УДК 512.627.4, 511.288

DOI: 10.21779/2542-0321-2021-36-2-31–38

С.В. Шалагин, И.Д. Глазков

Сложность реализации нелинейных полиномов на массивах ПЛИС класса FPGA

Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ; Россия, 420111, г. Казань, ул. Карла Маркса, 10; sshalagin@mail.ru

В статье определен размер массива ПЛИС класса FPGA, требуемый для вычисления заданной полиномиальной функции над полем Галуа на распределенных вычислительных системах, как существующих, так и перспективных. При решении различных задач из области обработки массивов данных в реальном масштабе времени широко применяются распределенные вычисления. Под распределенными вычислениями будем понимать вычисления, выполняемые параллельно, с сохранением промежуточных результатов. При реализации распределенных вычислений требуется обеспечить примерно равное время задержки функционирования вычислителей (процессоров, ядер).

В работе показано, что нелинейный полином, определенный над полем Галуа, может быть реализован при использовании распределенных вычислений в архитектуре ПЛИС/FPGA, а также что вычисление нелинейного полинома может быть сведено к однотипным операциям, выполняемым над полем Галуа заданной размерности и реализуемым на вычислителях булевых функций от заданного количества переменных. Показано, что время задержки функционирования конвейера при вычислении нелинейного полинома постоянно и не зависит от количества переменных указанного полинома. Распределенные вычисления реализованы за счет наличия во внутренней структуре ПЛИС/FPGA реконфигурируемых элементов различного назначения; как вычислителей булевых функций, так и запоминающих элементов. Показано, что для вычисления нелинейного полинома от большого количества переменных применимо множество корпусов, образующих массив ПЛИС/FPGA. Внутри каждого из корпусов реализуем нелинейный полином от меньшего количества переменных, участвующий в вычислении значения исходного нелинейного полинома. Расчет размерности данного массива производится с учетом коэффициентов использования соответствующих ресурсов ПЛИС/FPGA, а также количества переменных указанных функций.

Ключевые слова: *полиномиальная функция, распределенная вычислительная система, ПЛИС.*

Введение

При реализации широкого класса систем обработки цифровой информации в реальном масштабе времени применимы распределенные вычисления: параллельное и/или конвейерное выполнение одних и тех же операций над потоком данных, которые реализуемы на многопроцессорных вычислительных системах (МВС) [1]. Вычислительными узлами МВС являются микропроцессоры (МП) как общего, так и специального назначения, реализуемые на сверхбольших интегральных схемах, в частности на

программируемых логических интегральных схемах (ПЛИС) класса FPGA [2]. Применение МВС с программируемой архитектурой, элементами которых служат ПЛИС класса FPGA [3], возможно в качестве бортовых систем и/или встроенных систем, реализующих различные устройства с высокой степенью надежности в разные периоды времени. В данном случае в качестве МП могут выступать аппаратные IP-ядра внутри ПЛИС/FPGA – генераторы булевых функций от заданного количества переменных внутри конфигурируемых логических блоков [2; 4; 5]. На основе указанных IP-ядер может быть реализована широкая номенклатура вычислительных устройств различного назначения [6–15].

В работе даны оценки для количества ресурсов определенной ПЛИС/FPGA, требуемых для распределенного вычисления нелинейной полиномиальной функции над полем Галуа определенной степени от заданного количества переменных. Получены оценки временной и аппаратной сложности распределенного вычисления в архитектуре ПЛИС/FPGA нелинейной полиномиальной функции (или нелинейного полинома) от m переменных (НПФ (m)), определенной над полем Галуа вида $GF(2^k)$.

Нелинейная полиномиальная функция

Введем в рассмотрение НПФ (m) над $GF(2^k)$ [16]

$$f(x_1, ..., x_m) = \sum_{i_1=0}^{w} ... \sum_{i_m=0}^{w} a_{i_1 ... i_m} x_1^{i_1} ... \; x_m^{i_m} , \qquad (1)$$

где $a_{i_1 ... i_m}, x_1^{i_1}, ..., x_m^{i_m} \in GF(2^k)$, $i_j = \overline{0, w}$, $j = \overline{1, m}$, $w = 2^k - 1$, символом $\sum$ обозначена операция вычисления поразрядной суммы по модулю два. Выражения $a_{i_1 ... i_m} x_1^{i_1} ... \; x_m^{i_m}$ в (1) определены как элементарные полиномы (ЭП), вычисление которых в общем случае требует выполнения m операций умножения элементов $GF(2^k)$ (обозначим данную операцию как $\otimes$). Наличие в (1) констант вместо коэффициентов $a_{i_1 ... i_m}$ позволяет уменьшить оценки сложности вычисления на одну $\otimes$ для ЭП. Кроме того, если внутри заданного ЭП существует Z_{ML} переменных таких, что $(i_j = 0) \vee (i_j = w)$: $Z_{ML} \in [1, m]$, то сложность его вычисления уменьшается на Z_{ML} $\otimes$.

Введем

Определение. Элементарный полином в (1), включающий Z_{ML} переменных таких, что $(i_j = 0) \vee (i_j = w)$: $Z_{ML} \in [1, m]$, имеет степень $m - Z_{ML}$.

Вычисление НПФ(m) требует дополнительно сложить $\left(2^k\right)^m - Z_{XR}$ значений ЭП, для чего требуется $\left(2^k\right)^m - 1 - Z_{XR}$ операций сложения элементов $GF(2^k)$ (обозначим данную операцию как $\oplus$). Величина Z_{XR} определяет количество ЭП, значение которых тождественно равно нулю при любых наборах переменных $x_1, ..., x_m$. Определим переменные $P_{XR}(d)$ как количество ЭП, тождественно равные нулю, для которых

$Z_{ML} = m - d$, $d = \overline{1, m}$, причем $\sum_{d=1}^{m} P_{XR}(d) = Z_{XR}$. В соответствии с изложенным выше справедливо

Утверждение 1. Количество операций $\otimes$ и $\oplus$ при вычислении полинома вида (1) от m переменных равно $\sum_{d=1}^{m}\left((d-1)\cdot w^d \cdot \left(C_m^d - P_{XR}(d)\right)\right)$ и $\left(2^k\right)^m - 1 - Z_{XR}$ соответственно.

Выражение (1) для m переменных перепишем как многочлен от q переменных, $q > m$:

$$f\left(x_1, \ ..., \ x_q\right) = \sum_{i_{m+1}=0}^{w} ... \sum_{i_q=0}^{w} f_{i_{m+1}...i_q}\left(x_1, \ ..., x_m\right) \cdot x_{m+1}^{i_{m+1}} \cdot ... \cdot x_q^{i_q}, \ w = 2^k - 1. \quad (2)$$

Наличие в (2) вместо $f_{i_{m+1}...i_q}\left(x_1, \ ..., \ x_m\right)$ постоянных значений позволяет существенно уменьшить оценки сложности вычисления НПФ(q). Количество таких постоянных значений определим как $\overline{Z}_{XR}$. Согласно (2) получение значения НПФ(q) реализуем путем распределенного вычисления, которое включает в себя получение $\left(2^k\right)^{q-m}$ значений НПФ (m) вида $f_{i_{m+1}...i_q}\left(x_1, ..., x_m\right)$, $i_j = \overline{0, 2^k - 1}$, $j = \overline{m+1, q}$, на основе которых вычисляется НПФ(q). Для вычисления многочлена (2) без учета сложности вычисления множества из $\left(2^k\right)^{q-m}$ значений НПФ(m) требуются $\sum_{d=1}^{q-m}\left(d \cdot w^d \cdot \left(C_{q-m}^d - \overline{P}_{XR}(d)\right)\right)$ $\otimes$ и $\left(2^k\right)^{q-m} - 1 - \overline{Z}_{XR}$ $\oplus$. Величина $\overline{P}_{XR}(d)$ определена для (2) по аналогии с (1). Справедливо

Утверждение 2. Количество операций $\otimes$ и $\oplus$ при вычислении полинома вида (2) от q переменных, $q > m$ равно соответственно

$$\left(\left(2^k\right)^{q-m} - \overline{Z}_{XR}\right) \cdot \sum_{d=1}^{m}\left((d-1)\cdot w^d \cdot \left(C_m^d - P_{XR}(d)\right)\right) + \sum_{d=1}^{q-m}\left(d \cdot w^d \cdot \left(C_{q-m}^d - \overline{P}_{XR}(d)\right)\right) \quad \text{и}$$

$$\left(\left(2^k\right)^{q-m} - \overline{Z}_{XR}\right) \cdot \left(\left(2^k\right)^m - 1 - Z_{XR}\right) + \left(2^k\right)^{q-m} - 1 - \overline{Z}_{XR}.$$

Согласно утверждениям 1 и 2 определено количество операций $\otimes$ и $\oplus$ над двумя элементами поля $GF(2^k)$. Их количество экспоненциально зависит от таких параметров, как количество переменных НПФ(m) и порядок $GF(2^k)$, и линейно зависит от наличия ЭП, тождественно равных нулю при любых входных наборах переменных НПФ(m).

Вычисление нелинейных полиномиальных функций на ПЛИС/FPGA

ПЛИС/FPGA включают в свой состав МП, реализующие генераторы булевых функций от $2 \cdot k$ переменных – GBF($2 \cdot k$), а также параллельные регистры на k разрядов – $RG(k)$, $k = 2, 3, \ldots$ [4; 5]. Нижняя оценка общего количества ПЛИС класса FPGA, необходимого для реализации НПФ(q), определена как:

$$Q_{\text{FPGA}} \geq \left\lceil \max\left(\frac{N_{\text{GBF}(2k)}}{k_{\text{GBF}} \cdot Q_{\text{GBF}(2k)}} \quad \frac{N_{\text{D}}}{k_{\text{D}} \cdot Q_{\text{D}}} \quad \frac{N_{\text{IOB}}}{k_{\text{IOB}} \cdot Q_{\text{IOB}}} \right) \right\rceil, \tag{3}$$

где $N_{\text{GBF}(2k)}$, N_{D} и N_{IOB} – общее количество GBF($2 \cdot k$), D-триггеров и блоков ввода-вывода (БВВ), требуемых для реализации заданной НПФ(q); ПЛИС/FPGA заданного типа включает в свой состав $Q_{\text{GBF}(2k)}$ GBF($2 \cdot k$), Q_{D} D-триггеров и Q_{IOB} БВВ, соответственно. Согласно [11] значения k_{GBF}, k_{D} и k_{IOB} обычно принимают равными 0,5–0,7. Справедливо

Утверждение 3. Нижняя оценка количества программируемых логических интегральных схем класса FPGA, требуемого для реализации НПФ(q) вида (2), определена согласно (3).

Найдем оценки сложности вычисления НПФ вида (1) и (2) на ПЛИС/FPGA по количеству $N_{\text{GBF}(2k)}$, N_{D} и N_{IOB} в соответствии с [7–11]. В ПЛИС/FPGA реализуема произвольная функция от $2 \cdot k$ переменных посредством GBF($2 \cdot k$).

Для НПФ вида (1) при использовании k GBF($2 \cdot k$) возможно вычисление множества ЭП степени один и два от двух заданных переменных, а для реализации ЭП степени d, $d = \overline{3, m}$ требуется $k\left(d-2\right)$ GBF($2 \cdot k$). Что касается операций сложения элементов $GF(2^k)$ по модулю два, то k GBF($2 \cdot k$) позволяют реализовать сложение $2 \cdot k$ элементов $GF(2^k)$. Согласно утверждению 1

$$N^{(1)}_{\text{GBF}(2k)} = k \cdot \left(\sum_{d=2}^{m} \left(\left(d-1\right) \cdot w^d \cdot \left(C_m^d - P_{XR}(d) \right) \right) + \left\lceil \frac{\left(2^k\right)^m - 1 - Z_{XR}}{2k-1} \right\rceil \right). \tag{4}$$

На вход устройства для вычисления НПФ(m) поступают m переменных, а с выхода снимается одна переменная над $GF(2^k)$. Каждая из переменных представлена k разрядами, что при условии реализации данного устройства на одном корпусе ПЛИС/FPGA требует

$$N^{(1)}_{\text{IOB}} = k\left(m+1\right). \tag{5}$$

Результаты, полученные в k GBF($2 \cdot k$), а также поступившие в БВВ, сохраняются в $RG(k)$ или в k D-триггерах, что обеспечивает конвейерную обработку процесса вычисления значений НПФ как вида (1), так и вида (2)

$$N^{(1)}_{\text{D}} = N^{(1)}_{\text{GBF}(2k)} + N^{(1)}_{\text{IOB}}. \tag{6}$$

Для НПФ вида (2) при использовании k GBF($2\cdot k$) возможно вычисление множества ЭП степени один от двух заданных переменных, а для реализации ЭП степени d, $d=\overline{2,m}$ требуется $k\left(d-1\right)$ GBF($2\cdot k$). Сумма значений ЭП рассчитывается по аналогии с НПФ вида (1). Согласно утверждению 2

$$N_{\mathrm{GBF}(2k)}^{(2)}=\left(\left(2^k\right)^{q-m}-\overline{Z}_{XR}\right)N_{\mathrm{GBF}(2k)}^{(1)}+$$

$$+k\cdot\left(\sum_{d=1}^{q-m}\left(d\cdot w^d\cdot\left(C_{q-m}^d-\overline{P}_{XR}(d)\right)\right)+\left\lceil\frac{\left(2^k\right)^{q-m}-1-\overline{Z}_{XR}}{2k-1}\right\rceil\right). \tag{7}$$

По аналогии с (5) определим

$$N_{\mathrm{IOB}}^{(2)}=k\left(q+1\right). \tag{8}$$

$$N_{\mathrm{D}}^{(2)}=N_{\mathrm{GBF}(2k)}^{(2)}+N_{\mathrm{IOB}}^{(2)}. \tag{9}$$

В формулах (4) и (7) значения $P_{XR}(d)$, Z_{XR}, $\overline{P}_{XR}(d)$ и $\overline{Z}_{XR}$ рассчитываются для заданной НПФ как вида (1), так и вида (2). При увеличении количества переменных НПФ (1) и (2), m и q значения $N_{\mathrm{GBF}(2k)}$, N_{D} растут экспоненциально, а N_{IOB} – линейно. Имеют место

Утверждение 4. Реализуемость НПФ вида (1) от m переменных на заданной ПЛИС/FPGA определяется выполнением неравенства (3) для $Q_{\mathrm{FPGA}}=1$ при вычислении значений $N_{\mathrm{GBF}(2k)}=N_{\mathrm{GBF}(2k)}^{(1)}$, $N_{\mathrm{IOB}}=N_{\mathrm{IOB}}^{(1)}$ и $N_{\mathrm{D}}=N_{\mathrm{D}}^{(1)}$ согласно формулам (4), (5) и (6).

Утверждение 5. Реализуемость НПФ вида (2) от q переменных на заданной ПЛИС/FPGA определяется выполнением неравенства (3) для $Q_{\mathrm{FPGA}}=1$ при вычислении значений $N_{\mathrm{GBF}(2k)}=N_{\mathrm{GBF}(2k)}^{(2)}$, $N_{\mathrm{IOB}}=N_{\mathrm{IOB}}^{(2)}$ и $N_{\mathrm{D}}=N_{\mathrm{D}}^{(2)}$ согласно формулам (7), (8) и (9).

Следует отметить, что современные ПЛИС/FPGA включают в свой состав большое количество GBF($2\cdot k$) и D-триггеров, порядка 1 млн [4; 5]. Поэтому, несмотря на экспоненциальный рост значений $N_{\mathrm{GBF}(2k)}$ и N_{D}, внутри одного корпуса возможна реализация НПФ вида (1) и (2) для достаточно большого количества переменных. Данное обстоятельство позволяет обеспечить конвейерную реализацию произвольного нелинейного отображения одного множества в другое. При этом оценка времени задержки функционирования проектируемого устройства, реализующего НПФ как вида (1), так и вида (2) составляет:

$$T=t_D+t_{IC}+\max\left(t_{in},t_{\mathrm{GBF}(2k)},t_{out}\right),$$

где t_D, t_{IC}, $t_{\mathrm{GBF}(2k)}$ – времена задержки функционирования D-триггеров, межсоединений и GBF($2\cdot k$) для заданной ПЛИС/FPGA; t_{in} и t_{out} – времена задержек БВВ, работа-

ющих на ввод и вывод информации из корпуса заданной ПЛИС/FPGA соответственно. Время задержки межсоединений t_{IC} рассчитывается для заданного устройства с применением специализированной САПР либо может быть определено как величина, составляющая не более 70 % от общего времени задержки функционирования [17]. Имеет место верхняя оценка времени задержки функционирования, которая для современных ПЛИС/FPGA будет существенно меньше за счет развитой системы межсоединений

$$T \le \frac{10}{3}\left(t_D + \max\left(t_{in}, t_{\mathrm{GBF}(2k)}, t_{out}\right)\right).$$

Заключение

В статье представлены формулы для вычисления количества ресурсов ПЛИС/FPGA, необходимых для реализации устройств распределенного вычисления нелинейных полиномов над $GF(2^k)$ в зависимости от заданного количества переменных. Теоретически обоснована количественная оценка количества переменных, для которых будет реализована заданная НПФ от определенного количества переменных на заданной ПЛИС класса FPGA. Показана допустимость конвейерного вычисления НПФ на ПЛИС/FPGA за счет возможности использования как логических ресурсов, так и ресурсов памяти данного типа аппаратных модулей. Получены оценки времени задержки функционирования вычисления заданной НПФ на ПЛИС определенного семейства. Полученные результаты позволяют определить оценки аппаратной и временной сложности реализации заданной НПФ на МВС с программируемой архитектурой как на существующих, так и на перспективных.

Литература

1. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 600 с.

2. *Кузелин М.О., Кнышев Д.А., Зотов В.Ю.* Современные семейства ПЛИС фирмы Xilinx: справочное пособие. – М.: Горячая линия – Телеком, 2004. – 440 с.

3. *Каляев И.А., Левин И.И., Семерников Е.А.* Семейство вычислительных систем с высокой реальной производительностью на основе ПЛИС // Вестник УГАТУ. – 2010. – № 5 (40). – С. 91–101.

4. FPGA Leadership across Multiple Process Nodes / Xilinx Inc. Cop. 2021. [Электронный ресурс]. – Режим доступа: https://www.xilinx.com/products/silicon-devices/fpga.html.

5. 7-Series Product Selection Guide / Xilinx Inc. Cop. 2014-2020. [Электронный ресурс]. – Режим доступа: https://www.xilinx.com/support/documentation/selection-guides/7-series-product-selection-guide.pdf.

6. *Zakharov V.M., Shalagin S.V.* Executing discrete orthogonal transformations based on computations on the Galois field in the FPGA architecture // International Siberian Conference on Control and Communications (SIBCON). – Moscow, 2016. – Pp. 1–4.

7. *Шалагин С.В.* Сложность вычисления нелинейных полиномиальных функций над полем $GF((2^2)^k)$ на ПЛИС/FPGA // АКТО-2014: сб. докл. Межд. научно-практ. конф. Т. II (19–21 ноября, Казань, 2014 г.). – Казань, 2014. – С. 661–664.

8. *Шалагин С.В.* Сложность вычисления нелинейной полиномиальной функции над полем Галуа вида GF(2^k) в базисе булевых функций от 2k переменных // Дискретные модели в теории управляющих систем: труды IX Межд. конф. (Москва и Подмосковье, 20–22 мая 2015). – М.: МАКС Пресс, 2015. – С. 264–266.

9. *Шалагин С.В.* Оценка сложности распределенного вычисления нелинейной полиномиальной функции над полем $GF(2^K)$ на многопроцессорной вычислительной системе // Новые информационные технологии и системы: сб. научн. статей XI Межд. научно-технич. конф. (г. Пенза, 25–27 ноября 2014 г.). – Пенза: Изд-во ПГУ, 2014 – С. 9–12.

10. *Шалагин С.В.* Реализация цифровых устройств в архитектуре ПЛИС/FPGA при использовании распределенных вычислений в полях Галуа. – Казань: Изд-во КНИТУ-КАИ, 2016. – 228 с.

11. *Шалагин С.В.* Вычисление нелинейных полиномиальных функций на распределенных вычислительных системах с программируемой архитектурой // Аналитическая механика, устойчивость и управление: труды XI Межд. Четаевской конф. Т. 4. Секция 4. Компьютерные технологии в науке, образовании, управлении производством (г. Казань, 13–17 июня 2017 г.). – Казань: КНИТУ-КАИ, 2017. – С. 243–247.

12. *Raikhlin V.A., Vershinin I.S., Gibadullin R.F. et all.* Reliable recognition of masked binary matrices. Connection to information security in map systems Lobachevskii J. Math. –2013. – № 34. – Pp. 319–325. – URL: https://doi.org/10.1134/S1995080213040112.

13. *Vershinin I.S., Gibadullin R.F., Pystogov S.V. et all*. Associative Steganography. Durability of Associative Protection of Information // Lobachevskii J. Math. – 2020, № 41. – Pp. 440–450. – URL: https://doi.org/10.1134/S1995080220030191.

14. *Захаров В.М., Шалагин С.В.* О развитии аппаратных средств статистического моделирования // Развитие вычислительной техники и ее программного обеспечения в России и странах бывшего СССР: история и перспективы, труды Третьей Межд. конф. (г. Казань, 13–17 октября 2014 г.). – Казань, 2014. – С. 103 – 108.

15. *Захаров В.М., Нурутдинов Ш.Р., Шалагин С.В.* Полиномиальное представление цепей Маркова над полем Галуа // Вестник Казанского государственного технического университета им. А.Н. Туполева. – 2001. – № 3. – С. 27–31.

16. *Лидл Р., Нидеррайтер Г.* Конечные поля: в 2 т. – М.: Мир, 1988.

17. *Шалагин С.В.* Экспериментальное исследование методики синтеза комбинационных схем на программируемых микросхемах класса FPGA // Микроэлектроника. – 2004. – Т. 33, № 1. – С. 56–67.

Поступила в редакцию 22 марта 2021 г.

UDC 512.627.4, 511.288

DOI: 10.21779/2542-0321-2021-36-2-31–38

The Complexity of Computing Nonlinear Polynomials on the PLD FPGA-Class Dimensions

S.V. Shalagin, I.D. Glazkov

Kazan National Research Technical University named after A.N. Tupolev – KAI; Russia, 420111, Kazan, Karl Marks st., 10; sshalagin@mail.ru

The size of the FPGA-class PLD array required for calculating a given polynomial function over the Galois field on distributed computing systems, both existing and prospective, is determined. Distributed computing is widely used in solving various tasks in the field of processing data arrays in real time scale. Under distributed computing, we will understand calculations performed in parallel, with the intermediate results preservation. When implementing distributed computing, it is necessary to provide approximately equal delay time for the operation of computers (processors, cores).

It is shown that the nonlinear polynomial defined over the Galois field can be implemented using distributed computing in the PLD/FPGA architecture. It is shown that the calculation of a nonlinear polynomial can be reduced to the same type of operations performed on the Galois field of a given dimension and implemented on calculators of Boolean functions on a given number of variables. It is shown that the delay time of the pipeline operation when calculating a nonlinear polynomial is constant and does not depend on the number of variables of the specified polynomial. Distributed computing is implemented due to the presence of reconfigurable elements for various purposes in the internal structure of the PLD/FPGA: both Boolean function calculators and memory elements. It is shown that to calculate a nonlinear polynomial from a large number of variables, a set of cases forming an PLD/FPGA array is applicable. Inside each of the cases, we implement a nonlinear polynomial from a smaller number of variables, which is involved in calculating the value of the original nonlinear polynomial. The dimension of this array is calculated taking in terms of the utilization coefficients of the corresponding PLD/FPGA resources, as well as the number of variables of the specified functions.

Keywords: *polynomial function, distributed computer systems, PLD.*

Received 22 March 2021