

7. Vasilev, Y. A., Nanova, O. G., Blokhin, I. A., Reshetnikov, R. V., Vladzimirskyy, A. A., & Omelyanskaya, O. V. (2024). Priority radiomics parameters for computed tomography in head and neck malignancies: a systematic review. *Digital Diagnostics*, 5(2). doi: 10.17816/DD623240
8. Avanzo M, Wei L, Stancanella J, et al. Machine and deep learning methods for radiomics. *Med Phys*. 2020;47(5):e185-e202. doi:10.1002/mp.13678
9. De Araujo Faria V, Azimbagirad M, Viani Arruda G, et al. Prediction of Radiation-Related Dental Caries Through PyRadiomics Features and Artificial Neural Network on Panoramic Radiography. *J Digit Imaging*. 2021;34(5):1237-1248. doi:10.1007/s10278-021-00487-6
10. Franzese C, Lillo S, Cozzi L, et al. Predictive value of clinical and radiomic features for radiation therapy response in patients with lymph node-positive head and neck cancer. *Head & Neck*. 2023; 45(5): 1184-1193. doi:10.1002/hed.27332
11. Gillies RJ, Kinahan PE, Hricak H. Radiomics: Images Are More than Pictures, They Are Data. *Radiology*. 2016;278(2):563-577. doi:10.1148/radiol.2015151169
12. Mwangi B, Tian TS, Soares JC. A review of feature reduction techniques in neuroimaging. *Neuroinformatics* 2014; 12:229–244.
13. Koçak B, Durmaz EŞ, Ateş E, Kılıçkesmez Ö. Radiomics with artificial intelligence: a practical guide for beginners. *Diagn Interv Radiol*. 2019 Nov;25(6):485-495. doi: 10.5152/dir.2019.19321. PMID: 31650960; PMCID: PMC6837295.

## МОДЕЛИРОВАНИЕ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР В ПРОГРАММНОМ КОМПЛЕКСЕ GPTN

Муравьев Н. Д., Кулагин В. П.\*, Выпова А. М.\*  
РТУ МИРЭА, \* Московский технический университет связи и информатики  
+7 (995) 658-91-89, muravev.nd@mail.ru

Статья демонстрирует возможности разрабатываемого программного комплекса GPTN, который предназначен для работы с разными видами сетей Петри. В качестве демонстрации рассмотрено решение задачи анализа и модифицирования сложной вычислительной структуры. В процессе решения задачи показаны возможности программного комплекса в области анализа структур выраженных в терминах сетей Петри, что позволило выявить недостатки модификации модели и исправить их.

Ключевые слова: сеть Петри, дерево достижимых разметок, сложные вычислительные структуры, моделирование вычислительных структур.

The design procedure of specified operating life of fiber- optic cables. Muravev N. D., Kulagin V. P.\*, Vypova A. M.\* RTU MIREA, \* Moscow Technical University of Communications and Informatics.

The article demonstrates the capabilities of the developed GPTN software package, which is designed to work with different types of Petri nets. As a demonstration, the solution to the problem of analyzing and modifying a complex computational structure is considered. In the process of solving the problem, the capabilities of the software package in the field of analyzing structures expressed in terms of Petri nets were shown, which made it possible to identify shortcomings in the modification of the model and fix them.

Keywords: Petri net, reachable marking tree, complex computational structures, modeling of computational structures.

### Введение

Использование сетевых моделей, основанных на сетях Петри (СП) [1, 2, 3] всегда привлекало исследователей, начиная с момента появления данного формализма. В настоящее время, с возрастанием интереса к распределенным структурам, многоагентным системам и алгоритмам взаимодействия отдельных подсистем, интерес к СП только возрастает [4-6]. Аппарат СП широко используется в моделировании задач бизнеса, химии, биологии, медицине и других областях, также решаются вопросы, связанные с проектированием и верификацией сложных систем, с построением программных средств на основе сетевых моделей и оценки возможности распараллеливания программ, с применением сетевых моделей для описания и анализа автоматизированных систем, сетевых протоколов, вычислительных сетей, многопроцессорных систем и др. [7-10] Кроме того, СП являются инструментом, эффективно поддерживающим структурный подход к проектированию сложных систем [11-14].

Для моделирования и анализа СП существует множество программных средств [16-18], которые имеют разнообразный функционал, который отличается от продукта к продукту. Эти программные средства решают ряд важных задач при исследовании сложных систем, выраженных в терминах СП (СП-модели). Со временем некоторые программные средства моделирования СП перестали поддерживаться

разработчиками по тем или иным причинам, что конечно говорит об их устаревании по отношению к развитию теории сетей Петри.

В данной статье будет продемонстрирован функционал программного комплекса GPTN [15] в задачах моделирования вычислительных систем с помощью сетей Петри. Данный комплекс активно разрабатывается с использованием современных технологий и языков программирования.

#### Постановка проблемы

Основной задачей данной статьи является оценка возможностей программного комплекса GPTN в решении следующих вопросов:

- графическое представление вычислительной структуры в терминах сетей Петри;
- построение дерева достижимых разметок (ДДР);
- анализ ДДР;
- симуляция СП-модели.

В качестве примера возьмем сложную вычислительную структуру (рис. 1), которая состоит из 3 вычислительных структур BC1 (Вычислительная структура 1), BC2, BC3 и канала ввода-вывода, состоящий из подканалов ПКВ1, ПКВ2, ПКВ3. BC1 выполняет ввод данных с использованием подканалов ПКВ1 и ПКВ2. BC2 выполняет обработку данных на процессоре со следующей структурой ((ПЭ1 || ПЭ2) – ПЭ3) (ПЭ – процессорный элемент). BC3 выполняет вывод данных с использованием подканалов ПКВ2 и ПКВ3.

Задача данной статьи продемонстрировать возможности анализируемого программного комплекса в анализе заданной сложной вычислительной структуре, а также поиска и исправления неисправного поведения этой структуры.

В данном разделе представлен анализ функционала программного комплекса GPTN по заданному порядку действий:

- графическое представление вычислительной модели в терминах СП;
- анализ полученной СП-модели при помощи матричных методов и дерева достижимых разметок;
- исправление неисправностей модели;
- анализ исправленной модели.

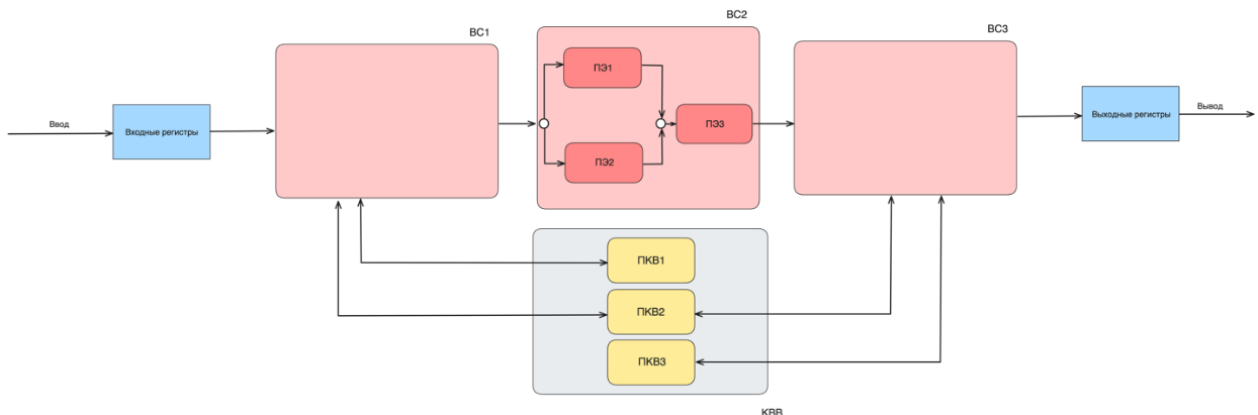


Рисунок 1 – Составная вычислительная структура.

#### Анализ

Комплексную вычислительную структуру (рис. 1) можно описать в терминах СП как показано на (рис. 2), где каждый компонент модели имеет физический смысл: P1 – условие наличия данных для ввода в BC; T1 – начало ввода данных в BC1; P2 – данные введены в BC1; T2 – окончание ввода данных в BC1; P3 – данные введены в BC1 и готовы для ввода в BC2; T3 – начало ввода данных в BC2; P4 – данные введены в BC2; T4 – окончание ввода данных в BC2; P5 – данные введены в BC2 и готовы для ввода на ПЭ; T5 – организация параллельного ввода данных в ПЭ1 и ПЭ2; P511, P521 – данные присутствуют для ввода в ПЭ1 и ПЭ2 соответственно; T511, T521 – окончание ввода данных в ПЭ1 и ПЭ2 соответственно; P512, P522 – данные введены и готовы для обработки в ПЭ1 и ПЭ2 соответственно; T512, T522 – начало обработки данных в ПЭ1, ПЭ2 соответственно; P513, P523 – идет обработка данных в ПЭ1 и ПЭ2 соответственно; T513, T523 – завершение обработки данных в ПЭ1 и ПЭ2 соответственно; P514, P524 – данные обработаны и готовы к выводу из ПЭ1, ПЭ2 соответственно; T6 – параллельный вывод обработанных данных из ПЭ1 и ПЭ2; P6 – данные выведены из ПЭ1 и ПЭ2 и готовы к вводу в ПЭ3; T7 – начало ввода данных в ПЭ3; P7 – данные введены и готовы для обработки в ПЭ3; T8 – окончание ввода данных и начало обработки данных в ПЭ3; P8 – идет обработка данных в ПЭ3; T9 – окончание обработки

данных в ПЭ3; Р9 – данные обработаны и готовы к выводу из ПЭ3; Т10 – вывод обработанных данных из ПЭ3; Р10 – данные выведены из ПЭ3 и готовы к выводу из ВС2; Т11 – вывод данных из ВС2; Р11 – данные выведены из ВС2 и готовы для ввода в ВС3; Т12 – начало ввода данных в ВС3; Р12 – данные введены в ВС3; Т13 – окончание ввода данных в ВС3; Р13 – данные введены в ВС3 и готовы к выводу из ВС; Т14 – начало вывода данных из ВС; Р14 – данные выведены из ВС; Т15 – окончание вывода данных из ВС; Р15 – условие занятости ПКВ3; Р16 – условие занятости ПКВ2; Р17 – условие занятости ПКВ1.

Для простоты восприятия модели её можно упростить, введя иерархические переход Т16 (рис. 3). Данный переход включает в себя фрагмент модели Т3 – Т13. Для построенной иерархической СП модели построим дерево достижимых разметок (рис. 4).

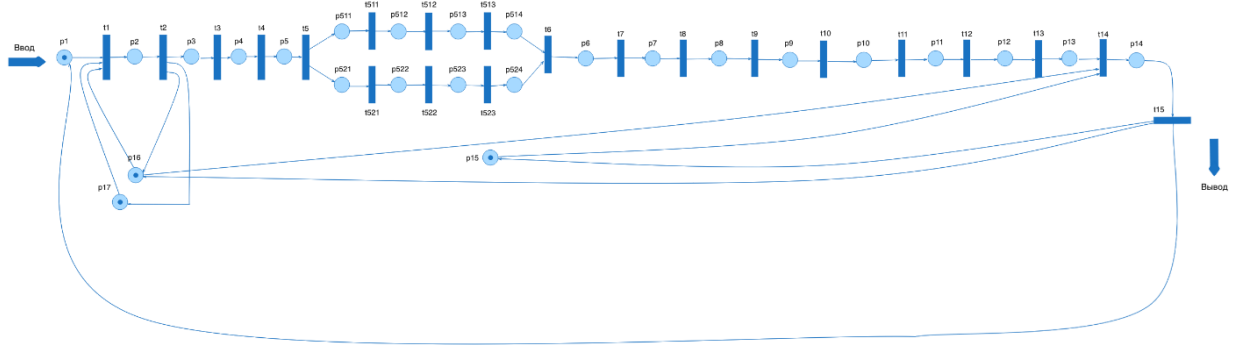


Рисунок 2 – СП модель комплексной вычислительной структуры.

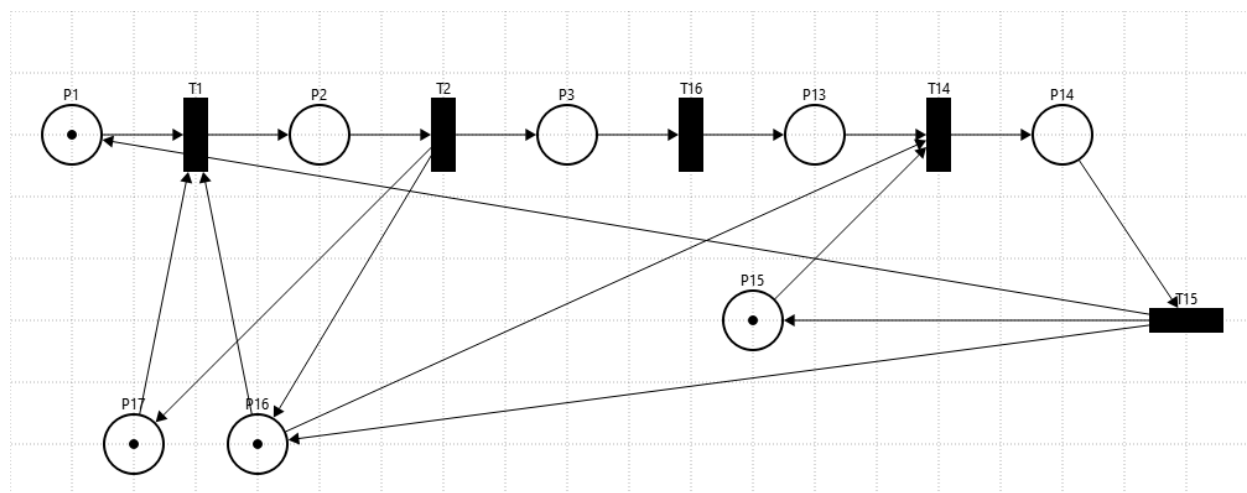


Рисунок 3 – СП модель комплексной вычислительной структуры с иерархическим переходом Т16.

Стоит отметить несколько особенностей программы GPTN:

- иерархический переход представлен так же, как и обычный переход, то есть имя начинается с латинской буквы “Т”;
- в дереве достижимых разметок появление бесконечной разметки в позиции считает эту маркировку равной “тупиковой”. Это упрощение позволяет уменьшить величину ДДР, также показать является ли сеть k-ограниченной и определить исправность функционирования сети;
- в ДДР используются 3 типа позиций: внутренняя – черный цвет, дублирующаяся – синий, тупиковая – красный.

Проанализировав построенное ДДР (рис. 4) можно сделать вывод, что сеть функционирует исправно.

Чтобы продемонстрировать возможности нахождения неправильно функционирующей модели, введем дополнительное условие, что ввод данных в систему разрешается после отправки предварительно введенных данных на обработку в процессорные элементы (ПЭ). Данное условие повышает интенсивность поступления входных данных. Чтобы оно выполнялось, надо убрать соединение Т15 – Р1, и добавить соединение Т2 – Р1 (рис. 5).

Анализ ДДР (рис. 6) модифицированной вычислительной структуры показывает, что в сети присутствует накопление, которое указывает на наличие проблемы проектирования. Накопление в данном случае может говорить о переполнении буфера.

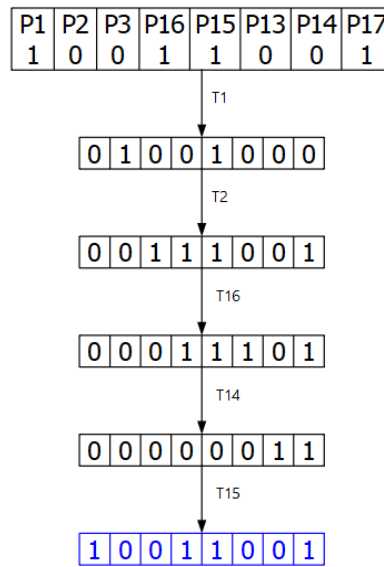


Рисунок 4 – ДДР построенная программой GPTN.

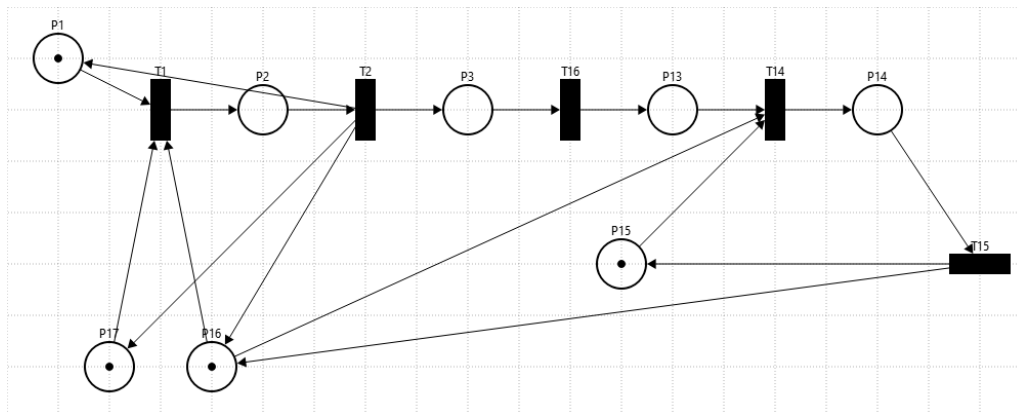


Рисунок 5 – СП модель с повышенной интенсивностью ввода, заданная в GPTN.

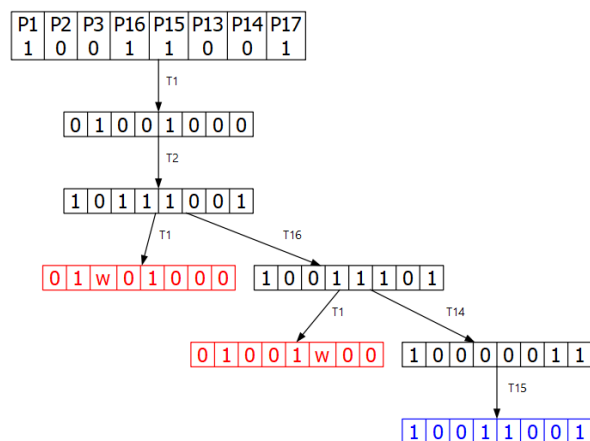


Рисунок 6 – ДДР построенная программой GPTN для модифицированной СП структуры.

Для устранения проблемы в работе модифицированной сети (рис. 5) введем дополнительную позицию P18, которая будет контролировать подачу данных в ВС (рис. 7). При этом позиция P18 означает, что данные выведены из ВС и готовы к вводу новых данных.

В результате построения ДДР (рис. 8) для исправленной СП структуры убеждаемся в корректном функционировании СП, то есть отсутствуют тупиковые ситуации и бесконечно размеченные позиции.

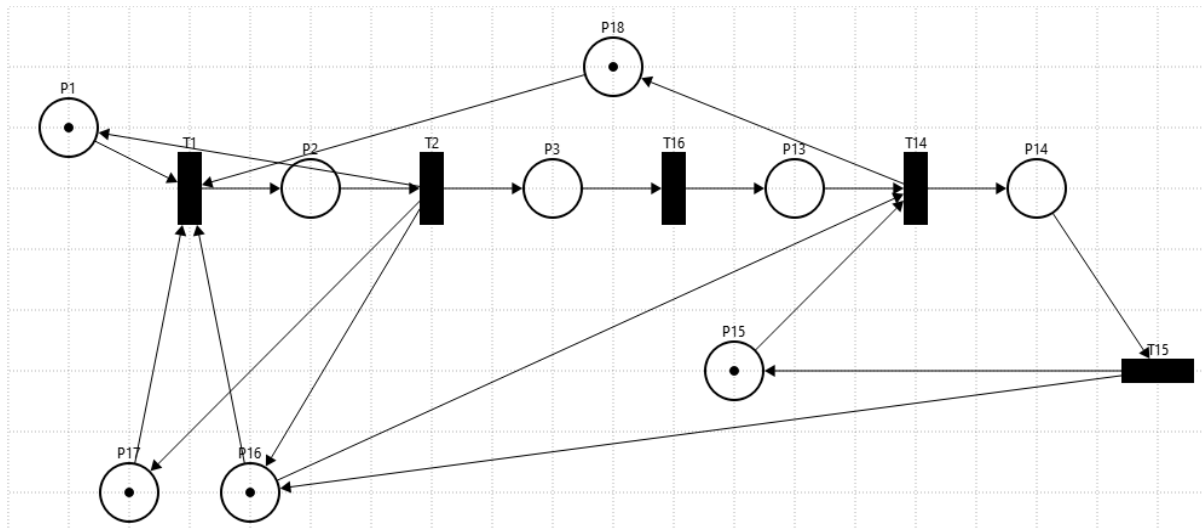


Рисунок 7 – Исправленная СП структура.

Построив новую модель вычислительной структуры в терминах СП, проанализируем работу иерархического перехода (рис. 9) с помощью дерева достижимых разметок (рис. 10). Стоит обратить внимание на наличие позиции P1000, она необходима, чтобы входной переход мог выполняться только после активации последнего перехода, то есть по принципу работы иерархического перехода. Такой подход обусловлен отсутствием поддержки иерархических переходов в GPTN, то есть данная структура (рис. 9) является отдельной моделью СП.

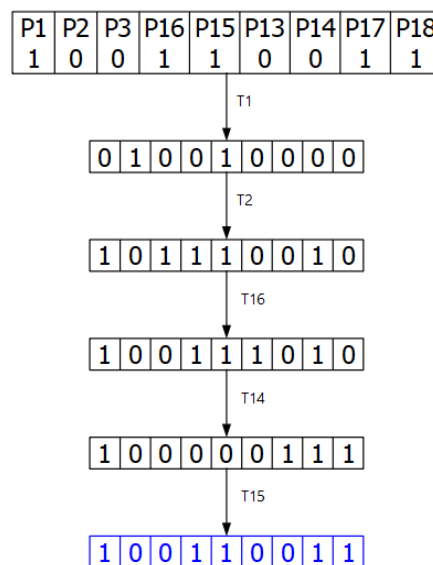


Рисунок 8 – ДДР исправленной СП структуры.

ДДР иерархического перехода (рис. 10) показывает, что он корректно функционирует, что в целом говорит о корректности функционирования всей СП структуры.

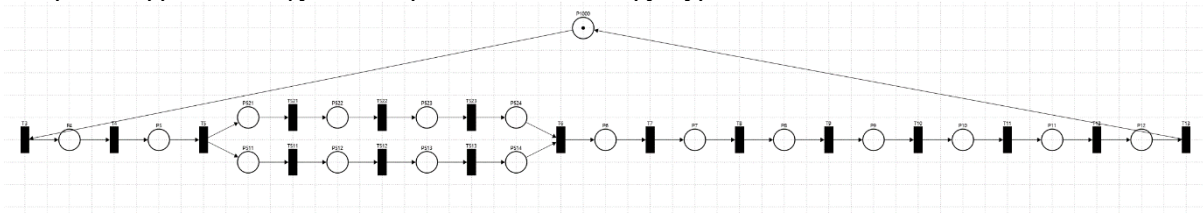


Рисунок 9 – Структура иерархического перехода.

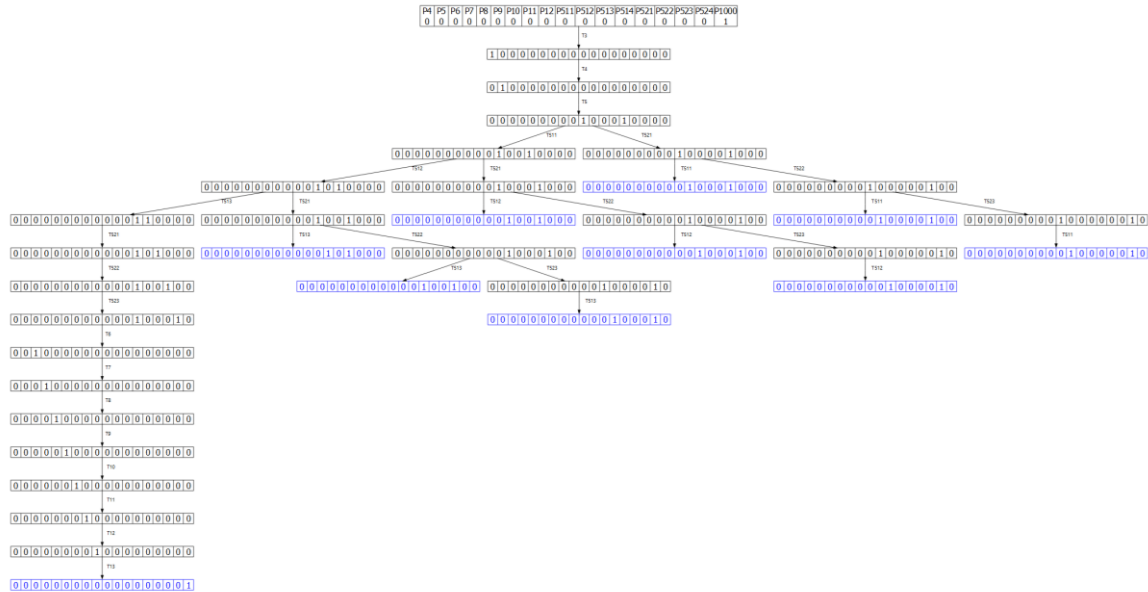


Рисунок 10 – ДПР иерархического перехода.

После модифицирования и исправления, новая комплексная вычислительная структура имеет вид, представленный на рисунке (рис. 11). В новой структуре появляется обратная связь от ВС3 к управлению вводом в многопроцессорную систему.

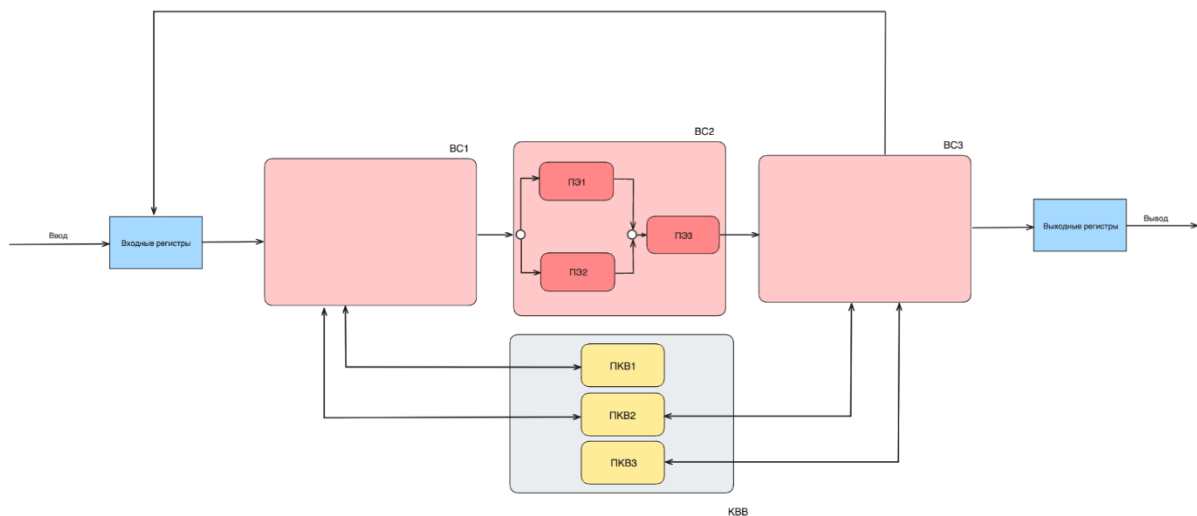


Рисунок 11 – Модифицированная структура ВС.

Также уделим внимание еще одной возможности программного комплекса GPTN, а именно симуляция модели сети Петри. Модуль симуляции с помощью построения временного графа позволяет оценить загруженность каждого перехода в сети Петри. Можно наглядно увидеть временной граф модифицированной СП-модели (рис. 12), который показывает моменты срабатывания переходов и моменты их простоя.

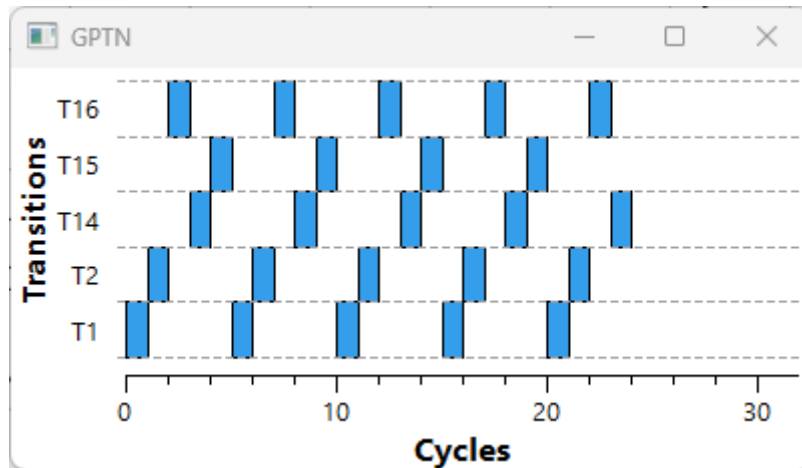


Рисунок 12 – Временной граф модифицированной СП-модели вычислительной структуры.

#### Заключение

В данной статье рассмотрены некоторые возможности разрабатываемого программного комплекса GPTN [15] на примере анализа и модифицирования комплексной вычислительной структуры. По ходу статьи программный комплекс GPTN использовался как вспомогательное средство анализа влияния модификаций на исходную СП структуру, что позволило выявить нестабильное поведение в функционировании сети после незначительного изменения её структуры. Опираясь на это знание, было найдено решение появившейся проблемы и снова воспользовавшись GPTN получилось подтвердить правильность найденного исправления.

В рамках данной статьи использованы основные возможности программного комплекса GPTN: графическое представление, моделирование некоторой вычислительной структуры в терминах СП, построение дерева достижимых разметок и работа с иерархическими переходами. Данная статья показывает, что программный комплекс GPTN уже помогает решать задачи анализа сложных вычислительных структур, при том, что была затронута только часть его возможностей.

#### Литература

1. Котов В. Е. Сети Петри. М.: Наука, 1984. 160 с.
2. Питерсон Дж. Теория сетей Петри и моделирование систем. М.: Мир, 1984. 264 с.
3. Мараховский В. Б., Розенблюм Л. Я., Яковлев А. В. Моделирование параллельных процессов. Сети Петри. Курс для системных архитекторов, программистов, системных аналитиков, проектировщиков сложных систем управления. СПб.: Профессиональная литература, АйТи-Подготовка, 2014. 400 с.
4. J. T. Butler, T. Sasao, "A set partition number system", Australasian Journal of Combinatorics, vol. 65, no. 2, pp. 152–169, 2016.
5. B. I. Freitas, J. C. Basilio, "Online Fault Diagnosis of Discrete Event Systems Modeled by Labeled Petri Nets Using Labeled Priority Petri Nets", IFAC-PapersOnLine, vol. 55, pp. 329-336, 2022.
6. A. S. Staines, "Concurrency and Petri Net Models", International Journal of Circuits, Systems and Signal Processing, vol. 16, pp. 852–858, 2022.
7. Kurt J., Susanna D., Maciej K. Transactions on Petri Nets and Other Models of Concurrency IV (Lecture Notes in Computer Science). Springer-Verlag. 2011. Vol. 6550. 225 с.
8. Kurt J., Susanna D., Jetty K. Transactions on Petri Nets and Other Models of Concurrency V (Lecture Notes in Computer Science). Springer-Verlag. 2012. Vol. 6900. 293 с.
9. Kurt J., Jetty K., Giuliana F. and etc. Transactions on Petri Nets and Other Models of Concurrency VI (Lecture Notes in Computer Science). Springer-Verlag. 2012. Vol. 7400. 365 p.
10. "Petri Nets World: Online Services for the International Petri Nets Community". Petri Nets World. <https://www2.informatik.uni-hamburg.de/TGI/PetriNets/index.php> (доступ 08 Мая 2024)
11. Кулагин В. П., Дубинин В. Н. Структурный анализ сетей Петри // Информационные технологии. 2016. Т. 22, № 1. С. 3–13.
12. Лескин А. А., Мальцев П. А., Спиридонов А. М. Сети Петри в моделировании и управлении. Л.: Наука, 1989. 135 с.
13. Анкундинов Г. И. Синтез структуры сложных объектов. Логико-комбинаторный подход. Л.: ЛГУ, 1986. 258 с.
14. Скорняков Л. А. Элементы теории структур. М.: Наука, 1982. 149 с.
15. "pulsarnet/GPTN". GPTN. <https://github.com/pulsarnet/GPTN> (доступ 08 Мая 2024)



16. “CPN Tools – A tool for editing, simulating, and analyzing Colored Petri nets”. CPN Tools. <https://cpntools.org/> (доступ 08 Мая 2024)
17. “CPN IDE – Based on Access/CPN”. CPN IDE. <https://cpnide.org/>
18. “Platform Independent Petri net Editor 2”. Pipe. <https://pipe2.sourceforge.net/> (доступ 08 Мая 2024)

## АНАЛИЗ НЕКОТОРЫХ РЕШЕНИЙ ПО ДЕКОДИРОВАНИЮ С ПРОВЕРКОЙ НА ЧЕТНОСТЬ С НИЗКОЙ ПЛОТНОСТЬЮ И ИХ РЕАЛИЗАЦИЙ НА ПЛИС

Бунина Л.В., Аминев Д.А., Данилов Р.А., Аверьянов П.Д.  
РТУ МИРЭА

Раскрыт математический аппарат принципа кодирования с проверкой на четность с низкой плотностью (LDPC). Определены этапы модифицированного алгоритма декодирования с использованием операций суммирования произведений. Представлены результаты компиляции и моделирования проекта ПЛИС для 24-битного декодера, проведена оценка задействованных этим проектом ресурсов микросхемы. Исследован проект LDPC декодера для макетной платы ICE40 фирмы Lattice Semiconductor, отмечены его основные особенности. Проведена сборка исходных модулей проекта на языке Verilog, выявлены его ограничения и недостатки.

Ключевые слова: код проверки четности низкой плотности, код исправления ошибок, ПЛИС, Modelsim; Xilinx, Verilog.

Analysis of some low-density parity-check decoding solutions and their FPGA implementations. Aminev D.A., Danilov R.A., Averyanov P.D., Bunina L.V. RTU MIREA.

The mathematical apparatus of the principle of low-density parity check (LDPC) coding is disclosed. The stages of a modified decoding algorithm using product summation operations are defined. The results of compilation and simulation of the FPGA project for a 24-bit decoder are presented, and the chip resources involved in this project are assessed. The LDPC decoder project for the ICE40 development board from Lattice Semiconductor has been studied, and its main features have been noted. The initial modules of the project were assembled in the Verilog language, its limitations and disadvantages were identified.

Key words: low density parity code, error correction code, FPGA, Modelsim; Xilinx, Verilog.

### Введение

Коды проверки на четность с низкой плотностью (LDPC), предложенные Р.Г. Галлагером в 1962 году [1], изначально не были востребованы, поскольку аппаратное обеспечение в то время не соответствовало требованиям, необходимым для процесса кодирования. С увеличением мощности компьютеров и развитием соответствующих теорий, таких как алгоритм распространения убеждений, коды LDPC были заново открыты Маккеем и Нилом в 1996 году [2]. Коды LDPC представляют собой линейные блочные коды, которые можно обозначить как  $(n, k)$  или  $(n, wc, wr)$ , где  $n$  – длина кодового слова,  $k$  – длина битов сообщения,  $wc$  – вес столбца (т. е. количество ненулевых элементов в столбце матрицы контроля четности), а  $wr$  – вес строки (т. е. количество ненулевых элементов в строке матрицы контроля четности).

Коды LDPC с проверкой на четность представлены матрицей с проверкой на четность  $H$ , где  $H$  – двоичная матрица, которая должна удовлетворять условию:

$$cHT = 0$$

где  $c$  – кодовое слово,  $HT$  – транспонированная матрица четности.

$$H = \begin{matrix} \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{matrix} & \begin{pmatrix} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 & V_9 & V_{10} & V_{11} & V_{12} & V_{13} & V_{14} & V_{15} & V_{16} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \end{pmatrix}$$

$H$  – разреженная матрица (т. е. количество «1» намного меньше, чем количество «0»). Именно разреженность  $H$  гарантирует низкую вычислительную сложность.