

ISSN 2306-1561

**Automation and Control in Technical Systems (ACTS)**

2015, No 2, pp. 13-24.

DOI: 10.12731/2306-1561-2015-2-2

---



# Model for Describing the Functioning of Neuroprocessor as a Complex Parallel System

**Vitaly Aleksandrovich Romanchuk**

Russian Federation, Ph.D., Senior Lecturer, Department of «Computer Science and Engineering».

Ryazan State University named for S.A. Esenin, 390000, Russian Federation, Ryazan, Svobody Str., 46, aud. 41. Tel.: +7 (4912) 28-05-00, <http://www.rsu.edu.ru>

[v.a.romanchuk@yandex.ru](mailto:v.a.romanchuk@yandex.ru)

**Vladislav Vladislavovich Lukashenko**

Russian Federation, Postgraduate Student, Department of «Computer Science and Engineering».

Ryazan State University named for S.A. Esenin, 390000, Russian Federation, Ryazan, Svobody Str., 46, aud. 41. Tel.: +7 (4912) 28-05-00, <http://www.rsu.edu.ru>

[v.lukashenko@rsu.edu.ru](mailto:v.lukashenko@rsu.edu.ru)

**Abstract.** Consider the specifics of the operation of the neural represented as a complex self-parallel system. Proposed schemes, algorithms, and a description of finite state machines of each node on the neural example family NM640x, produced RC "Module".

**Keywords:** neuroprocessor, model of description, function, parallel systems, computer system.

---

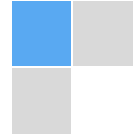
ISSN 2306-1561

**Автоматизация и управление в технических системах (АУТС)**

2015. – № 2. – С. 13-24.

DOI: 10.12731/2306-1561-2015-2-2

---



УДК 004.383.3

## **Модель описания функционирования нейропроцессора как сложной параллельной системы**

**Романчук Виталий Александрович**

Российская Федерация, кандидат технических наук, старший преподаватель кафедры «Информатики и вычислительной техники».

ФГБОУ ВПО «Рязанский государственный университет имени С.А. Есенина», 390000, Российская Федерация, г. Рязань, ул. Свободы, 46, ауд. 41, Тел.: +7 (4912) 28-05-00, <http://www.rsu.edu.ru>

[v.a.romanchuk@yandex.ru](mailto:v.a.romanchuk@yandex.ru)

**Лукашенко Владислав Владиславович**

Российская Федерация, аспирант кафедры «Информатики и вычислительной техники».

ФГБОУ ВПО «Рязанский государственный университет имени С.А. Есенина», 390000, Российская Федерация, г. Рязань, ул. Свободы, 46, ауд. 41, Тел.: +7 (4912) 28-05-00, <http://www.rsu.edu.ru>

[v.lukashenko@rsu.edu.ru](mailto:v.lukashenko@rsu.edu.ru)

**Аннотация.** Рассматривается специфика функционирования нейропроцессора, представленного в виде сложной самостоятельной параллельной системы. Предложены схемы и описания в виде конечных автоматов каждого узла нейропроцессора на примере семейства NM640x, выпускаемых НТЦ "Модуль". Также предложены алгоритмы оценки эффективности выполнения микрокоманды нейропроцессора.

**Ключевые слова:** нейропроцессор, модель описания, функционирование, параллельные системы, вычислительная система.

### **1. Введение**

Мультипроцессорные решения для встроенных систем разрабатываются для обеспечения все возрастающих потребностей в вычислительных мощностях. Такие архитектуры включают два или более процессора вместе с разнообразным периферийным оборудованием, чем обеспечивается создания высокопроизводительных

вычислительных систем. В то время как для таких проблем проектирования, как разделение систем на модули, проверка аппаратно-программного взаимодействия сложных систем, - существует достаточное количество инструментов, для облегчения труда разработчиков прикладного программного обеспечения (ПО) практически ничего не сделано. Но для подготовки эффективных параллельных программ, особенно для нейропроцессорных систем(НПС), имеется ряд проблем [1]:

1. НПС являются пока дорогим и штучным товаром и не каждая организация может себе позволить приобрести её.

2. Даже при наличии своей НПС бывает трудно организовать постановку экспериментальных программ. Это связано с большой сложностью разработки программ, риском ошибки и другими факторами.

3. Специфика параллельного программирования такова, что часто бывает необходимо организовать эксперименты на НПС с различной структурой. К сожалению, большинство организации не может себе позволить содержать целый парк НПС на все случаи жизни или на ходу перестраивать имеющиеся.

Указанные выше проблемы можно обойти путём постановки экспериментов на моделях. Поэтому целью исследования является реализация модели описания функционирования нейропроцессора как сложной параллельной системы.

## 2. Теоретические исследования

Для описания класса нейропроцессоров будем использовать описание процессоров семейства NM640x, выпускаемых НТЦ "Модуль" (г.Москва). Нейропроцессоры обрабатывают данные параллельно, по принципу вектора, то есть он представляет собой параллельную систему [2], а получаемые нейропроцессорные системы при соответствующем уровне детализации могут рассматриваться как "системы в системе".

В процессоре семейства NM640x возможно два варианта команд (рисунок 1) [2]: скалярные команды – используются для подготовки данных к выполнению операции на векторном процессоре и векторные команды – основные операции процессора.

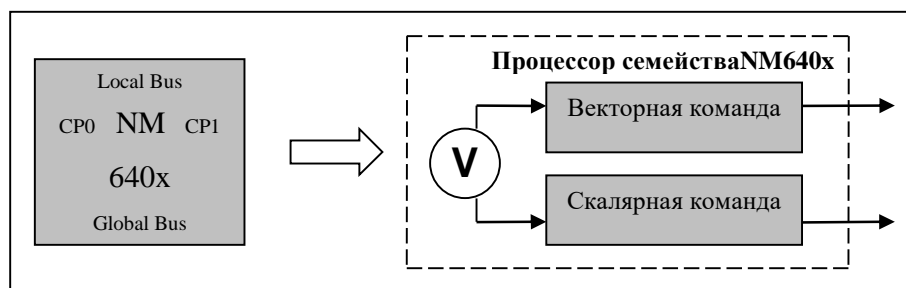


Рисунок 1 – Команды нейропроцессора семейства NM640x

Каждая из команд может выполнять несколько операций параллельно. Тогда каждый процессорный модуль (ПМ) можно представить в виде системы параллельной обработки данных.

## 2.1. Анализ архитектуры нейропроцессора как системы параллельной обработки информации при выполнении скалярной команды

Скалярная команда нейропроцессора семейства NM640x имеет левую и правую часть, операции каждой части выполняются параллельно. Возможна пустая (nul) команда в левой или правой части.

Структурная схема выполнения скалярной команды выглядит следующим образом (рисунок 2):

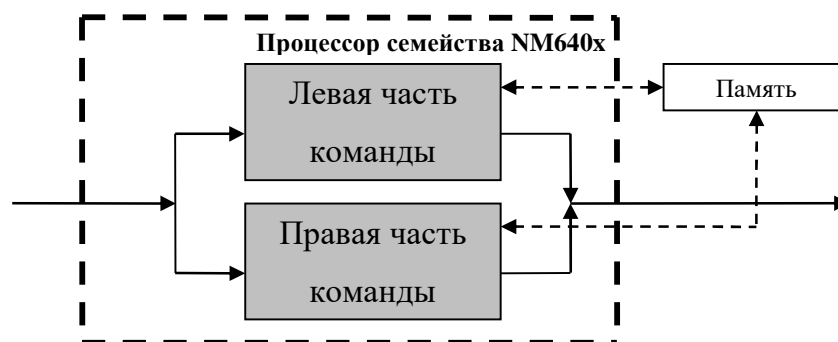


Рисунок 2 – Скалярная команда нейропроцессора

Левая часть содержит команды:

- команды загрузки/записи, пересылки значений регистров;
- команды адресной арифметики;
- специальные скалярные команды;
- команды безусловного и условного перехода;
- команды перехода и возврата из подпрограммы или прерывания.

Правая часть содержит команды:

- арифметические операции;
- логические операции;
- сдвиговые операции.

Детализированная схема выполнения скалярной операции представлена на рисунке 3.

Команды для реализации блоков загрузки и модификации находятся в левой части команды. Блоки загрузки и модификации для различных регистров можно объединить в один или 2 отдельных модуля: загрузки и модификации, т.к. операция выполняется только для какого-либо одного регистра (пары регистров).

Блоки вычисления находятся в правой части команды и также могут быть объединены в один модуль. Тогда, исходя из приведенных выше схем, можно представить процессор при выполнении скалярной операции как параллельную систему следующим образом (рисунок 4).

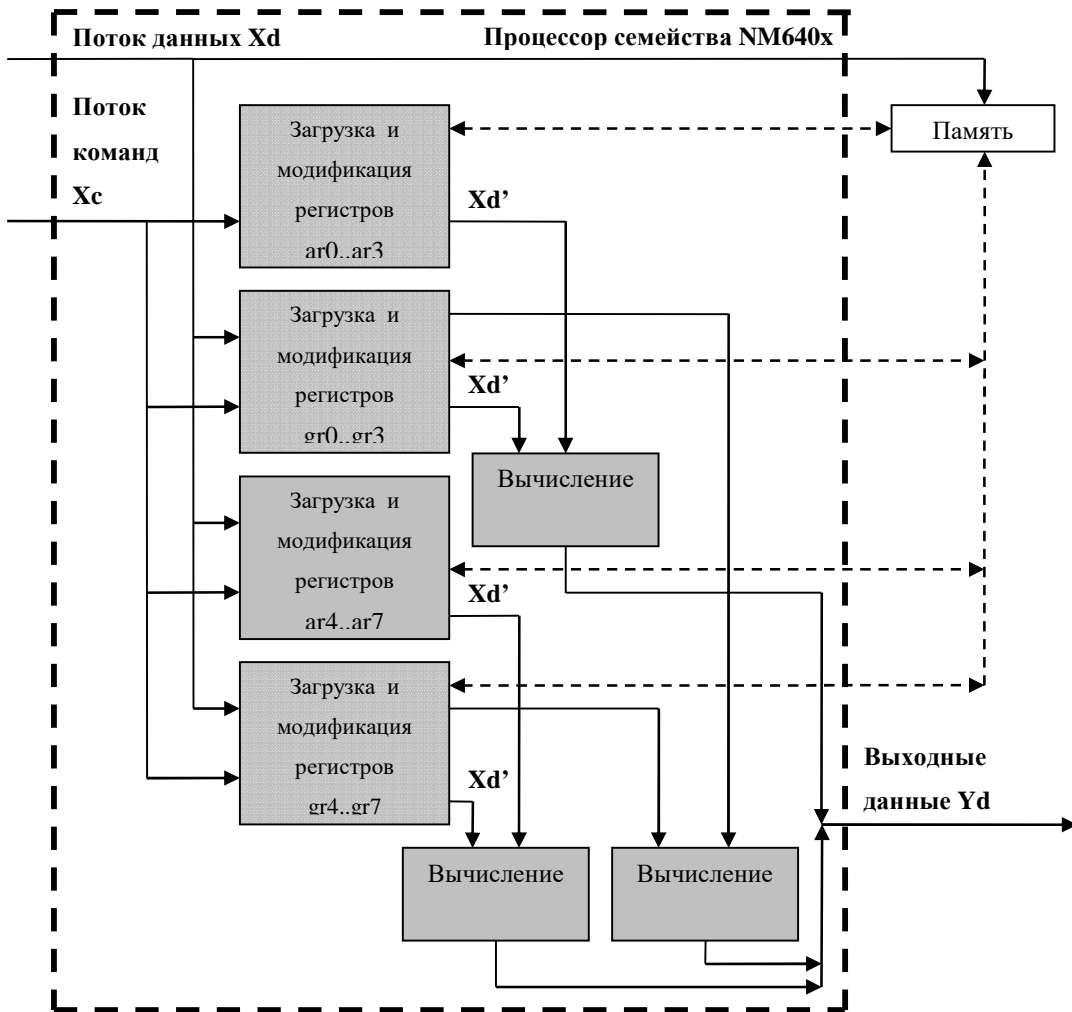


Рисунок 3 – Детализированная схема выполнения скалярной операции

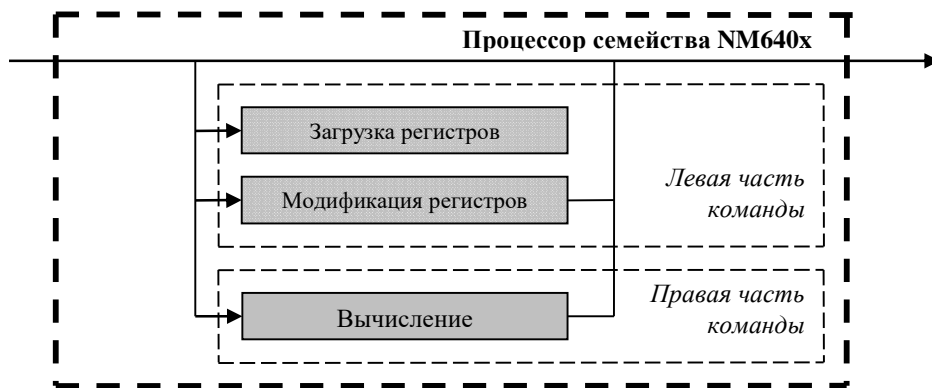


Рисунок 4 – Выполнение скалярной команды процессора, представленное в виде параллельной системы

## 2.2. Архитектура нейропроцессора как системы параллельной обработки информации при выполнении векторной команды

Векторная команда нейропроцессора семейства NM640x также имеет левую и правую часть, возможна пустая команда в левой или правой части.

Левая часть содержит команды:

- команды загрузки данных в векторный процессор;
- команды выгрузки данных из векторного процессора;
- специальные векторные команды.

Правая часть содержит команды:

- взвешенное суммирование (матричное умножение);
- маскирование;
- арифметические операции;
- логические операции;
- операция циклического сдвига;
- операции активации операндов;
- выгрузка управляющих векторных регистров.

Кроме этого, как описано в первой главе, нейропроцессор семейства NM640x работает с "упакованными" данными – блоками по 64 бита каждый [2]. Таким образом, в правой части за 1 такт может выполняться 2 операции по 32 бита, 8 операций по 8 бит и т.д. Максимальное количество выполняемых за один такт операций в векторном процессоре в правой части команды составляет 64 однобитных операции.

Сложность состоит в том, что трудно оценить число операций в той или иной команде, следовательно, необходимо рассматривать каждую операцию отдельно. Кроме того, операции имеют разное время выполнения (для операций с константой – 2 такта, для остальных операций – 1 такт).

Детализируем выполнение левой части команды (рисунок 5) [2].

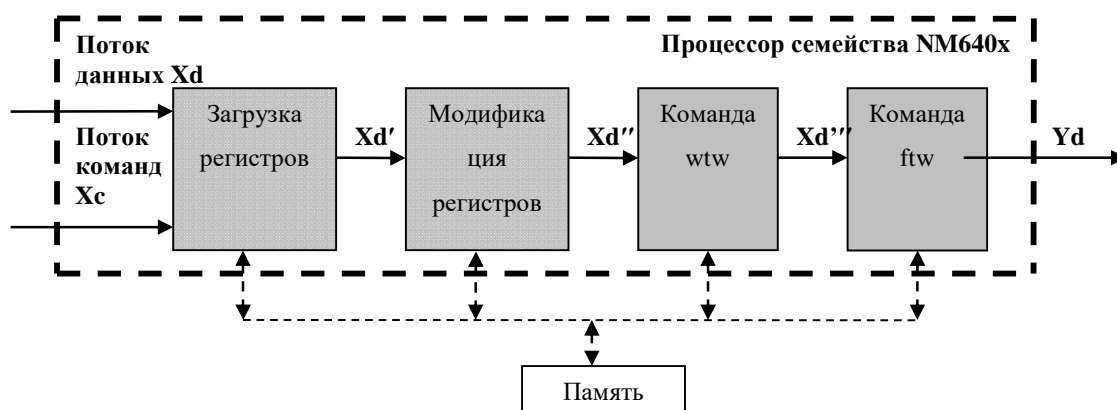
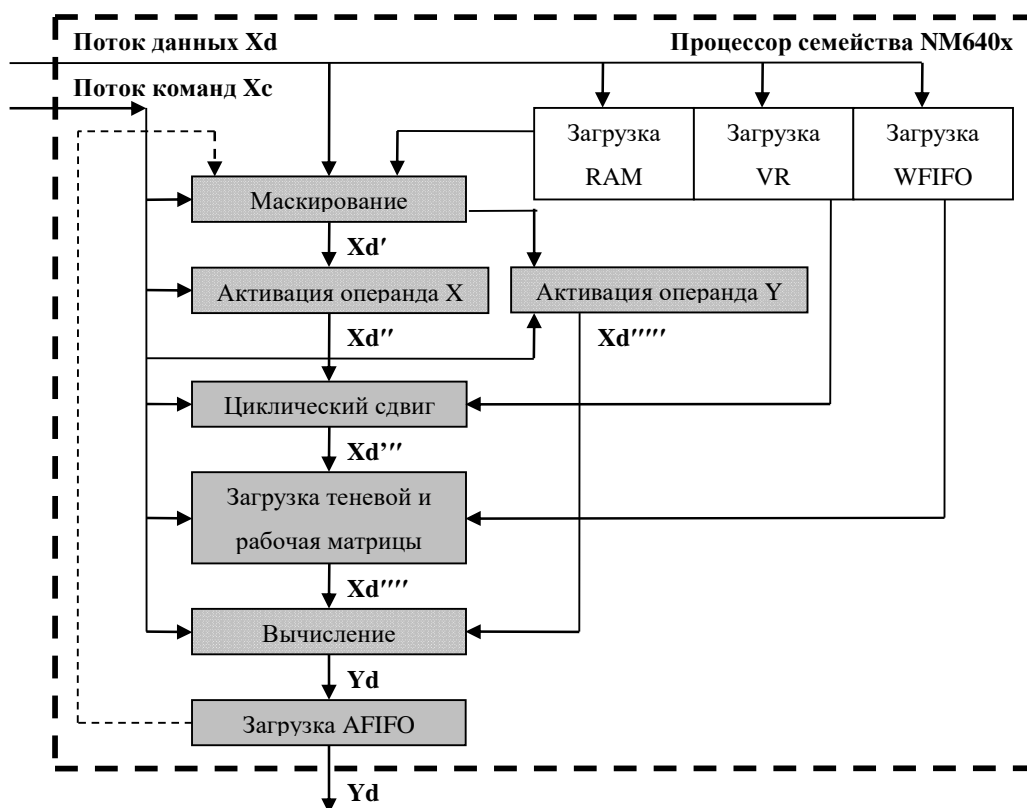


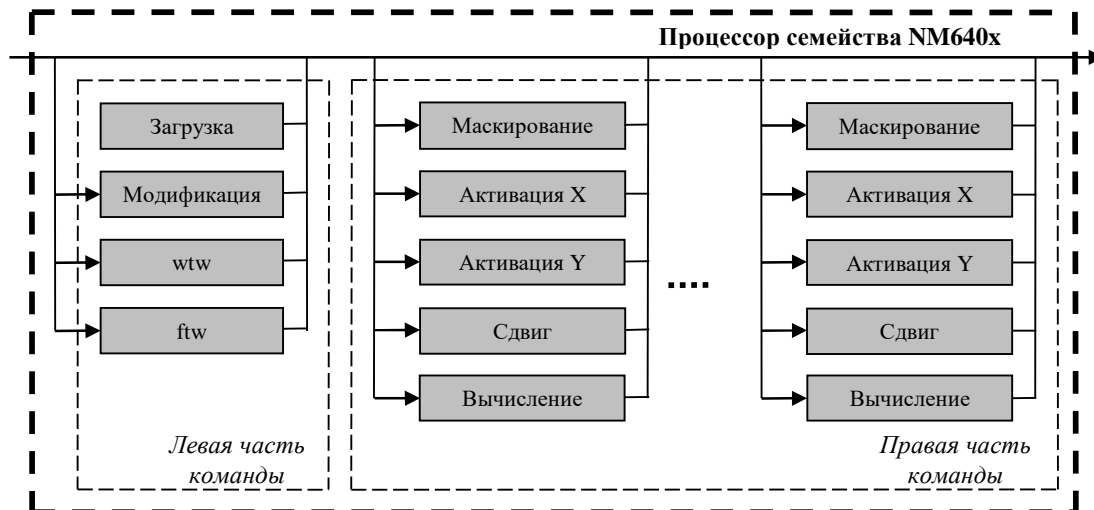
Рисунок 5 - Детализированная схема выполнения левой части векторной команды

Детализируем выполнение правой части команды (рисунок 6).



**Рисунок 6 – Детализированная схема выполнения правой части векторной команды**

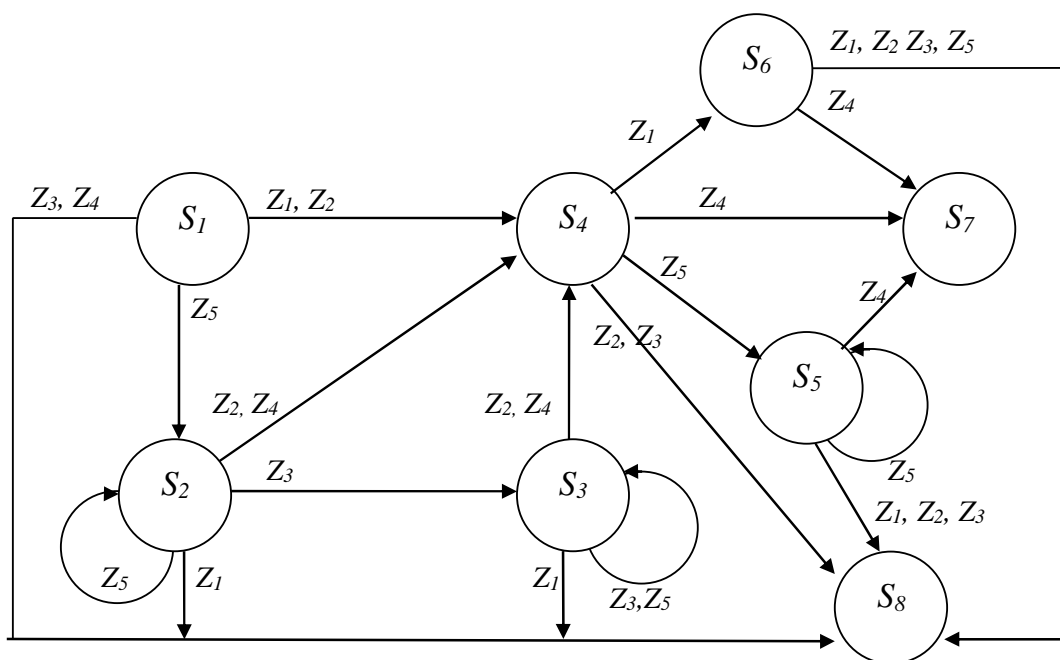
Исходя из приведенных схем, ПМ при выполнении векторной команды, можно представить в виде параллельной системы (рисунок 7).



**Рисунок 7 – Выполнение векторной команды процессора, представленного в виде параллельной системы**

Для описания функционирования нейропроцессора семейства NM640х будем использовать представление нейропроцессора как параллельной вычислительной системы, показанное на рисунках 3 и 7.

Важное значение при функционировании ПМ играет кортеж микрокоманд  $PR = \langle MK_1, MK_2, \dots, MK_M \rangle$ . Пусть  $MK^{(k)}$  – некоторая микрокоманда из данного кортежа. Тогда можно описать функционирование ПМ с помощью конечного автомата  $KS = (S, Z, W, \delta, \lambda, S_1)$  при выполнении скалярной микрокоманды  $MK^{(k)}$  (рисунок 8) и конечного автомата  $KV = (S, Z, W, \delta, \lambda, S_1)$  при выполнении векторной микрокоманды  $MK^{(k)}$  (наиболее важный участок на рисунке 9) на процессоре семейства NM640х.



**Рисунок 8 – Конечный автомат  $KS$  при выполнении скалярной команды**

Рассмотрим множество состояний  $S$  автомата  $KS$ :

$S_1$  - начальное состояние;

$S_2$  - состояние, означающее что блок “Загрузка” не простаивает;

$S_3$  - состояние, означающее что блок “Модификация” не простаивает;

$S_4$  - состояние, означающее что закончена обработка левой части команды;

$S_5$  - состояние, означающее что блок “Вычисление” не простаивает;

$S_6$  - состояние, означающее что правая часть команды отсутствует;

$S_7$  - конечное состояние, означающее что команда закончена;

$S_8$  - состояние, означающее что команда имеет неверный формат.

Входной алфавит  $Z$  автомата  $KS$  представляет собой лексемы:

$Z_1$  - лексема “nul”;

$Z_2$  - лексема “with”;



$Z_3$  - лексемы модификации (“++”, “--”, “==”, ...).

$Z_4$  - лексема, определяющая конец команды.

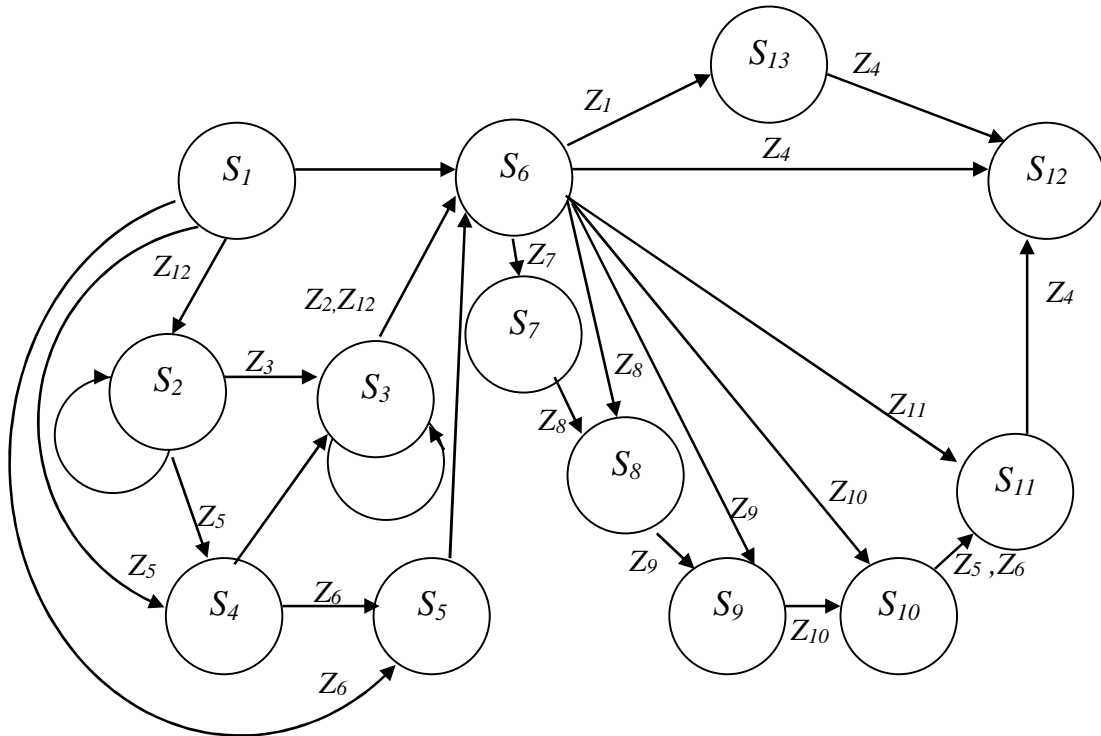
$Z_5$  - остальные допустимые лексемы языка нейроассемблера.

Выходной алфавит  $W$  автомата  $KV$ :

$W_1 = \lambda(S_2)$ ; сигнал означает, что задействован блок “Загрузка”.

$W_2 = \lambda(S_3)$ ; сигнал означает, что задействован блок “Модификация”.

$W_3 = \lambda(S_5)$ ; сигнал означает, что задействован блок “Вычисление”.



**Рисунок 9 – Участок конечного автомата  $KV$  при выполнении векторной микрокоманды**

Рассмотрим множество состояний  $S$  автомата  $KV$ :

$S_1$  - начальное состояние;

$S_2$  -  $S_5$  состояния, означающие что блоки “Загрузка”, “Модификация”, “wtw”, “ftw” не простаивают;

$S_6$  - состояние, означающее, что закончена обработка левой части команды;

$S_7$  -  $S_{11}$  - состояния, означающие что блоки “Маскирование”, “Активация x”, “Активация y”, “Сдвиг”, “Вычисление” не простаивают;

$S_{12}$  - конечное состояние, означающее что команда закончена.

Входной алфавит автомата  $KV$ :

$Z_1$  - лексема “nul”;  $Z_2$  - лексема “with”;  $Z_3$  - лексемы модификации (“++”, “--”, “==”, ...);  $Z_4$  - лексема, определяющая конец команды;  $Z_5$  - лексема “wtw”;  $Z_6$  -

лексема “ftw”;  $Z_7$ - лексема “mask”;  $Z_8$ - лексема “active x”;  $Z_9$ - лексема “active y”;  $Z_{10}$ - лексема “shift”;  $Z_{11}$ - лексема “vsum”,  $Z_{12}$ - остальные допустимые лексемы языка нейроассемблера.

Выходной алфавит  $W$  автомата  $KV$  :

$W_1 = \lambda(S_2)$ ; сигнал означает, что задействован блок “Загрузка”.

$W_2 = \lambda(S_3)$ ; сигнал означает, что задействован блок “Модификация”.

$W_3 = \lambda(S_4)$ ; сигнал означает, что задействован блок “wtw”.

$W_4 = \lambda(S_5)$ ; сигнал означает, что задействован блок “ftw”.

$W_5 = \lambda(S_7)$ ; сигнал означает, что задействован блок “Маскирование”.

$W_6 = \lambda(S_8)$ ; сигнал означает, что задействован блок “Активация x”.

$W_7 = \lambda(S_9)$ ; сигнал означает, что задействован блок “Активация y”.

$W_8 = \lambda(S_{10})$ ; сигнал означает, что задействован блок “Сдвиг”.

$W_9 = \lambda(S_{11})$ ; сигнал означает, что задействован блок “Вычисление”.

Тогда оценки эффективности для ПМ будут определяться в зависимости от выходных сигналов  $\{W_1, W_2, W_3\}$  автомата  $KS$  и выходных сигналов  $\{W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8, W_9\}$  автомата  $KV$ .

Пусть  $T_i$  - время работы микрокоманды  $MK^{(k)}$ ;

$T_{ps}^{(k)}$  - время обработки скалярных микрокоманд;

$T_{pv}^{(k)}$  - время обработки векторных микрокоманд;

$Sk \in \{1, 2, \dots, 64\}$  - величина, зависящая от числа векторных операций в команде за счет “упакованных” операндов;

$C_s$ ;  $C_s \in \{0, 1, \dots, 3\}$  - число зафиксированных сигналов из множества  $\{W_1, W_2, W_3\}$  автомата  $KS$  во время выполнения команды  $MK^{(k)}$ ;

$C_{v1}$ ;  $C_{v1} \in \{0, 1, \dots, 4\}$  - число зафиксированных сигналов из множества  $\{W_1, W_2, W_3, W_4\}$  автомата  $KV$  во время выполнения команды  $MK^{(k)}$ ;

$C_{v2}$ ;  $C_{v2} \in \{0, 1, \dots, 5\}$  - число зафиксированных сигналов из множества  $\{W_5, W_6, W_7, W_8, W_9\}$  автомата  $KV$  во время выполнения команды  $MK^{(k)}$ .

### 3. Алгоритмы оценки эффективности ПМ

Алгоритм оценки эффективности ПМ при выполнении команды  $MK_j$  можно описать следующим образом:

**Шаг 1.** Если получен любой сигнал из множества  $\{W_1, W_2, W_3\}$  автомата  $KS$ , то  $T_{ps}^{(k)} = T_{ps}^{(k)} + T_i$ ,  $C_s = C_s + 1$ .

**Шаг 2.** Если получен любой сигнал из множества  $\{W_1, W_2, W_3, W_4\}$  автомата  $KV$ , то  $T_{pv}^{(k)} = T_{pv}^{(k)} + T_i$ ,  $C_{v1} = C_{v1} + 1$ ;

**Шаг 3.** Если получен любой сигнал из множества  $\{W_5, W_6, W_7, W_8, W_9\}$  автомата  $KV$ , то  $T_{pv}^{(k)} = T_{pv}^{(k)} + T_i * Sk$ ,  $C_{v2} = C_{v2} + 1$ .

**Шаг 4.** Конец алгоритма.

Таким образом, в результате работы конечного автомата можно получить оценки эффективности ПМ при выполнении микрокоманды  $MK^{(k)}$ :

- время обработки для скалярной микрокоманды:  $T_{ps}^{(k)} = C_s * T_i$
- время обработки для векторной команды:  $T_{pv}^{(k)} = T_i * (C_{v1} + C_{v2} * Sk)$
- время простоев для скалярной микрокоманды:  $T_{nps}^{(k)} = T_i * (3 - C_s)$
- время простоев для векторной микрокоманды:  $T_{npv}^{(k)} = T_i * (4 - C_{v1} + (6 - C_{v2}) * Sk)$
- время проигрыша:  $T_n^{(k)} = T_i - T_i * (C_{v1} + C_{v2} + C_s)$

Тогда оценки эффективности ПМ для всего кортежа  $PR = \langle MK_1, MK_2, \dots, MK_M \rangle$  будут определяться следующим образом:

- время обработки:  $T_{psv} = \sum_{i=1}^k (C_s * T_i + T_i * (C_{v1} + C_{v2} * Sk))$
- время простоев:  $T_{npsv} = \sum_{i=1}^k (T_i * (3 - C_s) + T_i * (4 - C_{v1} + (5 - C_{v2}) * Sk))$
- время проигрыша:  $T_{nsv} = \sum_{i=1}^k (T_i - T_i * (C_{v1} + C_{v2} + C_s))$

Рассмотрим алгоритм оценки эффективности ПМ с использованием реализации конечных автоматов  $KS$  и  $KV$ , который представляет собой сумматор оценок для всех команды кортежа  $PR = \langle MK_1, MK_2, \dots, MK_M \rangle$ .

**Шаг 1.** Если  $j \geq k$ , то переход на шаг 5.

**Шаг 2.** Расчет  $T_p^{(k)} = T_{ps}^{(k)} \vee T_{pv}^{(k)}$ ,  $T_p = T_p + T_p^{(k)}$ .

**Шаг 3.** Расчет  $T_{np}^{(k)} = T_{nps}^{(k)} \vee T_{npv}^{(k)}$ ,  $T_{np} = T_{np} + T_{np}^{(k)}$ .

**Шаг 4.** Расчет  $T_n^{(k)} = T_{ns}^{(k)} \vee T_{nv}^{(k)}$ ,  $T_n = T_n + T_n^{(k)}$ ,  $j = j + 1$ , переход на шаг 1.

**Шаг 5.** Конец алгоритма.

## 4. Заключение

В рамках исследования были предложены:

1. Схемы выполнения векторной и скалярной операций на нейропроцессоре на примере семейства NM640x, выпускаемых НТЦ "Модуль".

2. Описания моделей скалярного и векторного узла нейропроцессора в виде конечных автоматов на примере семейства NM640x, выпускаемых НТЦ "Модуль".

3. Алгоритмы оценки эффективности выполнения скалярной и векторной команды нейропроцессора на примере семейства NM640x, выпускаемых НТЦ "Модуль".

### Список информационных источников

- [1] Галушкин А.И., Нейрокомпьютеры. Кн.3. М: ИПРЖР. 2000.
- [2] НТЦ «Модуль»: сайт НТЦ «Модуль», 2011: URL: [http:// www.module.ru](http://www.module.ru).
- [3] Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры: Учеб.пособие для вузов. М.: Изд-во МГТУ им. Н.Э.Баумана, 2004. 400 с.
- [4] Ручкин В.Н., Романчук В.А., Фулин В.А. Когнитология и искусственный интеллект. – Рязань: Узорочь, 2012. – 260 с.
- [5] Vladimir Ruchkin, Vitaliy Romanchuk, Roman Sulitsa. Clustering, Restorability and Designing Of Embedded Computer System Based On Neuroprocessors // Proceedings of the 2nd Mediterranean Conference on Embedded Computing (MECO). - Budva, Montenegro, 2013. - С.58-62.
- [6] Романчук В.А., Ручкин В.Н. Оценка результатов моделирования вычислительных систем на базе нейропроцессоров // Известия тульского государственного университета. Технические науки. – Тула: Издательство ТулГУ, 2013. – Вып.9. - Ч.2. – С.194-203.
- [7] Романчук В.А., Ручкин В.Н. Алгоритмы анализа вычислительных структур на базе нейропроцессоров // Вестник РГРТУ. – Рязань: РГРТУ, 2012. – №2. – Вып.40. – С.60–66.
- [8] Романчук В.А. Моделирование нейропроцессорных систем // Отраслевые аспекты технических наук. – 2013. – №10(34). – С.19-24.
- [9] Романчук В.А. Разработка алгоритмов определения связей элементов вычислительной структуры на базе нейропроцессоров // Информатика и прикладная математика: межвуз. сб. науч. тр. – Рязань: РГУ имени С.А.Есенина, 2011. – Вып.17. – С.102–105.
- [10] Романчук В.А., Ручкин В.Н. Разработка программного комплекса для моделирования и анализа нейропроцессорных систем обработки изображений // Цифровая обработка сигналов. – 2010. – №1. – С.53–58.
- [11] Романчук В.А., Ручкин В.Н., Колмыков М.В. Возможности программного комплекса NM Model для разработки и отладки программ обработки изображений // Вестник РГРТУ. – Рязань: РГРТУ, 2008. – №2. – Вып. 24. – С.83–85.
- [12] Романчук В.А., Ручкин В.Н., Фулин В.А. Разработка модели сложной нейропроцессорной системы // Цифровая обработка сигналов. – 2012. – №4. – С.70–74.
- [13] Романчук В.А., Лукашенко В.В. Разработка математической модели реструктуризуемого под классы задач, виртуализируемого кластера GRID-системы на базе нейропроцессоров // Вестник РГУ имени С.А.Есенина. – Рязань: РГУ имени С.А.Есенина, 2014. – №1(42). – С. 177- 182.
- [14] Романчук В.А. Инновационный программный комплекс моделирования вычислительных систем на базе нейропроцессоров "НейроКС" // Современные научные исследования и инновации. – Декабрь, 2012 [Электронный ресурс]. – URL: <http://web.snauka.ru/issues/2012/12/19407>.
- [15] Романчук В.А., Ручкин В.Н. Разработка программных средств анализа нейропроцессорных систем // Вестник РГРТУ. – Рязань: РГРТУ, 2010. – №2. – Вып.32. – С.61–67.