ISSN: 2994-6433

Research Article

Journal of Sensor Networks and Data Communications

Program Operators SIT, tS, S^{1}e, Set$_1$

Illia Danilishyn*, Oleksandr Danilishyn

Sumy State University, Vulytsya Mykoly Sumtsova, 2, Sumy, Sumy Oblast, 40000, Ukraine

***Corresponding author:**
Illia Danilishyn, Sumy State University, Vulytsya Mykoly Sumtsova, 2, Sumy, Sumy Oblast, 40000, Ukraine.

Submitted: 2023, Sep 17; **Accepted**:2023, Oct 21 : **Published**: 2023, Oct 26

Citation: Danilishyn, I., Danilishyn, O. (2023). Program Operators SIT, tS, S1e, Set1. *J Sen Net Data Comm, 3*(1), 138-143.

Abstract
The purpose of the article is to create new program operators for a fundamentally new type of neural network with parallel computing, and not with the usual parallel computing through sequential computing.

Keywords: Sit-program operators, tS-program operators, S^{1}e - program operators, Set$_1$- program operators.

1. Introduction

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through Sit-Networks [1] in one of the central departments of which a conventional computer system is located. The parallel processor is itself eprogram [1] with direct parallel computing not through serial computing.

Using conventional coding by a computer system, through a Target-block with a Sit -program operator $St_{activation}^{\{A\}f}$, it will be possible to obtain the execution of a parallel action A with the desired target weight f. Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For Sit-coding and Sit-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel program of operator $St_{activation}^{\{\text{UHF AC} := Q\}}$, we simultaneously enter the desired set of codes Q using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

2. Consider the Types of Direct Parallel Program Operators

- Sit-program operators
- tS-program operators
- S^{1}e - program operators
- Set$_1$- program operators

2.1 Here are Some of the Sit-Program Operators

- Simultaneous assignment of the expressions $\{p\}=(p_1,p_2,\dots,p_n)$ to the variables $\{a\}=(a_1,a_2,\dots,a_n)$. Implemented through

$$St_{\mathrm{x}}^{\{\{a\}:=\{p\}\}}, \text{ or } St_{\{p\}}^{\{a\}:=}.$$

- Simultaneous check the set of conditions $\{f\} = (f_1,f_2, \dots,f_n)$ for a set of expressions $\{B\} = (B_1,B_2, \dots,B_n)$. It is implemented through $St_{\mathrm{x}}^{IF\{\{B\}\{f\}\}\ then\ Q}$ where Q can be any.
- Similarly for loop operators and others.

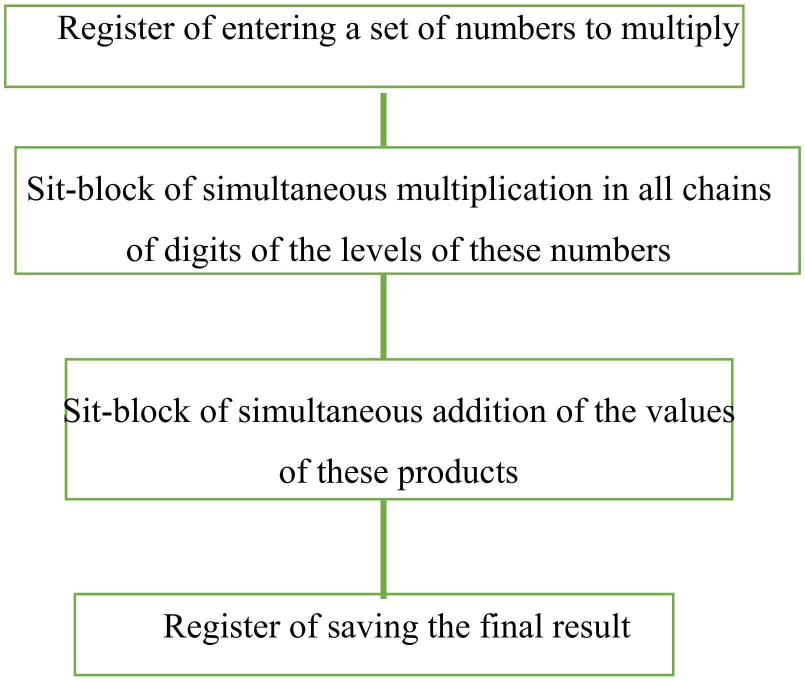
2.2 Sit-Algorithm Examples

- Simultaneous multiplication: $\mathrm{St}_{\mathrm{x}}^{\{a*\}}$: enter the notation of the set B with elements $b_{i_1 i_2 \dots i_n} = (St_x^{\{a_{1_{i_1}}*, a_{2_{i_2}}*, \dots, a_{n_{i_n}}\}})_R$ for any $\{i_1,i_2,\dots,i_n\}$ without repetitions, x= $St_a^{\{multiplication\ table\}}$,

R= $St_a^{\{i_1+,i_2+,\dots,i_n\}}$, R is the index of the lower discharge (we choose an index on the scale of discharges):

index	**discharge**
n	n
…	…
1	1
,	0
-1	1st digit to the right of the point
-2	2nd digit to the right of the point
...	...

Then $St_x^{\{B+\}}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. Here, in fact, sets of digits in the corresponding digits, representing numbers, are multiplied together simultaneously. The simplest functional scheme of the assumed arithmetic-logical device for Sit-multiplication:


Register of entering a set of numbers to multiply
Sit-block of simultaneous multiplication in all chains of digits of the levels of these numbers
Sit-block of simultaneous addition of the values of these products
Register of saving the final result

One example is pattern recognition: $St_B^{if\, St_B^{image\ archive} \ni St_B^{q}\ then\ Name\ of\ q}$.

The example of Sit-program is $St_x^{\{St_x^{\{\{a(x)\}:\{p\}\}}, St_x^{IF\{\{B\}\{f\}\}\ then\ Q}, St_Q^{Q}\}}$.

Consider a third type of capacity in itself [1]. For example, based on $St_x^{\{a\}}$, where $\{a\}=(a_1, a_2, ..., a_n)$, i.e. n - elements at one point, we can consider the capacity $S_3 f$ in itself with m elements from {a}, m<n, which is formed according to the form:

$$w_{mn}=(m,(n,1)) \quad (1)$$

that is, the structure $St_x^{\{a\}}$ contains only m elements.

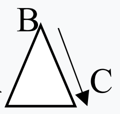
Capacities in themselves of the third type can be formed for any other structure, not necessarily Sit, only by necessarily reducing the number of elements in the structure, in particular, using form

$$w_{m_1 \cdots m_n} = (m_1, (m_2, (\dots (m_n, 1) \dots))) \quad (2)$$

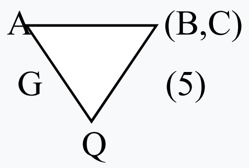
Structures more complex than $S_3 f$ can be introduced. For example, through a forms that generalizes (1):

$$w_{ABC} = (A, (B, C)) \quad (3)$$

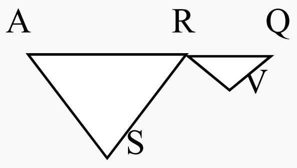
$$w_{ABC} = A$$



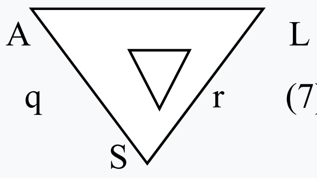
(4)



(5)



(6)



(7)

where A is compressed (fits) in C in the compression structure B in C (i.e. in the structure St_C^B); or through the more general forms that generalizes (2):

$$w_{A_1A_2\ldots A_nC} = (A_1,(A_2,(\ldots(A_n,C)\ldots))) \quad (8)$$

and corresponding generalizations of (8) on (4)-(7), etc.

(3), (8) are represented through the usual 2-bond. Science is the discipline of 2-connections, since everything in science is carried out through 2-connected logic, quantum logic is also a projection of 3-connected logic onto 2-connected logic. (4)-(7) schematically interpret the formation of capacity in itself through a pseudo 3-connected form with a 2-connected form. The ideology of Sit and S_3f can be used for programming.

$S_3 f$ – software operators [1] will differ only just because aggregates *{a},{p},{B},{f}* will be formed from corresponding Sit-program operators in form (1) for more complex operators in forms (2)-(4).

For example, $St_{g\{R\}}^{\{R\}}$ is the capacity in itself of the second type if *g{R}* is a program capable of generating $\{R\}$.

The example of self-program of the first type is $St_{\{St_{\mathrm{x}}^{\{\{a(x)\}:\{p\}\}},St_{\mathrm{x}}^{IF\{\{B\}\{f\}\}\ then\ Q},St_Q^Q\}}^{\{St_{\mathrm{x}}^{\{\{a(x)\}:\{p\}\}},St_{\mathrm{x}}^{IF\{\{B\}\{f\}\}\ then\ Q},St_Q^Q\}}$.

The example of Sit-program for Smnst [1]:
$St_B^{q:=}$ - assigning q to B.
$St_f^{tw:=}$ - assigning target weight tw to f .
$St_{activation}^{\{q\}f}$ - Smnst $activation$ for $\{q\}f$.

2.3 Sit-Coding

Sit-coding: 1) set A to set B, 2) set A to a point q, where the elements of the sets A, B can be continuous. For example, St_B^A.
There are Sit-coding, Sit-translation, Sit-realize of eprograms and of the programs from the archives without extraction theirs

2.4 Self-Coding

Self-coding: 1) set A to set A, i.e. A on itself 2) set A to a point q in form (1), where the elements of the sets A, B can be continuous. For example, St_A^A.

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with Sit-Networks [1], it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through Sit-Networks and series-parallel. Codes from a conventional type computer system will be used via Sit -connectors in Sit - coding, for example: $St_{activation}^{\{\text{UHF AC}:=Q\}}$. UHF AC field activation is used.

2.5 Dynamic Sit and $S_3f(t)$ Programming

The ideology of dynamic Sit and $S_3f(t)$ can be used for programming:

- The process of simultaneous assignment of the expressions $\{p(t)\}=(p_1(t),p_2(t),...,p_n(t))$ to the variables $\{a(t)\}=(a_1(t),a_2(t),...,a_n(t))$ is implemented through $St(t)_x^{\{\{a(t)\}:\{p(t)\}\}}$
- The process of simultaneous check the set of conditions $\{f(t)\} = (f(t)_1, f_2, (t) ...,f(t)_n)$ for a set of expressions $\{B(t)\} = (B_1(t),B_2(t),...,B_n(t))$ is implemented through $St(t)_x^{IF\{\{B(t)\}\{f(t)\}\}\ then\ Q(t)}$ where Q(t) can be any.
- Similarly for loop operators and others.
- $S_3 f(t)$– software operators will differ only in that the aggregates $\{a(t)\},\{p(t)\},\{B(t)\},\{f(t)\}$ will be formed from corresponding processes Sit(t) for the above mentioned programming operators through form (1) or forms (2)-(4) for more complex operators.

3. tS-Program Operators

The ideology of tS and t_{S_4f} [2] can be used for programming. Here are some of the tS-program operators.

- Simultaneous expelling assignment of the expressions $\{p\}=(p_1, p_2,...,p_n)$ from the variables $A = (a_1,a_2,...,a_n)$. It's implemented through ${}_{\{=:\{p\}\}}^{\{a\}}St$.
- Simultaneous expelling checks the set of conditions $\{f\} = (f_1, f_2, ..., f_n)$ for a set of expressions $\{B\} = (B_1, B_2, ..., B_n)$. It's implemented through ${}_{F\{\{B\}\{f\}\}\ then\ Q}^{x}St$ where Q can be any.

- Similarly for loop operators and others.

t_{S_4f} –software operators will differ only just because aggregates $\{a\},\{p\},\{B\},\{f\}$ will be formed from corresponding tS program operators in form (1) for more complex operators in forms (2)-(4).

Consider hierarchical tS-program operator

$${}_A^BSt = \left\{\begin{matrix} D + {}_{A-A\cap B}^{\{\}}St \\ (B - A\cap B) - (A - A\cap B) \end{matrix}\right\}, \text{where D is oself-set for } (A\cap B - Co(A\cap B)).$$

3.1 Dynamic tS and $t(q)_{S_4f}$ Programming at Time q

The ideology of tS and t_{S_4f} can be used for dynamic programming [2]. Here are some of the tS- dynamic programming operators.

- The process of simultaneous expelling assignment of the expressions $\{p(t)\}=(p_1(t),p_2(t),...,p_n(t))$ from the variables $a(t)=(a_1(t),a_2(t),...,a_n(t))$ is implemented through ${}_{\{=:\{p(t)\}\}}^{\{a(t)\}}St(t)$.

- The process of simultaneous expelling check the set of conditions $\{f(t)\} = (f(t)_1, f_2,(t) ..., f(t)_n)$ for a set of expressions $\{B(t)\} = (B_1(t),B_2(t),..., B_n(t))$ is implemented through ${}_{IF\{\{B(t)\}\{f(t)\}\}\ then\ Q(t)}^{x(t)}St(t)$, where Q(t) can be any.
- Similarly for loop operators and others.

t_{S_4f} – software operators will differ only just because aggregates $\{a(t)\}$, $\{p(t)\}$, $\{B(t)\},\{f(t)\}$ will be formed from corresponding processes tS(t) for above mentioned programming operators through form (1) or form (2) for more complex operators.
Consider hierarchical dynamic tS-program operator:

$$ {}_{A(q)}^{B(q)}St(q) = \begin{Bmatrix} t(q)_{S_1 f(A(q)\cap B(q)-Co(A(q)\cap B(q)))} + {}_{A(q)-A(q)\cap B(q)}^{\{\}}St(q) \\ (B(q)-A(q)\cap B(q)) - (A(q)-A(q)\cap B(q)) \end{Bmatrix} $$

S¹e -Program Operators (form $_D^B S^1 t_B^A$)

For example, $_{\{=:\{p(t)\}\}}^{\{a(t)\}} S^1 t_{\{a(t)\}}^{IF\{\{B(t)\}\{f(t)\}\}\ then\ Q(t)}$.

Consider hierarchical dynamic **S¹e-**program operator: (form $\begin{matrix} _A^B S^1 t_B^A * \\ _{D-A}^{B} S^1 t_B^A \end{matrix}$).

Set₁- Program Operators (form $_D^C S_1 t_B^A$)

$St_{t_0}^{\left\{ {}^{q\left({}_a^a S_1 t_a^a\right)}{}_{W_q}St_{q\left({}_a^a S_1 t_a^a\right)}^{E_q},\ St_{d_r}^{\left\{El^{d_r},\, q\left({}_a^a S_1 t_a^a\right)\right\}} \right\}}$ __ program structure example, where the assemblage point d_r is the cursor [2], it is quite complex self—program.

Remark. Energy of a living organism:

$$ f(r, a(E_q)) = St_{t_0}^{\{ {}^{q\left({}_a^a St_a^a\right)}{}_{W_q}St_{q\left({}_a^a St_a^a\right)}^{E_q},\ St_{d_r}^{\{El^{d_r},\, q\left({}_a^a St_a^a\right)\}} \}} $$

$_a^a St_a^a$ **-**internal energy of a living organism, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point d_r on the energy cocoon of a living organism, W_q- energy prominences from the gap in the cocoon of a living organism, E_q-external energy entering the gap in the cocoon of a living organism, El^{d_r}) - a bundle of fibers of external energy self-capacities, collected at the point of assembly of the cocoon of a living organism.

$_a^a S_1 t_a^a$-- sample $\begin{pmatrix} self \\ oself \end{pmatrix}$-program structure example.

$St_{t_0}^{\left\{ {}^{q\left({}_a^a S_1 t_a^a\right)}{}_{W_q}St_{q\left({}_a^a S_1 t_a^a\right)}^{E_q},\ St_{d_r}^{\left\{El^{d_r},\, q\left({}_a^a S_1 t_a^a\right)\right\}} \right\}}$ can be interpreted as a program operator.

$_a^a S_1 t_a^a$ can be interpreted as a $\begin{pmatrix} self \\ oself \end{pmatrix}$-program operator.

Consider structure examples hierarchical **Set₁**-program operator

1. $\left(\begin{matrix} S_{01}^{et} fB \\ _{Q-B}^{R-B} S_1 t_B^{A-B} \end{matrix} \right)$,
2. $\begin{pmatrix} S_{21}^{et} f_A^B \\ _{Q-A}^{R-A} S_1 t_B^A \end{pmatrix}$.

Entire neural network as instantaneous simultaneous RAM in Sit-elements and self-elements.$self^{self^{\dots^{self}}}$, $f1 \downarrow I \uparrow_{-1}^{1} f_2^{f1\downarrow I\uparrow_{-1}^{1} f_2,\dots^{f1\downarrow I\uparrow_{-1}^{1} f_2,}}$, $sin\infty^{sin\infty^{\dots^{sin\infty}}}$.When activated in a neural network, the entire neural network becomes a working memory. Use of self-energy as activation or from outside. $Q_0 = St\frac{St_{activation}^{S_{mnSt}}}{St_{activation}^{S_{mnSt}}} \to$ self-RAM, $Q_{00} = \frac{{}_{activation}^{S_{mnSt}}St}{{}_{activation}^{S_{mnSt}}St} Q_0$, $Q_{01} = \frac{St_{activation}^{S_{mnSt}}}{St_{activation}^{S_{mnSt}}} Q_0$.

Q_0,Q_{00},Q_{01}-coding,translation,realization in eprograms, use Q_0,Q_{00},Q_{01}-S_{mnSt},Assembler.

Declarations

Availability of Data and Material -

Competing Interest

There are no competing interests. All sections of the article are executed jointly.

Funding

There were no sources of funding for writing the article.

Authors' Contributions

The contribution of the authors is the same, we will not separate.

Acknowledgements

We are grateful to our academic supervisor scientific director: Volodymyr Pasynkov PhD of physic-mathematical science, assistant professor of applied mathematics and calculated techniques department of "National Metallurgical Academy" (Ukraine) for guidance and assistance in writing this article.

References

1. Danilishyn, I., Danilishyn, O., & Pasynkov, V. (2023). THE USAGE OF SIT-ELEMENTS FOR NETWORKS. Collection of scientific papers «ΛΌГОΣ», (March 31, 2023; Zurich, Switzerland), 129-134.
2. Danilishyn I.V. Danilishyn O.V. tS – ELEMENTS. Collection of scientific papers «ΛΌГОΣ» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. pp.156-161, DOI 10.36074/logos-23.06.2023.42.
3. Danilishyn, I., & Danilishyn, O. (2023). VARIABLE HIERARCHICAL DYNAMICAL STRUCTURES (MODELS) FOR DYNAMIC, SINGULAR, HIERARCHICAL SETS AND THE PROBLEM OF COLD THERMONUCLEAR FUSION. Collection of scientific papers «SCIENTIA», (July 14, 2023; Coventry, UK), 113-119.
4. Danilishyn I.V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the proceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), pp. 239-254.