbers. The organization of such addition is named vertical.

Length of the bit sum is defined as the  $n$ -multiple binary sum of 1-bit appends on the basis of the proven theory in [1]. Obtained decision of the current problem in number of bits is:

$$L_{BS} = \lfloor \log_2(r) \rfloor + 1. \quad (6)$$

The value defined by (6) is maximal possible and sufficient length of the binary bit sum (4).

Synthesis of the logical structure for 1-bit numbers parallel addition through (3:1) concentrators [7] submits to considerations, discussed in [1]. Adding scheme has pyramidal structure (fig.1), which have at first level complete 1-bit binary adders because of the entry appends. If they have 1-bit length, the results extend its length on each level to the final, defined by (6).



Фиг. 1 Bit sum computing structure

Generally the relevance between the number of the levels  $k_v$  and the number of entry appends  $r$  is:

$$k_v = \lfloor \log_3 r \rfloor. \quad (7)$$

Adding scheme is estimated in two measures: machine outlay of the implementation and switching time.

### Machine outlay estimation

As it seen at fig.1, there are 1-bit adders  $\Sigma$ , at the first level, which number  $N_1$  is defined by the current arranging:

$$N_1 = \left\lfloor \frac{r_1}{3} \right\rfloor. \quad (8)$$

There are two possible values different than zero (1 and 2) for the residual  $Rem(r_1) = (r_1)_{\text{mod } 3}$  of such arranging. Because appends have length of 1-bit, the two cases could be generalized in one and their sum is obtained by one half-adder. It will be missing if the division (8) is exact. So from the first level  $r_2$  2-bit sums  $Z=(p, z)$  are obtained. Generally the estimation of machine outlay could be defined as:

$$Q_1 = \begin{cases} N_1, & \text{if } Rem(r_1) = 0; \\ N_1 + 0,5, & \text{if } Rem(r_1) \neq 0. \end{cases} \quad (9)$$

Next levels from the pyramidal scheme implement parallel-consecutive addition of obtained 2-bit numbers through (3:1) concentrators. Maximal possible 2-bit sum is 3 ( $3=11_{(2)}$ ).

Because the number of appends at the second level is equal to the number of the adders at the first level, the count of obtained 2-bit numbers is:

$$r_2 = \begin{cases} N_1, & \text{if } \text{Rem}(r_1) = 0; \\ N_1 + 1, & \text{if } \text{Rem}(r_1) \neq 0. \end{cases} \quad (10)$$

Received from the first level sums are grouped in triads based on [1] so the count of concentrators  $S_2$  necessary for the second level could be defined as:

$$N_2 = \left\lfloor \frac{r_2}{3} \right\rfloor. \quad (11)$$

The residual from grouping in triads  $\text{Rem}(r_2)$  is utilized at the current level only if full adder could be used, i.e. there is two appends left. In case of one append left it will be carried to the next higher level. Maximal possible sum from three 2-bit numbers is 9 ( $1001_{(2)}$ ), i.e. 4-bit number. If the current level is marked as  $p$ , the length of obtained at this level sums  $L_p$ , can be computed as:

$$L_p = \left\lfloor \log_2(S_{\max}^{(p)}) \right\rfloor = \left\lfloor \log_2(3^p) \right\rfloor, \quad p = \overline{1, k_v}. \quad (12)$$

Received estimation for the sum length allows estimating machine outlay for the concentrators at the same level as follows:

$$q_p = 2.L_{p-1} - 1, \quad p = \overline{1, k_v}. \quad (13)$$

Hence the estimation for the machine outlay  $Q_2$  about implementation at second level in number of 1-bit binary adders generally can be defined as:

$$Q_2 = \begin{cases} q_2.N_2 + (L_1 - 1), & \text{if } \text{Rem}(r_2) = 2; \\ q_2.N_2, & \text{if } \text{Rem}(r_2) \neq 2. \end{cases} \quad (15)$$

The number of obtained from the second level sums is marked as  $r_3$ . The exact number is defined as follows:

$$r_3 = \begin{cases} N_2, & \text{if } \text{Rem}(r_2) = 0; \\ N_2 + 1, & \text{if } \text{Rem}(r_2) \neq 0. \end{cases} \quad (14)$$

Maximal length of the binary sums at the output of the second level is  $L_2=4$  bits. These numbers are grouped in triads and form the third level concentrators  $S_3$ . Analogically, their count is:

$$N_3 = \left\lfloor \frac{r_3}{3} \right\rfloor. \quad (15)$$

Length of sums  $L_3$ , which will be received at this level, is defined in accordance with (12) and the outlay  $q_3$  for a concentrator with such length is defined in accordance with (13).

The machine outlay for whole third level implementation is estimated as:

$$Q_3 = \begin{cases} q_3.N_3 + (L_2 - 1), & \text{if } \text{Rem}(r_3) = 2; \\ q_3.N_3, & \text{if } \text{Rem}(r_3) \neq 2. \end{cases} \quad (16)$$

For synthesis of arbitrary level in accordance with the estimations made above, there are following generalizations:

1. About the length of single sum obtained at the current level №  $p$ :

$$L_p = \left\lfloor \log_2(3^p) \right\rfloor. \quad (17)$$

2. About machine outlay estimation for single concentrator at the current level:

$$q_p = \begin{cases} 1, & p = 1; \\ 2.L_{p-1} - 1, & p = \overline{2, k_v}. \end{cases} \quad (18)$$

3. About the number of necessary at the current level №  $p$  concentrators:

$$N_p = \left\lfloor \frac{r_p}{3} \right\rfloor. \quad (19)$$

4. Where the total number of addends  $r_p$  is defined as follows:

$$r_p = \begin{cases} N_{p-1}, & \text{if } \text{Rem}(r_{p-1}) = 0; \\ N_{p-1} + 1, & \text{if } \text{Rem}(r_{p-1}) \neq 0. \end{cases} \quad (20)$$

5. About machine outlay for current level implementation №  $p$ :

$$\begin{aligned} & \text{if } (r_p > 3) \text{ then} \\ & \quad \text{if } (p = 1) \text{ then } Q_1 = \begin{cases} N_1, & \text{if } \text{Rem}(r_1) = 0; \\ N_1 + 0,5, & \text{if } \text{Rem}(r_1) \neq 0. \end{cases} \\ & \quad \text{else } Q_p = \begin{cases} q_p \cdot N_p + (L_{p-1} - 1), & \text{if } \text{Rem}(r_p) = 2; \\ q_p \cdot N_p, & \text{if } \text{Rem}(r_p) \neq 2. \end{cases} \\ & \quad \text{end if} \\ & \text{else } Q_p = \begin{cases} q_p, & \text{if } r_p = 3; \\ L_{p-1} - 1, & \text{if } r_p < 3. \end{cases} \\ & \text{end if} \end{aligned} \quad (21)$$

6. About continuation condition:

$$\begin{aligned} & \text{if } (r_p > 3) \text{ then go to point } l; \\ & \quad \text{else end.} \\ & \text{end if} \end{aligned} \quad (22)$$

In the event, total machine outlay for implementation of the structure from fig.1 is estimated with the following sum:

$$Q_{BS} = Q_1 + Q_2 + \dots + Q_k, \quad (23)$$

and the outlay for parallel receipt/obtaining of all  $n$  bit sums (4) is estimated as:

$$Q = n \cdot Q_{BS}. \quad (24)$$

#### *Switching time estimation*

Because of parallel working schemes computing the bit sums the necessary time is defined from the switching time of the logical structure (fig.1). This switching time is estimated with conditions, accepted in [1] and considering that the entry appends are 1-bit numbers, the total switching time estimation of the structure is:

$$t_{BS} = (k_v + l) \cdot \tau. \quad (25)$$

### 3) Logical structure for sum of bit sums computing

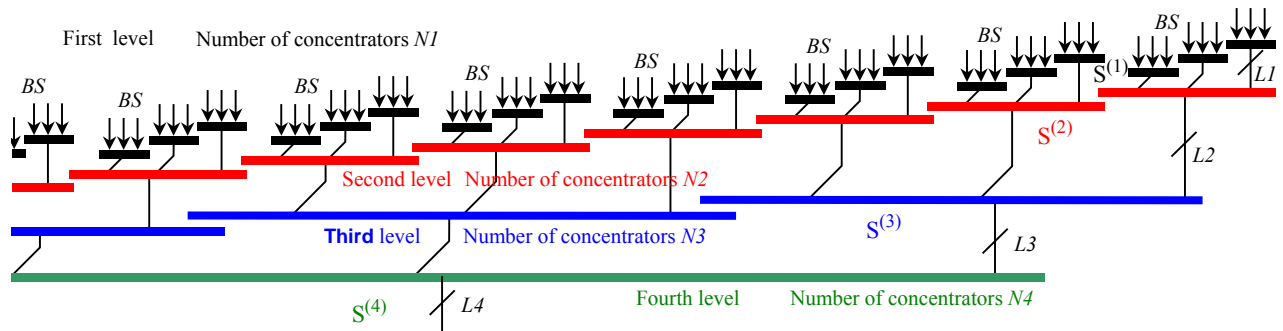
This sum is presented by (5) and because of grouping in triads it could be convert into:

$$\begin{aligned} y = & \{ \dots \{ \dots \{ [(BS_1 + 2 \cdot BS_2 + 4 \cdot BS_3) \cdot 2^0 + (BS_4 + 2 \cdot BS_5 + 4 \cdot BS_6) \cdot 2^3 + \\ & \quad + (BS_7 + 2 \cdot BS_8 + 4 \cdot BS_9) \cdot 2^6] + \\ & + [(BS_{11} + 2 \cdot BS_{12} + 4 \cdot BS_{13}) \cdot 2^0 + (BS_{14} + 2 \cdot BS_{15} + 4 \cdot BS_{16}) \cdot 2^3 + \\ & \quad + (BS_{17} + 2 \cdot BS_{18} + 4 \cdot BS_{19}) \cdot 2^6] \cdot 2^9 + \\ & + [(BS_{20} + 2 \cdot BS_{21} + 4 \cdot BS_{22}) \cdot 2^0 + (BS_{23} + 2 \cdot BS_{24} + 4 \cdot BS_{25}) \cdot 2^3 + \\ & \quad + (BS_{26} + 2 \cdot BS_{27} + 4 \cdot BS_{28}) \cdot 2^6] \cdot 2^{18} \} + \\ & + \{ \dots \dots \dots \} \cdot 2^{27} + \{ \dots \dots \dots \} \cdot 2^{54} \} + \{ \dots \dots \dots \} \cdot 2^{81} + \dots \end{aligned} \quad (26)$$

As it seen, the sum  $y$  from consecutive shifted bit sums BS could be accumulated at the time of parallel-consecutive addition with shifting and (3:1) concentrators applied, showed at fig.2.

It is necessary to explain that fig.2 shows only junior part of the scheme. Because of mutual relative left shifting of the addends, the methodology for creating used concentrators is analogical to the described in [2]. Parallel-consecutive structure has to be creating from concentrators with different conditions reporting – inconstant/variable step of the relative shifting and, at the other hand, variable and shift-depending extension of the sum length. For example, the concentrators from the first level collect three bit sum BS, which are shifted in a bit one toward another. Concentrators from the second level collect on triads sums  $S^{(l)}$ , obtained from the first level, but the

relative shifting is three bits, i.e. with relative positional weights  $2^0$ ,  $2^3$ , and  $2^6$  (look (26)). Concentrators at the third level collect on triads sums  $S^{(2)}$ , obtained from the second level, but with mutual relative shifting of 9 bits ( $2^0$ ,  $2^9$ ,  $2^{18}$ ). So the relative shifting at the inputs of the corresponding concentrators increase consecutively in geometrical progression: 1, 3, 9, 27, ...



Фиг. 2 Scheme of horizontal parallel bit sums addition

On other side, the length of intermediate results increases to the output too. Because all the concentrators collect three numbers, their left extension depends from the relative shifting of the entry appends. Hence all the mentioned considerations have to be in mind about the estimation of the logical structure at fig.2.

The level's number in the pyramidal structure at fig.2 is defined from the number of the bit sums, i.e. from the length  $n$  of appends as well as from the grouping in triads methodology, the grouping residual ( $Rem_k=0,1,2$ ), is transferred to the next higher level respectively. The machine outlay estimation differs and complicates comparing to the previously examined cases because of individual at every level relative shifting of appends as well as of their varying length. Thus to define the length of the concentrators at every level became autonomous problem.

At the first level each concentrator collects three bit sums, shifted at 1 bit one to another. The length of binary sums received at first level could be defined as:

$$L_{S1} = L_{BS} + 3. \quad (27)$$

The machine outlay estimation for single first level concentrator in terms of 1-bit binary adders' number is:

$$Q_{S1} = 2.(L_{BS} - 1). \quad (28)$$

Total machine outlays for first level implementation are:

$$Q1 = N1.Q_{S1}, \quad (29)$$

where total number concentrators  $N1$  is defined from the grouping in triads methodology.

At the second level each concentrator collects three sums  $S^{(1)}$ , shifted on 3 bits one to another. The length of received at second level sums  $S^{(2)}$  is:

$$L_{S2} = L_{S1} + 2.3 + 1. \quad (30)$$

The machine outlays  $Q_{S2}$  for concentrators implementing sums  $S^{(2)}$ , are defined analogically. Finally the estimation can be presented as:

$$Q_{S2} = 2.(L_{S1} - 2.3) + 2.3 + \frac{3-1}{2}. \quad (31)$$

Total machine outlays for implementing of the second level are:

$$Q2 = N2.Q_{S2}. \quad (32)$$

At the third level each concentrator collects three sums  $S^{(2)}$ , shifted at 9 bits one to another. The length of received at third level sums  $S^{(3)}$  is:

$$L_{S3} = L_{S2} + 2.9 + 1. \quad (33)$$

The machine outlays  $Q_{S3}$  for concentrators implementing sums  $S^{(3)}$  are defined analogically. Finally the estimation can be presented as:

$$Q_{S3} = 2.(L_{S2} - 2.9) + 2.9 + \frac{9-1}{2}. \quad (34)$$

Total machine outlays for implementing of the third level are:

$$Q_3 = N_3 \cdot Q_{S_3} . \quad (35)$$

Presented analysis is sufficient to generalize inductively the estimations about the length of binary sums, necessary machine outlays for concentrators' implementation, as well as for total machine outlays at arbitrary level. However it is necessary to explicate that in accordance with the grouping methodology the estimations (29), (32) and (35) are valid only in case the grouping residual is zero. If the residual is one ( $Rem_k=1$ ), the number left ungrouped is transferred to the next level. If the residual of the entry appends grouping is  $Rem_k=2$ , there is extra adder at the current level to collect these two numbers with the mutual shifting in consideration. The machine outlay for the extra adder has to be added to the total outlays for the current level. Generally if the residual is different from zero, the number of the entry appends increases with one.

The machine outlays for extra adder depend from the length of mutual shifting and the length of the addends. General formal estimation of the outlays for this adder is defined with consideration that there is lack of real addition in the junior  $m_k$  bits and in the senior  $m_k$  bits is possible only carry propagation. There is real addition only in the middle bits, namely from  $N_2(m_k)$  bit to  $N_2(L_{S_{k-1}}-m_k)$  bit. Finally the machine outlay estimation for the extra adder could be presented as:

$$Q_{\Sigma_k} = (L_{S_{k-1}} - 2 \cdot m_k) + \frac{m_k}{2} . \quad (36)$$

So, obtained total estimations are:

1. About the length of appends' mutual shifting for the next level as function of level's number  $k$ :

$$m_k = 3^{k-1} . \quad (37)$$

2. About the length of obtained sums as function of level's number  $k$ :

$$L_{S_k} = (L_{BS} + 3) + (2 \cdot 3^1 + 1) + (2 \cdot 3^2 + 1) + (2 \cdot 3^3 + 1) + \dots + (2 \cdot 3^{k-1} + 1) . \quad (38)$$

Last expression could be simplified to:

$$L_{S_k} = L_{BS} - 1 + k + 3^k . \quad (39)$$

3. About concentrator implementation machine outlays as function of level's number  $k$ :

$$Q_{S_k} = 2 \cdot L_{S_{k-1}} - \frac{3^k + 1}{2} . \quad (40)$$

4. About the machine outlays at level  $k$ :

$$Q_k = \begin{cases} Q_{S_k} \cdot N_k + (L_{S_{k-1}} - 2 \cdot m_k) + \frac{m_k}{2}, & \text{if } Rem_k = 2 ; \\ Q_{S_k} \cdot N_k , & \text{if } Rem_k \neq 2 . \end{cases} \quad (41)$$

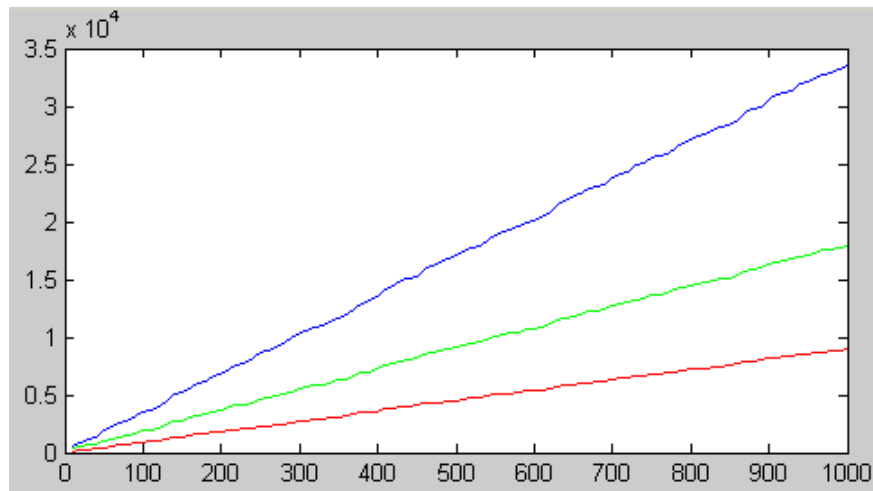
where total number of concentrators  $N_k$  is defined by grouping in triads methodology, represented from (19) and (20). The condition for ending iteration process of synthesizing the logical structure is analogical to (22).

5. About the total machine outlays:

$$Q = Q_1 + Q_2 + \dots + Q_k . \quad (42)$$

6. Sum of (24) and (42) estimations defines total machine outlays for the considered version of vertical organization.

Graphical expression of (26) and (44) estimations' sum is showed at fig.6 as function of addends' count  $r$ , if there are 8-bit numbers ( — ); 16 bit numbers ( — ) and 30 bit numbers ( — ).



Фиг. 6 Machine outlays  $Q$  as function of numbers count (in 10 to 1000 interval)

#### Switching time estimation

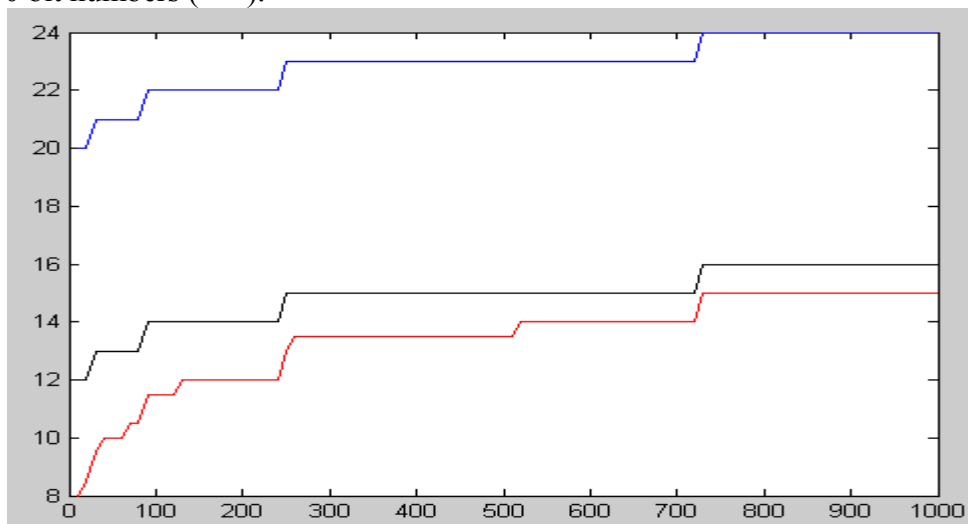
The estimation of the synthesized structure's switching time is obtained with consideration that it consists of two consecutive parts –  $n$  parallel working structures, computing bit sums followed by “horizontal” structure for their parallel addition. It results in next expression:

$$t_{\Sigma} = t_v + t_h, \quad (42)$$

where the switching time estimations for the “vertical” and for the “horizontal” structures are respectively as follows:

$$t_v = t_{BS}, \quad t_h = \left(\frac{L_{BS}}{2} + k_h\right) \cdot \tau. \quad (43)$$

As it seen, the length of bit sums  $L_{BS}$  is taken in two, considering appends' mutual shifting at the input of concentrators in the second structure. Graphical expression of the estimation (45) is showed at fig.7 as function of appends' number  $r$ , if there are 8-bit numbers (—); 16 bit numbers (—) and 30 bit numbers (—).



Фиг. 7 Switching time  $t_x$  as function of number count (in 10 to 1000 interval)

#### 4) Conclusions

Defined problem (1) is exhausted by explored in this paper “vertical” version of parallel addition of many numbers organization in conjunction with published in [1] “horizontal” version. Obtained results for both organizations are compared and the conclusions are as follows:

1. The machine outlays for synthesized in this paper structure have linear increment with appends' number increasing (fig.7).
2. Main part of the machine outlays (more than 95%) for the vertical organized structure fall upon the sub-structure computing bit sums.

3. Computing bit sums sub-structure is characterized by high level of homogeneity because is made of  $n$  parallel working identical schemes. In these terms, generally the implementation of vertical structure could be adapted faster if there are problem's parameters change.

4. Machine outlays for implementation of both alternative organizations (horizontal and vertical) are equivalent with identical parameters of problem (1).

5. The comparison of the two structures with different problem's parameters is discussed. In other words, which structure is preferred in case of plenty of numbers with short length or few numbers with large length? The approximately parity of the final sums modules is assumed as identity condition for obtained estimations comparison. For example, if  $n=8$  and  $r=643$ , maximal possible sum is  $X_{\max} = 643.255 = 163965$ . Approximately same result ( $X_{\max}=10.16383= 163830$ ) is obtained with parameters  $n=14$  and  $r=10$ . The answer: because the machine outlays for vertical and horizontal organizations in both cases are identical, it is impossible to lie down preference based on this criterion.

6. Argument interval  $r$ , showed at fig.7, allows exponential tendency in switching time increasing to be noted.

7. The comparison of structures' switching times in conditions of example from point 5 allows next conclusion to be made: with identical parameters of problem (1) vertical structure's switching time is identical to horizontal's structure switching time if parameters are in correlation  $r \gg n$ . Otherwise the vertical structure's switching time is twice better than the horizontal structure's.

### Literature:

- [1]. Тянев Д.С., Попова С.И., Иванов А.И., Янев Д.В., *Синтез и сравнителен анализ на паралелни многовходови суматори*, Списание "Компютърни науки и технологии", №2/2005, година III, ISSN 1312-3335, стр. 52-61.
- [2]. Тянев Д.С., Попова С.И., Янев Д.В., Иванов А.И., *Конвейерен умножител с концентратори*, Списание "Компютърни науки и технологии", №1,2/2006, година IV, ISSN 1312-3335, стр. 23-28.
- [3]. ASIC Design for Signal Processing - Carry Save Arithmetic: <http://www.geoffknagge.com/fyp/carrysave.shtml> .
- [4]. Loh, *Carry Save Addition, Processor Design*, Spring 2005, [http://www3.cc.gatech.edu/classes/AY2005/cs3220\\_spring/csa-notes.pdf](http://www3.cc.gatech.edu/classes/AY2005/cs3220_spring/csa-notes.pdf).
- [5]. *Digital Computer Arithmetic Datapath Design Using Verilog Hdl*, James E. Stine, 2004. [http://books.google.com/books?vid=ISBN1402077106&id=Jc3cz5dvIxYC&pg=RA1-PA60&lpg=RA1-PA60&ots=K-meqDviLN&dq=design+carry+save+adder&sig=1fINOk4NXsgKFZIOIHZAcmepPPIc#PR\\_A1-PA179,M1](http://books.google.com/books?vid=ISBN1402077106&id=Jc3cz5dvIxYC&pg=RA1-PA60&lpg=RA1-PA60&ots=K-meqDviLN&dq=design+carry+save+adder&sig=1fINOk4NXsgKFZIOIHZAcmepPPIc#PR_A1-PA179,M1)
- [6] Behrooz Parhami, *"Computer Arithmetic" - part Carry Save Arithmetic*, Oxford Press, 2000, pp.131.
- [7]. Тянев Д.С., *Организация на компютъра (цифрова аритметика)*, ISBN 954-20-0258-0, ТУ - Варна, 2007 год.
- [8]. Карцев М. А., *Арифметика цифровых машин*, Издательство "Наука", Москва, 1969.
- [9]. Карцев М. А., Брик В. А., *Вычислительные системы и синхронная арифметика*, Издательство "Радио и связь", 1981.
- [10]. *Специализированные процессоры для высокопроизводительной обработки данных*, Новосибирск, Издательство "Наука", 1998.
- [11]. Фет И., *Специализированные однородные структуры. Цифровые компрессоры*, АН Институт Математики, Препринт №27, 1988.
- [12]. Kiefer G., Waker A., Thumm A., *Rechnerorganisation*, Institut fur Informatik, Stuttgart, 2001.
- [13]. Schiller, Jochen, *Rechnerstrukturen*, Freie Universitat, Berlin, 2003.
- [14]. T. Kim, W. Jao, and S. Tjiang, *"Circuit Optimization using Carry-Save-Adder Cells"*, IEEE Trans. on CAD, Vol.17, No.10,1998.