ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ТЕХНОЛОГИЙ ПЕРЕДАЧИ И ОБРАБОТКИ ИНФОРМАЦИИ И СИГНАЛОВ

УДК 621.391

УСКОРЕНИЕ ТЕНЗОРНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ ОСТАТОЧНЫХ КЛАССОВ

Червяков Н.И., Ляхов П.А., Ионисян А.С., Оразаев А.Р.
Северо-Кавказский федеральный университет, Ставрополь, РФ
E-mail: ljahov@mail.ru

Основным научно-практическим барьером для широкого распространения методов машинного обучения является высокая вычислительная сложность тензорных операций, используемых в них. Мы предлагаем метод реализации тензорных вычислений в системе остаточных классов с использованием табличной арифметики для модульных операций шириной до 8 бит включительно. Экспериментальное моделирование предложенного метода на FPGA Xilinx Spartan6 xc6slx9 показало, что он может быть использован для быстрой организации вычислений при реализации таблиц на блоках памяти BRAM. Предложенный подход позволяет ускорить вычисления в два раза, по сравнению с вычислениями в двоичной системе счисления, что может быть использовано для создания аппаратных ускорителей тензорных вычислений на практике.

Ключевые слова: *тензорные вычисления, система остаточных классов, FPGA, табличная арифметика*

Введение

Интенсивное развитие вычислительной техники и методов хранения больших объемов информации в облачных средах привело к серьезному прогрессу в области разработки методов машинного обучения. Одним из наиболее перспективных подходов в области искусственного интеллекта являются глубокие нейронные сети, которые уже показывают впечатляющие результаты в распознавании речи [1], изображений [2] и настольных играх типа шахмат [3]. Базовым элементом глубоких нейронных сетей является искусственный нейрон, математическая модель которого состоит в вычислении нелинейной функции от скалярного произведения входного сигнала на вектор весовых коэффициентов, что является простейшим примером тензорной операции. Искусственные нейроны объединяются в слои, которые, в свою очередь, организуются в последовательную структуру таким образом, что выходной сигнал предыдущего слоя подается на вход последующего слоя. Размерность пространства сигнала, передаваемого между слоями глубокой нейронной сети, может варьироваться в зависимости от решаемой задачи.

Основным научно-практическим барьером для широкого распространения глубоких нейронных сетей является высокая вычислительная сложность тензорных операций, используемых в них. Например, сеть AlexNet, разработанная для распознавания цветных изображений размером 224×224 пикселя, содержит 650 тысяч нейронов, 60 миллионов настраиваемых параметров, 630 миллионов связей для передачи данных между нейронами и обучалась в течение недели с использованием двух GPU Nvidia [4]. Другая сеть, разработанная для решения той же задачи, GoogLeNet, требует выполнения около 1,5 млрд арифметических действий при обработке одного изображения [5]. Реализация таких нейронных сетей на современных CPU и GPU является малоэффективной с точки зрения аппаратных, временных и энергетических затрат. Это делает задачу поиска альтернативных путей решения данной проблемы весьма актуальной.

Одним из возможных решений задачи оптимизации тензорных вычислений для глубоких нейронных сетей является реализация вычислительных алгоритмов на современных сверхбольших интегральных схемах FPGA [6] и ASIC [7]. Базовым блоком при реализации тензорных алгоритмов типа скалярного произведения или свертки является умножитель с накоплением. Одним из примеров универсальной реализации такого устройства является сигнальный процессор DSP48A1, разработанный компанией Xilinx, одним из ведущих производителей FPGA на сегодняшний день [8]. Другим примером специализированной разработки, оптимизированной под тензорные вычисления в глубоких нейронных сетях, является тензорный процессор TPU Google, объединяющий в одном устройстве $64 \cdot 10^3$ умножителей с накоплением [9]. При этом практически все усилия исследователей в области оптимизации тензорных вычислений направлены на

разработку новых устройств, функционирующих в традиционной двоичной арифметике.

Авторы статьи предлагают альтернативный подход к решению данной проблемы, основанный на арифметике системы остаточных классов (СОК). Идея использования СОК состоит в преобразовании вычислительного диапазона системы в прямую сумму конечных колец по модулям малой величины. Такой подход позволяет выполнять арифметические операции сложения, вычитания и умножения параллельно и без взаимодействия между вычислительными каналами, при этом малый размер модулей СОК позволяет существенно уменьшить аппаратные и временные затраты на реализацию арифметико-логических устройств. Известные на сегодняшний день архитектуры устройств, оптимизирующих вычисления в глубоких нейронных сетях и использующих СОК, построены в виде комбинационных схем [10–12]. В данной работе мы продемонстрируем эффективность использования табличной памяти в СОК для ускорения тензорных вычислений на FPGA.

Тензорные вычисления в системе остаточных классов

Рассмотрим тензор как полилинейную форму от n векторных аргументов

$$\varphi = \varphi(x_1, x_2, ..., x_n) = \\ = \sum_{i_1}\sum_{i_2}...\sum_{i_n} a_{i_1 i_2 ... i_n} x_{i_1} x_{i_2} ... x_{i_n}. \quad (1)$$

Основными тензорными операциями, используемыми в глубоких нейронных сетях, являются скалярное произведение, являющееся тензором валентности 2, и свертка

$$I_f(x, y, k) = \\ = \sum_{i=-t}^{t}\sum_{j=-t}^{t}\sum_{z=0}^{D-1} W_{i+t, j+t, z, k} I(x+i, y+j, z) \quad (2)$$

трехмерных данных, например, цветного изображения $I(x, y, z)$ с маской W тензора, описывающего слой нейронов. Тензорные операции, определяемые (1), (2), требуют выполнения только арифметических операций сложения и умножения, которые могут быть эффективно реализованы в СОК. Математической основой СОК является представление кольца вычетов по модулю M в виде прямой суммы колец вычетов по попарно взаимно простым модулям

$$m_1, m_2, ..., m_n, \ \text{НОД}(m_i, m_j) = 1, \ i \neq j: \\ Z_M = Z_{m_1} \oplus Z_{m_2} \oplus ... \oplus Z_{m_n}, \quad (3)$$

причем $m_1 m_2 ... m_n = M$ принято называть динамическим диапазоном СОК. Такое представление обосновывается китайской теоремой об остатках, устанавливающей взаимно однозначное соответствие между числами $X \in Z_M$ и векторами

$$(x_1, x_2, ..., x_n) = (X \bmod m_1, \\ X \bmod m_2, ..., X \bmod m_n) \quad (4)$$

по формуле

$$X = \left| \sum_{i=1}^{n} \left| M_i^{-1} \right|_{m_i} M_i x_i \right|_M, \quad (5)$$

где $M_i = \dfrac{M}{m_i}$ и $\left| M_i^{-1} \right|_{m_i}$ означает мультипликативный обратный элемент для M_i по модулю m_i, $i = 1, 2, \ldots, n$. Практический смысл представления, описываемого (3), (4), состоит в гомоморфизме арифметических операций: числам $(X + Y) \in Z_M$ и $(XY) \in Z_M$ соответствуют векторы

$$((x_1 + y_1) \bmod m_1, (x_2 + y_2) \bmod m_2, ... \\ ..., (x_n + y_n) \bmod m_n) = \\ = ((X + Y) \bmod m_1, (X + Y) \bmod m_2, ... \\ ..., (X + Y) \bmod m_n) \quad (6)$$

и

$$((x_1 y_1) \bmod m_1, (x_2 y_2) \bmod m_2, ... \\ ..., (x_n y_n) \bmod m_n) = \\ = ((XY) \bmod m_1, (XY) \bmod m_2, ... \\ ..., (XY) \bmod m_n). \quad (7)$$

При выборе модулей m_i достаточно малыми можно реализовать операции $(x_n + y_n) \bmod m_n$ $(x_n y_n) \bmod m_n$ в табличной форме.

Пример. Пусть $m = 5$. Двоичное представление элементов кольца вычетов Z_5 требует 3 бит информации. Таблицы сложения и умножения в кольце Z_5 чисел, представленных в двоичной форме, имеют следующий вид:

X+Y mod 5		Y				
		000	001	010	011	100
X	000	000	001	010	011	100
	001	001	010	011	100	000
	010	010	011	100	000	001
	011	011	100	000	001	010
	100	100	000	001	010	011

XY mod 5		Y				
		000	001	010	011	100
X	000	000	000	000	000	000
	001	000	001	010	011	100
	010	000	010	100	001	011
	011	000	011	001	100	010
	100	000	100	011	010	001

Таблица. Аппаратные ресурсы FPGA Spartan6 xc6slx9

Компонент FPGA	Количество
SLICE LUTs	5720
18 Кбит память BRAM	32
PLL модули	1
DSP48A1 модули	16

Вычислим в кольце Z_5 выражение $2\cdot 3+4$, которое представляет собой операцию умножения с накоплением. Из таблицы умножения по модулю 5 находим: $2\cdot 3 = 010_2 \cdot 011_2 = 001_2 = 1$.

Из таблицы сложения по модулю 5 находим: $1+4 = 001_2 + 100_2 = 000_2 = 0$. Проверка показывает, что $2\cdot 3+4 \equiv 0 \pmod 5$.

Приведенный пример демонстрирует быстроту и удобство выполнения арифметических операций со сложностью $O(1)$ ценой экстенсивного использования памяти. Для хранения таблицы умножения по модулю m с разрядностью $b = \lceil \log_2 m \rceil$ бит требуется $m^2 b$ бит памяти. С учетом того, что для сложения требуется таблица такого же размера, полные затраты памяти на реализацию арифметики по модулю m составят $2m^2 b$ бит памяти. Например, затраты памяти для реализации табличных вычислений по 6-битным модулям составят от 13 068 до 49 152 бит памяти, а для реализации табличных вычислений по 8-битным модулям составят от 266 256 до 1 048 576 бит памяти. Дальнейшее увеличение разрядности модулей приведет к резкому росту затрат памяти и может оказаться нецелесообразным.

Рассмотрим множество попарно взаимно простых 6-битных модулей {64, 63, 61, 59, 55, 53, 47, 43, 41, 37}, из которых можно составить СОК с диапазоном $129\,685\,918\,863\,695\,040 \approx 2^{56,8}$ для представления 56-битных чисел. Такой диапазон достаточен для реализации большинства современных глубоких нейронных сетей [13], при этом попытка составить таблицы сложения и умножения по модулю такого диапазона потребует использовать $\approx 2^{120}$ бит памяти, что на нынешнем уровне развития технологий вычислительной техники представляется невозможным.

Поэтому для реализации арифметических действий в таком диапазоне необходимо использовать комбинационные устройства сложения и умножения, имеющие сложности времени вычисления порядка $O(b)$ и $O(b^2)$ соответственно. Применение же предложенного подхода на основе СОК позволяет уменьшить каждую из этих сложностей до $O(1)$ за счет примерно 500 Кбит памяти, что окажется особенно ощутимым при выполнении большого количества операций умножения с накоплением в тензорных вычислениях.

Экспериментальное моделирование тензорных вычислений в СОК

Экспериментальная проверка предложенного метода проводилась на основе моделирования устройств, реализующих тензорные вычисления, с использованием FPGA Xilinx Spartan6 xc6slx9, установленной в FPGA Development board ALINX AX309. Данное устройство обладает ресурсами, представленными в таблице.

Было разработано устройство для проверки возможности размещения и эффективной работы 20 BRAM-блоков ПЗУ объемом 4096 6-битных ячеек каждая. Испытания показали, что ресурсов FPGA-микросхемы Spartan6 xc6slx9 достаточно для хранения 20 BRAM-блоков емкостью 4096 6-битовых ячеек, работающих в режиме ПЗУ. Для моделирования вычислений по формуле (7) были выбраны задача выполнения умножения целых неотрицательных чисел в СОК по модулю 31 и умножение 6-битных чисел в двоичной системе счисления без использования методов акселерации на основе блоков DSP48A1 с вычислением результата по модулю 64 простым взятием 6 младших бит результата вычислений. Экспериментальное моделирование показало двухкратное превосходство умножения в СОК над результатом в двоичной системе счисления. Время минимального прохождения электрического импульса через схему составило 6,119 нс, что соответствует максимальной частоте работы устройства 163 МГц. Внешний вид разработанного устройства показан на рисунке.

Моделирование операции свертки по формуле (2) проводилось методом замера скорости выполнения операции, содержащей от 900 до 3100 слагаемых, каждое из которых представляет собой произведение двух 56-битных двоичных чисел. Каждое двоичное 56-битное число было представлено в виде 10 остатков от деления на набор взаимно простых оснований СОК {64, 63, 61, 59, 55, 53, 47, 43, 41, 37} по формуле (4). Генерация входных аргументов для операции свертки производилась 10-ю аппаратными 16-битными LFSR-генераторами псевдослучайных чисел. Время минимального прохождения электрического импульса через схему составило 6,384 нс, что соответствует максимальной частоте работы устройства 156 МГц.

Обобщая результаты моделирования, можно сделать вывод о двукратном превосходстве скорости тензорных вычислений в СОК по сравнению с вычислениями в двоичной системе счисления. Данный факт может быть использован при разработке аппаратных ускорителей на основе FPGA для глубоких нейронных сетей.

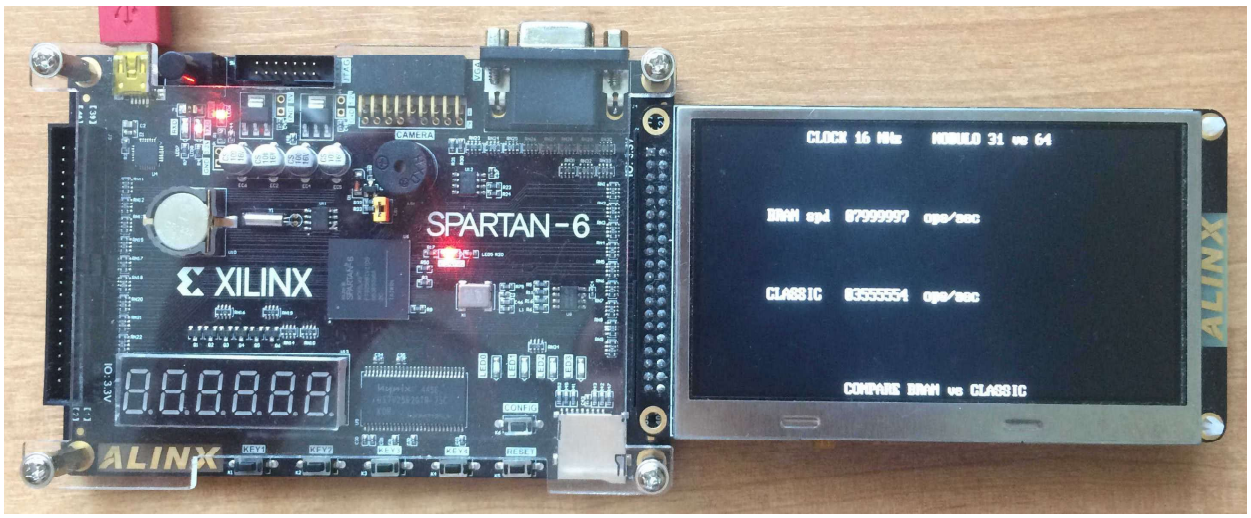

Рисунок. Внешний вид разработанного устройства ускорения тензорных вычислений на базе FPGA Development board ALINX AX309

Заключение

Мы предложили метод реализации тензорных вычислений в системе остаточных классов с использованием табличной арифметики для модульных операций шириной до 8 бит включительно. Экспериментальное моделирование предложенного метода на FPGA Xilinx Spartan6 xc6slx9 показало, что он может быть использован для быстрой организации вычислений при реализации таблиц на блоках памяти BRAM. Моделирование показало, что предложенный подход позволяет ускорить вычисления в два раза по сравнению с вычислениями в двоичной системе счисления, что может быть использовано для создания аппаратных ускорителей тензорных вычислений на практике.

Интересным направлением дальнейших исследований, на наш взгляд, является разработка методов оптимизации расходования памяти при реализации табличных вычислений в СОК, что может способствовать снижению ресурсных затрат при их реализации.

Благодарности

Работа выполнена при финансовой поддержке базовой части государственного задания (№2.6035.2017/БЧ), РФФИ (проекты №18-07-00109 А, №19-07-00130 А и №18-37-20059 мол-а-вед), Совета по грантам Президента РФ (проект СП-2245.2018.5).

Литература

1. Tu Y., Du J., Lee C. Speech enhancement based on teacher–student deep learning using improved speech presence probability for noise-robust speech recognition // IEEE/ACM Transactions on Audio, Speech and Language Processing. 2019. Vol. 27. № 12. P. 2080–2091.
2. Horror image recognition based on context-aware multi-instance learning / B. Li [et al.] // IEEE Transactions on Image Processing. 2015. Vol. 24. № 12. P. 5193–5205.
3. Mastering the game of go without human knowledge / D. Silver [et al.] // Nature. 2017. Vol. 550. № 7676. P. 354.
4. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. P. 1097–1105.
5. Going deeper with convolutions / C. Szegedy [et al.] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. P. 1–9.
6. Efficient network construction through structural plasticity / X. Du [et al.] // IEEE Journal on Emerging and Selected Topics in Circuits and Systems. 2019. Vol. 9. № 3. P. 453–464.
7. UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision / J. Lee [et al.] // IEEE Journal of Solid-State Circuits. 2019. Vol. 54. № 1. P. 173–185.
8. Spartan-6 FPGA DSP48A1 Slice User Guide. URL: https://www.xilinx.com/support/documentation/user_guides/ug389.pdf (дата обращения: 21.11.2019).
9. In-datacenter performance analysis of a tensor processing unit / N.P. Jouppi [et al.] // 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA). 2017. P. 1–12.
10. Hardware implementation of a convolutional neural network using calculations in the residue number system / N.I. Chervyakov [et al.] // Computer Optics. 2019. Vol. 43. № 5. P. 857–868.

11. Area-efficient FPGA implementation of minimalistic convolutional neural network using residue number system / N.I. Chervyakov [et al.] // 2018 23rd Conference of Open Innovations Association (FRUCT), Bologna. 2018. P. 112–118.
12. Nakahara H., Sasao T.A. High-speed low-power deep neural network on an FPGA based on the nested RNS: Applied to an object detector // 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence. 2018. P. 1–5.
13. Efficient processing of deep neural networks: A tutorial and survey / V. Sze [et al.] // Proceedings of the IEEE. 2017. Vol. 105. № 12. P. 2295–2329.

Получено 10.11.2019

Червяков Николай Иванович, заслуженный деятель науки РФ, д.т.н., профессор, заведующий кафедрой прикладной математики и математического моделирования (ПМ и ММ) Северо-Кавказского федерального университета (СКФУ). 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1. Тел. +7 865 235-48-61. E-mail: k-fmf-primath@stavsu.ru

Ляхов Павел Алексеевич, к.ф-м.н., доцент кафедры ПМ и ММ СКФУ. 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1. Тел. +7 962 028-72-14. E-mail: ljahov@mail.ru

Ионисян Андрей Сергеевич, к.ф-м.н., доцент кафедры ПМ и ММ СКФУ. 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1. Тел. +7 918 762-57-62. E-mail: asion@mail.ru

Оразаев Анзор Русланович, аспирант СКФУ. 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1. Тел. +7 928 982-45-10. E-mail: anz.orazaev95@gmail.com

ACCELERATION OF TENSOR COMPUTATIONS USING THE RESIDUAL CLASS SYSTEM

Chervyakov N.I., Lyakhov P.A., Ionisyan A.S., Orazaev A.R.
North-Caucasus Federal University, Stavropol, Russian Federation
E-mail: ljahov@mail.ru

The main scientific and practical barrier to the widespread dissemination of machine learning methods is the high computational complexity of tensor operations used in them. We propose a method for implementing tensor computations in a system of residual classes using table arithmetic for modular operations up to 8 bits wide, inclusive. Experimental modeling of the proposed method on FPGA Xilinx Spartan6 xc6slx9 showed that this method can be used to quickly organize computations when implementing tables on BRAM memory blocks. Modeling showed that the proposed approach allows us to accelerate the computations by a factor of two, compared with computations in the binary number system, which can be used to create hardware accelerators of tensor computations in practice.

Keywords: *tensor computations, system of residual classes, FPGA, table arithmetic*

DOI: 10.18469/ikt.2019.17.4.01

Chervyakov Nikolay Ivanovich, North-Caucasus Federal University, 1, Pushkin Street, Stavropol, 355000, Russian Federation; Head of Department of Applied Mathematics and Mathematical Modeling, Doctor of Technical Sciences, Professor. Tel. +7 865 235-48-61. E-mail: k-fmf-primath@stavsu.ru

Lyakhov Pavel Alekseyevich, North-Caucasus Federal University, 1, Pushkin Street, Stavropol, 355000, Russian Federation; Assistant Professor, Department of Applied Mathematics and Mathematical Modeling, PhD. Tel. +7 962 028-72-14. E-mail: ljahov@mail.ru

Ionisyan Andrey Sergeevich, North-Caucasus Federal University, 1, Pushkin Street, Stavropol, 355000, Russian Federation; Assistant Professor, Department of Applied Mathematics and Mathematical Modeling, PhD. Tel. +7 918 762-57-62. E-mail: asion@mail.ru

Orazaev Anzor Ruslanovich, North-Caucasus Federal University, 1, Pushkin Street, Stavropol, 355000, Russian Federation; PhD student, B.Sc. Tel. +7 928 928-45-10. E-mail: anz.orazaev95@gmail.com

References

1. Tu Y., Du J., Lee C. Speech enhancement based on teacher–student deep learning using improved speech presence probability for noise-robust speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2019, vol. 27, no. 12, pp. 2080–2091.
2. Li B. et al. Horror image recognition based on context-aware multi-instance learning. *IEEE Transactions on Image Processing*, 2015, vol. 24, no. 12, pp. 5193–5205.
3. Silver D. et al. Mastering the game of go without human knowledge. *Nature*, 2017, vol. 550, no. 7676, p. 354.
4. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012, pp. 1097–1105.
5. Szegedy C. et al. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
6. Du X., Li Z., Ma Y., Cao Y. Efficient network construction through structural plasticity. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2019, vol. 9, no. 3, pp. 453–464.
7. Lee J. et al. UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision. *IEEE Journal of Solid-State Circuits*, 2019, vol. 54, no. 1, pp. 173–185.
8. Spartan-6 FPGA DSP48A1 Slice User Guide. Available at: https://www.xilinx.com/support/documentation/user_guides/ug389.pdf (accessed 21.11.2019).
9. Jouppi N.P. et al. In-datacenter performance analysis of a tensor processing unit. *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 1–12.
10. Chervyakov N.I. et al. Hardware implementation of a convolutional neural network using calculations in the residue number system. *Computer Optics*, 2019, vol. 43, no. 5, pp. 857–868.
11. Chervyakov N.I. et al. Area-efficient FPGA implementation of minimalistic convolutional neural network using residue number system. *2018 23rd Conference of Open Innovations Association (FRUCT)*, 2018, pp. 112–118.
12. Nakahara H., Sasao T.A. High-speed low-power deep neural network on an FPGA based on the nested RNS: Applied to an object detector. *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.
13. Sze V. et al. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 2017, vol. 105, no. 12, pp. 2295–2329.

Received 10.11.2019

УДК 621.396.98

ЦИФРОВОЕ МОДЕЛИРОВАНИЕ МНОГОЛУЧЕВОГО КАНАЛА СВЯЗИ

Мишин Д.В., Тяжев А.И.
Поволжский государственный университет телекоммуникаций и информатики, Самара, РФ
E-mail: mishin@psati.ru, tyagev@psati.ru

Рассматриваются цифровые математические модели многолучевого канала радиосвязи. В рамках общей гауссовской модели радиоканала описано четырехпараметрическое распределение модуля и фазы комплексного коэффициента передачи радиоканала. Канал с рассеянием характеризуется дискретной многолучевостью, поэтому комплексный коэффициент передачи такого канала представляется в виде суммы конечного числа слагаемых с флуктуирующими коэффициентами передачи и задержками. Сигнал на выходе канала связи представлен через квадратурные компоненты, в которых учтены флуктуации коэффициентов передачи и задержек в канале связи. Эти компоненты в каналах образуются суммированием большого числа слагаемых, когда выполняются требования центральной предельной теоремы теории вероятностей, поэтому их можно считать независимыми нестационарными гауссовскими процессами. Приведены формулы, которые определяют методику моделирования сигнала на выходе многолучевого канала связи. Показано, что, используя четырехпараметрическое распределение ампли-