

ОЦЕНКА ВЫИГРЫША ВО ВРЕМЕНИ ОБРАБОТКИ ПРИ АППАРАТНО-ОРИЕНТИРОВАННОЙ КЛАССИФИКАЦИИ БИНАРНЫХ ОТНОШЕНИЙ ВЕРШИН ГРАФ-СХЕМ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

В статье рассмотрена задача аппаратно-ориентированной классификации бинарных отношений вершин граф-схем параллельных алгоритмов, возникающая при проектировании систем логического управления в базисе логических мультиконтроллеров, при автоматизированном распараллеливании вычислений и др. Программная реализация указанных действий является сравнительно медленной, ввиду чего с целью сокращения затрат времени на выполнение классификации отношений предложено выполнять ее с использованием специализированного аппаратно-ориентированного акселератора. В данной статье приведено краткое описание его структуры, базирующееся на использовании специализированной многопортовой памяти с матричной структурой, и специализированной схмотехнической реализации операции умножения битовых векторов. В ходе выполненной оценки затрат времени показано, что на матрицах малой плотности, к которым относятся рассматриваемые матрицы бинарных отношений, предложенное устройство на 1–2 порядка быстрее высоко оптимизированной программной реализации.

Классификация бинарных отношений, умножение матриц, проектирование специализированных вычислительных средств, граф-схемы параллельных алгоритмов.

The article considers the problem of hardware-oriented classification of binary relations of vertexes of the graph-schemes of parallel algorithms that arises in the design of logic control systems within the logic multicontrollers basis and automated parallelization. Software implementation of these actions is relatively slow, that is why in order to reduce the time required to perform the classification of relationships it is suggested to perform it with the use of specialized hardware-based accelerator. This article contains a brief description of its structure, based on the use of a dedicated multi-port memory with a matrix structure, and specialized circuitry implementing multiplication of bit vectors. In the course of assessing the cost of time it is shown that low-density matrices, which are considered a matrix of binary relations, the proposed device 1–2 orders of magnitude faster than highly optimized software implementation.

Classification of binary relations, matrices multiplication, special computing devices design, graph-schemes of parallel algorithms.

Введение.

В различных задачах, к которым относятся задачи проектирования однородных многомодульных мультисистем [5], [16], [18], [19], в том числе систем логического управления (СЛУ) в базисе логических мультиконтроллеров, задачи параллельной обработки информации [17] и пр., возникает необходимость в обработке и анализе параллельных алгоритмов, одной из форм представления которых является их оформление в виде взвешенных ориентированных графов специального вида, именуемых граф-схемами параллельных алгоритмов. В ряде задач (например, при поиске разбитий) возникает необходимость в классификации бинарных отношений вершин граф-схем параллельных алгоритмов, к которым относятся [13] отношения следования (ν), связи (ϕ), параллельности (ω) и альтернативы (ψ). Они обладают рядом особых свойств, на базе которых возможно их последовательное определение при программной реализации указанных действий. Однако при этом на построение матрицы отношений M_R затрачивается [7] существенное (до 20 %) количество вычислительного времени, которое можно сократить путем переноса данных операций с программного уровня на аппаратный в ходе разработки специализированного аппаратно-ориентированного акселератора [21], [23], [24], [25], структура которого жестко адаптирована к условиям решаемой задачи.

Основная часть.

Структурная схема разработанного устройства-акселератора приведена на рис. 1.

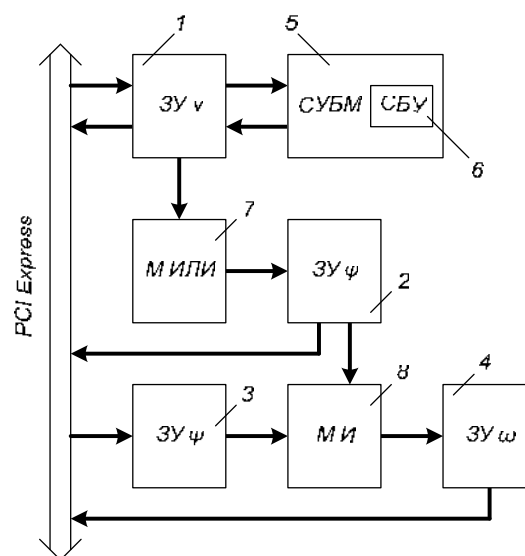


Рис. 1. Структурная схема устройства-акселератора для классификации бинарных отношений

Оно включает в своем составе запоминающие устройства (ЗУ) 1–4 для хранения бинарных отношений следования, связи, параллельности и альтер-

нативы соответственно, схему транзитивного замыкания (СТЗ) 5, схему битового умножения векторов (СБУ) 6, а также матрицы логических элементов ИЛИ 7 и И 8.

Запоминающие устройства подключены к шине PCI Express с целью загрузки исходных данных для построения матрицы отношений и выгрузки результата. В соответствии с рассмотренным выше алгоритмом на первом шаге работы устройства производится загрузка начальных значений для определения отношения следования. Вслед за этим в работу вступает схема умножения битовых матриц, которая с использованием схему умножения битовых векторов в соответствии с рассмотренным выше алгоритмом, базирующемся на алгоритме Флойда-Уоршалла, производит нахождение транзитивного замыкания отношения следования, после чего обновленная матрица отношения следования может быть выгружена в оперативную память. Затем с использованием матрицы элементов ИЛИ 7 возможно определение значений отношения связи, которое формируется в ЗУ 2, после чего может быть выгружено в оперативную память. Параллельно с определением значений следования и связи возможна загрузка отношения альтернативы, определяемого на программном уровне, в ЗУ 3. После определения матриц отношений связи и альтернативы с использованием матрицы элементов И возможно определение отношения параллельности, записываемого в ЗУ 4 и затем выгружаемого в оперативную память.

При программной обработке матрица M_R хранится в оперативной памяти в виде двумерного массива множеств (битовых векторов), элементы которого располагаются в памяти подряд. При аппаратно-ориентированной обработке данный способ хранения является неудобным, так как при каждом обращении требуется вычисление соответствующего адреса, что, в свою очередь, приводит к появлению в схеме дополнительных сумматоров и блоков умножения (последние могут быть заменены на сдвигатели при условии кратности N степени двойки), увеличивая аппаратную сложность, тепловыделение и

время задержки распространения сигнала. Выходом из положения является разработка специализированного двухкоординатного запоминающего устройства [23], ориентированного на хранение именно матричной информации, структурная схема ячейки которого приведена на рис. 2.

Условное обозначение однородной среды, базирующейся на объединении ячеек (рис. 2), приведено на рис. 3. По сравнению с однородной средой [7] предложенная ячейка упрощена за счет исключения элементов, реализующих ассоциативные операции, а среда имеет двухкоординатную адресацию вместо однокоординатной, несмотря на внешнее сходство.

Схемотехническая реализация операции булева умножения (СБУ) i -й строки на j -й столбец матрицы M_R приведена на рис. 4 [24]. В основе ее работы лежит принцип, согласно которому из ЗУ поочередно выбираются соответствующие элементы умножаемых векторов, после чего производится их побитовое умножение. В случае, если в результате умножения на одном из шагов работы схемы получается единичное значение, оно фиксируется в триггере 5 и формирует единичное значение признака готовности Θ на выходе элемента ИЛИ 10 и работа схемы досрочно прекращается. При использовании предложенной схемотехнической реализации выигрыш в скорости обработки достигается за счет использования специализированного запоминающего устройства, позволяющего адресацию отдельных бит информации и параллельное выполнение операций чтения и записи в различные ячейки с использованием различных портов (до 4 параллельно выполняемых операций в предложенной схеме). Также число итераций снижено по сравнению с программной реализацией за счет ранней проверки факта установления достижимости.

С использованием схемы битового умножения векторов представляется возможным реализовать схему умножения матрицы самой на себя [21] с целью транзитивного замыкания бинарного отношения (рис. 5).

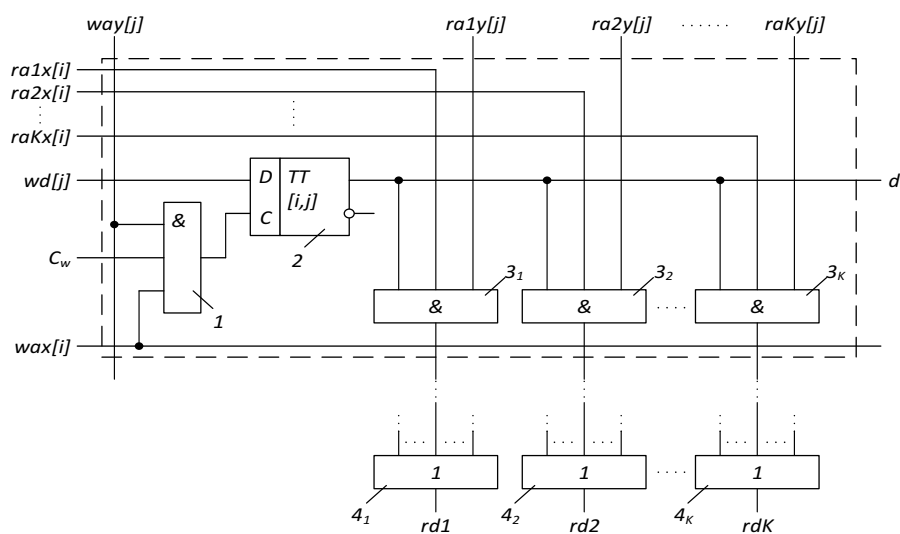


Рис. 2. Схема ячейки однородной запоминающей среды для хранения битовых признаков отношения следования

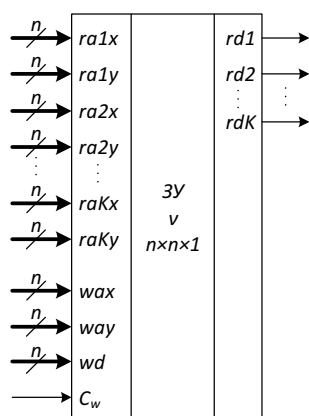


Рис. 3. Условное обозначение запоминающего устройства для хранения бинарного отношения следования

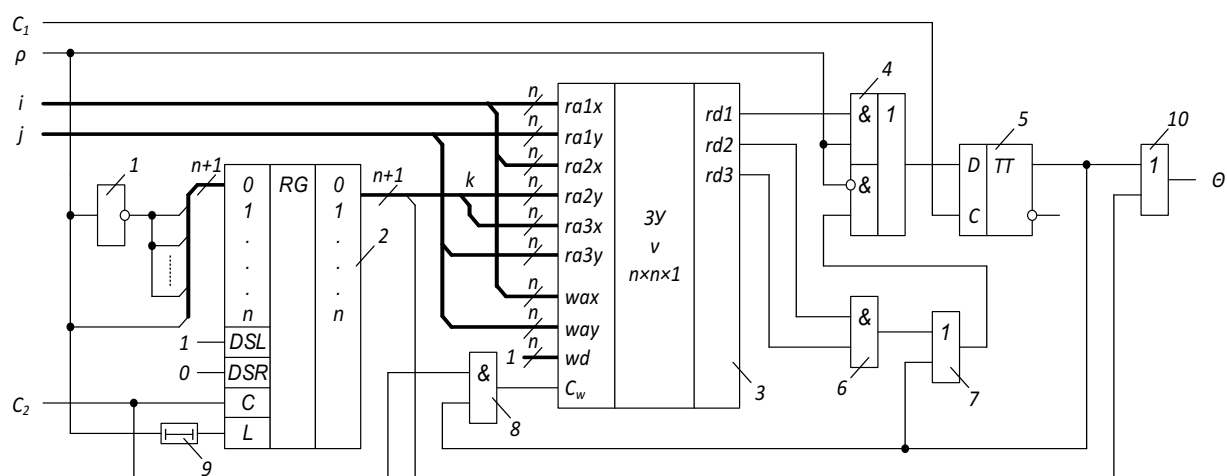


Рис. 4. Схемотехническая реализация операции булева умножения строки и столбца

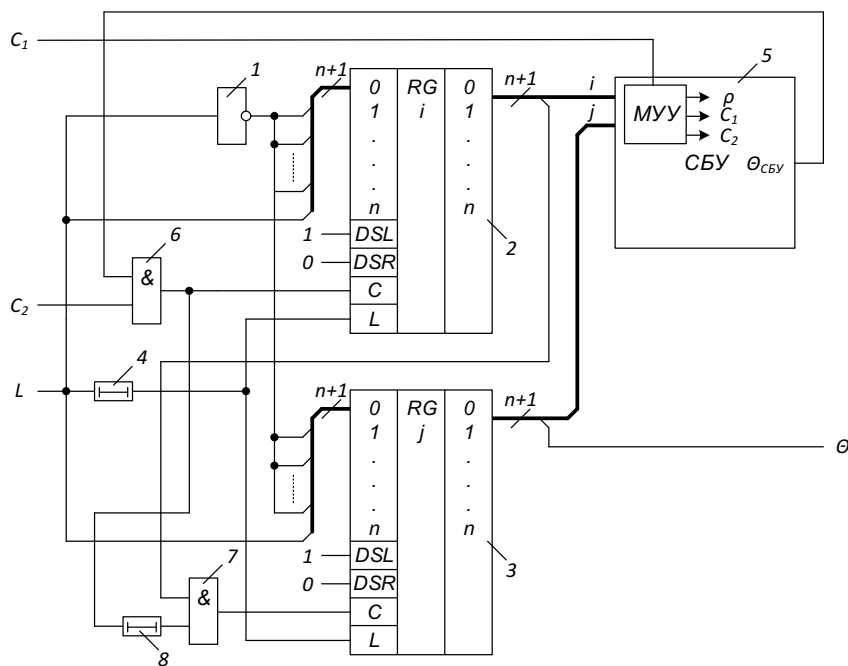


Рис. 5. Схемотехническая реализация операции умножения битовой матрицы самой на себя (схема транзитивного замыкания, СТЗ)

Она обеспечивает поочередную выборку строк и столбцов обрабатываемой бинарной матрицы (два внешних цикла в алгоритме Флойда-Уоршалла) с последующей их передачей на вход схемы СБУ для выполнения умножения битовых векторов.

Аппаратная реализация матриц элементов ИЛИ 7 и И 8 (рис. 1) тривиальна и включает в своем составе в простейшем случае N^2 соответствующих двух-входовых элементов. При этом допускается не хранить диагональные элементы (для ЗУ 2 и 3) и использовать лишь верхнюю треугольную подматрицу (для ЗУ 4) ввиду симметричности бинарного отношения параллельности.

Произведем оценку выигрыша во времени классификации бинарных отношений. Итоговое время обработки $T_{\text{акс}}$ с использованием предложенного специализированного вычислительного устройства можно определить как

$$T_{\text{акс}} = T_{\text{загр}} + T_{\text{обр}} + T_{\text{выгр}},$$

где $T_{\text{загр}}$ – время передачи данных из оперативной памяти (RAM) в память устройства-акселератора, $T_{\text{обр}}$ – время обработки, $T_{\text{выгр}}$ – время передачи данных из памяти устройства-акселератора в оперативную память. С учетом введенных обозначений получаемый выигрыш в общем случае можно определить как

$$\eta = \frac{T_{\text{прог}}}{T_{\text{акс}}},$$

где $T_{\text{прог}}$ – время выполнения преобразований с использованием программной реализации. Время передачи данных $T_{\text{пер}}$ между оперативной памятью и памятью акселератора можно определить как

$$T_{\text{пер}} = \frac{V}{P},$$

где V – объем передаваемых данных (байт), P – пропускная способность интерфейса (байт/с). В рассматриваемых условиях пропускная способность определяется как

$$P = \min \{P_{\text{PCI-E}}, P_{\text{RAM}}, P_{\text{акс}}\},$$

где $P_{\text{PCI-E}}$ – пропускная способность шины PCI Express [22], P_{RAM} – пропускная способность оперативной памяти, $P_{\text{акс}}$ – пропускная способность памяти акселератора. На современном этапе развития средств вычислительной техники в рассматриваемом случае бутылочным горлышком является пропускная способность интерфейса PCI Express (несмотря на выход новых и более скоростных версий стандарта, последней из которых на данный момент является версия 3.0), поэтому можно положить $P \approx P_{\text{PCI-E}}$.

С учетом результатов измерения реальной пропускной способности интерфейса PCI Express можно оценить затраты времени, необходимые для передачи одной матрицы бинарного отношения размером $N \times N$ (табл. 1).

Таблица 1

Затраты времени, необходимые для передачи матрицы бинарного отношения размером $N \times N$ через интерфейс PCI Express

Размер матрицы, $N \times N$	Занимаемый объем данных (с учетом битовой компрессии)	Время передачи с использованием интерфейса PCI Express (с учетом битовой компрессии)
128×128	16 КБ (2 КБ)	1,29 мкс (0,16 мкс)
256×256	64 КБ (8 КБ)	5,17 мкс (0,65 мкс)
512×512	256 КБ (32 КБ)	20,7 мкс (2,59 мкс)
1024×1024	1 МБ (128 КБ)	82,8 мкс (10,3 мкс)
2048×2048	4 МБ (512 КБ)	331 мкс (41,4 мкс)
4096×4096	16 МБ (2 МБ)	1,29 мс (166 мкс)
8192×8192	64 МБ (8 МБ)	5,16 мс (662 мкс)

С целью определения выигрыша от использования предложенного устройства-акселератора необходима оценка величины $T_{\text{прог}}$, для чего можно воспользоваться программной реализацией [6], [12] после выполнения алгоритмической оптимизации [16–18], работающей в составе среды PAE [4], [14], [15]. Результаты измерения времени определения отношения следования приведены в табл. 2.

Таблица 2

Результаты определения отношения следования при программной реализации операций обработки граф-схем параллельных алгоритмов с различным числом вершин

Размер матрицы, $N \times N$	Время программной обработки (CPU Intel Core i7 4770 @ 3,4 ГГц, ядро Haswell)
128×128	78,2 мс
256×256	349,1 мс
512×512	2,13 с
1024×1024	13,8 с
2048×2048	87,1 с
4096×4096	549 с
8192×8192	3457 с

Приведенные результаты наглядно демонстрируют стремительный рост затрат вычислительного времени с ростом размерности задачи, причиной чего являются как асимптотический рост числа шагов алгоритма $t \simeq O(n^3)$, так и исчерпание емкость кэш-памяти процессора.

Произведем оценку быстродействия аппаратно-ориентированного акселератора. При реализации операции записи в ЗУ (см. рис. 2) сигналу с входа данных необходимо переключить входящие в состав

D -триггера 2 инверторы (необходимое время – $2t_0$) по приходу синхросигнала, которому требуется пройти через трехходовой элемент И 1 (необходимое время – $2t_0$). Таким образом, общее время записи одной ячейки составляет $t_{\text{зп}} = 4t_0$. При реализации операции чтения сигнал с выхода D -триггера 2 проходит через один из трехходовых элементов И 3, на что затрачивается время $2t_0$, и через N^2 -входной элемент ИЛИ 4, на что, при реализации последнего с использованием пирамидальной схемы двухвходовых элементов, необходимо время $2\lceil \log_2 N \rceil t_0$. Таким образом, общее время чтения данных из одной ячейки составляет $t_{\text{чт}} = 2t_0 + 2\lceil \log_2 N \rceil t_0$.

На этапе инициализации схемы СБУ наиболее длинный путь, возникающий при прохождении сигнала через логические элементы, представлен элементами 1–2, а необходимое время инициализации составляет $t_{\text{СБУ}0} = 3t_0$. После чтения данные из ЗУ 3 через цепочку элементов 6–7–4 поступают в триггер 5, где фиксируются по приходу синхросигнала C_1 , на что необходимо время

$$t_{\text{СБУ}1} = \underbrace{2t_0 + 2\lceil \log_2 N \rceil t_0}_{\text{ЗУ 3}} + \underbrace{t_0}_{\text{И 6}} + \underbrace{t_0}_{\text{ИЛИ 7}} + \underbrace{2t_0}_{\text{К 4}} + \underbrace{2t_0}_{\text{Т 5}} = 8t_0 + 2\lceil \log_2 N \rceil t_0.$$

Далее обновленное значение записывается в ЗУ 3, на что требуется $t_{\text{СБУ}2} = \underbrace{t_0}_{\text{И 8}} + \underbrace{4t_0}_{\text{Т 3}} = 5t_0$. Таким образом, общее время работы схемы булева умножения битовых векторов составляет

$$t_{\text{СБУ}} = t_{\text{СБУ}0} + k(t_{\text{СБУ}1} + t_{\text{СБУ}2}) = (3 + 13k + 2\lceil \log_2 N \rceil k)t_0,$$

где $1 \leq k \leq N$ – число итераций работы схемы, зависящее от исходных данных.

На этапе инициализации работы схемы транзитивного замыкания бинарного отношения сигнал C_2 проходит через элемент И 6 за время t_0 , параллельно с этим сигнал L проходит через элемент НЕ 1, после чего исходные единичные значения фиксируются в регистрах 2 и 3 за время $2t_0$, на что необходимо время $t_{\text{СТ}30} = t_0 + 2t_0 = 3t_0$. Внутренний (вложенный) цикл работы схемы складывается из времени работы схемы СБУ, времени сдвига регистра 3, равного $\underbrace{t_0}_{\text{И 6}} + \underbrace{3t_0}_{\text{Р 2}} = 4t_0$, и составляет

$$t_{\text{СТ}31} = t_{\text{СБУ}} + 4t_0 = (3 + 13\bar{k} + 2\lceil \log_2 N \rceil \bar{k})t_0 + 4t_0 = (7 + 13\bar{k} + 2\lceil \log_2 N \rceil \bar{k})t_0.$$

где $\bar{k}$ – среднее число итераций работы схемы СБУ. Итерация внешнего (объемлющего) цикла работы схемы складывается из n итераций внутреннего цик-

ла и сдвига информации регистра 3, на что затрачивается

$$t_{\text{СТ}32} = nt_{\text{СТ}31} + 4t_0 = n(7 + 13\bar{k} + 2\lceil \log_2 N \rceil \bar{k})t_0 + 4t_0.$$

С учетом того, что внешний цикл выполняется n раз, время работы схемы составляет

$$\begin{aligned} t_{\text{СТ}3} &= t_{\text{СТ}30} + nt_{\text{СТ}32} = \\ &= 3t_0 + N(n(7 + 13\bar{k} + 2\lceil \log_2 N \rceil \bar{k})t_0 + 4t_0) = \\ &= (13N^2\bar{k} + 2N^2\lceil \log_2 N \rceil \bar{k} + 7N^2 + 4N)t_0. \end{aligned}$$

Оценка выигрыша во времени работы предложенного устройства с учетом работы всех элементов операционной части устройства (рис. 1) для матриц различной плотности (что влияет на среднее число итераций работы схемы СБУ $\bar{k}$) приведена на рис. 6.

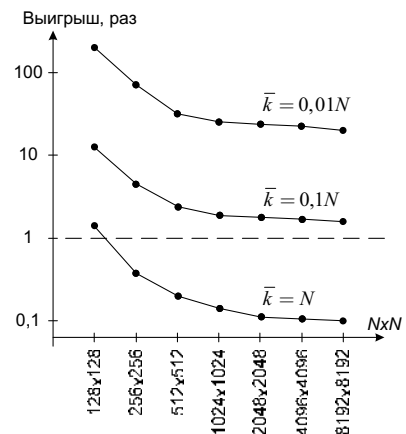


Рис. 6. Зависимость выигрыша во времени обработки от размера обрабатываемых матриц бинарных отношений при различном числе итераций работы схемы СБУ

Выводы.

Прежде всего, сопоставление времени обработки с временем передачи исходных данных (в данном случае, матрицы, содержащей исходные значения для определения значения следования) показывает, что время передачи занимает менее 1 % от общего времени обработки и может не учитываться при расчете выигрыша. При обработке матриц, для которых не происходит раннего прерывания работы схемы СБУ, использование предложенного устройства нецелесообразно, так как по сравнению с программной реализацией оно обеспечивает проигрыш во времени обработки до 6 раз при $N = 8196$. Транзитивное замыкание отношения следования для подобных матриц целесообразно находить с использованием классического умножения матриц с использованием SIMD-ориентированной [9], [10], CUDA-ориентированной [8], [11] программной или специализированной аппаратной реализации, ориентированной на использование матричных и/или систолических вычислительных структур [20]. При обработке матриц отношения следования для реальных граф-схем па-

параллельных алгоритмов наблюдается частая возможность досрочного прерывания работы схемы СБУ ($k \simeq 0,1 \div 0,01$), что позволяет эффективное использование предложенного устройства и обеспечивает выигрыш во времени обработки от нескольких десятков раз ($N=256 \div 8196$ при $\bar{k}=0,01N$, $N=128$ при $\bar{k}=0,1N$) до сотен раз ($N=128$ при $\bar{k}=0,01N$), подтверждая целесообразность его разработки.

Следует также отметить, что приведенные выше оценки получены для значения времени задержки распространения сигнала вентиля $t_0=1$ нс, по порядку величины совпадающего с ПЛИС-реализацией. При реализации устройства в заказном исполнении указанное время задержки может быть снижено приблизительно на порядок, что приблизительно на порядок увеличивает потенциальный теоретический выигрыш.

Литература

1. Ватутин, Э. И. Алгоритмическая оптимизация программной реализации метода параллельно-последовательной декомпозиции граф-схем параллельных алгоритмов / Э. И. Ватутин, В. С. Титов // Известия высших учебных заведений. Приборостроение. 2013. – Т. 56. – №6. – С. 23–29.
2. Ватутин, Э. И. Анализ узких мест программной реализации метода параллельно-последовательной декомпозиции граф-схем параллельных алгоритмов / Э. И. Ватутин // Распознавание – 2013. – Курск, 2013. – С. 235–237.
3. Ватутин, Э. И. Анализ эффективности и программная оптимизация методов синтеза разбиений параллельных алгоритмов логического управления в среде PAE / Э. И. Ватутин // Известия Юго-западного государственного университета. – Курск, 2012. – № 2. – Ч. 1. – С. 191–195.
4. Ватутин, Э. И. Визуальная среда синтеза разбиений параллельных алгоритмов логического управления / Э. И. Ватутин, И. В. Зотов // Свидетельство об официальной регистрации программы для ЭВМ № 2007613222 от 30.07.07.
5. Ватутин, Э. И. Комбинаторно-логические задачи синтеза разбиений параллельных алгоритмов логического управления при проектировании логических мультимикросхем / [Э. И. Ватутин и др.]. – Курск, 2010.
6. Ватутин, Э. И. Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов / Э. И. Ватутин, И. В. Зотов // Параллельные вычисления и задачи управления (РАСО'04). – М., 2004. – С. 884–917.
7. Ватутин, Э. И. Однородная среда электронной модели дерева для аппаратно-ориентированной обработки R-выражений / Э. И. Ватутин // Распознавание – 2008. Ч. 1. – Курск, 2008. – С. 90–92.
8. Ватутин, Э. И. Оценка реальной производительности современных видеокарт с поддержкой технологии CUDA в задаче умножения матриц / Э. И. Ватутин, И. А. Мартынов, В. С. Титов // Известия Юго-западного государственного университета. – 2014. – № 2. – С. 8–17.
9. Ватутин, Э. И. Оценка реальной производительности современных процессоров в задаче умножения матриц для однопоточной программной реализации с использованием расширения SSE (часть 1) / Э. И. Ватутин, В. С. Ти-

тов // Известия Юго-западного государственного университета. Принята к опубликованию.

10. Ватутин, Э. И. Оценка реальной производительности современных процессоров в задаче умножения матриц для однопоточной программной реализации с использованием расширения SSE (часть 2) / Э. И. Ватутин, В. С. Титов // Известия Юго-западного государственного университета. Принята к опубликованию.

11. Ватутин, Э. И. Оценка реальной производительности современных процессоров и видеокарт с поддержкой технологии CUDA в задаче умножения матриц / Э. И. Ватутин, И. А. Мартынов, В. С. Титов // CUDA альманах. – 2015. – Май. – С. 9–10.

12. Ватутин, Э. И. Параллельно-последовательный метод формирования субоптимальных разбиений параллельных управляющих алгоритмов / Э. И. Ватутин, И. В. Зотов // Свидетельство об официальной регистрации программы для ЭВМ №2005613091 от 28.11.05.

13. Ватутин, Э. И. Построение матрицы отношений в задаче оптимального разбиения параллельных управляющих алгоритмов / Э. И. Ватутин, И. В. Зотов // Известия Курского государственного технического университета. – Курск, 2004. – № 2. – С. 85–89.

14. Ватутин, Э. И. Программная система для нахождения разбиений параллельных алгоритмов логического управления / Э. И. Ватутин // Распознавание – 2005. – Курск, 2005. – С. 174–177.

15. Ватутин, Э. И. Программная система для построения разбиений параллельных управляющих алгоритмов / Э. И. Ватутин, И. В. Зотов // Идентификация систем и задачи управления (SICPRO'06). – М., 2006. – С. 2239–2250.

16. Ватутин, Э. И. Проектирование логических мультимикросхем. Синтез разбиений параллельных граф-схем алгоритмов / Э. И. Ватутин. – Saarbrücken, 2011.

17. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб., 2002.

18. Емельянов, С. Г. Архитектура параллельных логических мультимикросхем / С. Г. Емельянов, И. В. Зотов, В. С. Титов. – М., 2009.

19. Зотов, И. В. Организация и синтез микропрограммных мультимикросхем / [И. В. Зотов и др.]. – Курск, 1999.

20. Кун, С. Матричные процессоры на СБИС / С. Кун. – М., 1991.

21. Мартынов, И. А. Аппаратно-ориентированная реализация операции транзитивного замыкания бинарных отношений / И. А. Мартынов, Э. И. Ватутин, В. С. Титов // Распознавание – 2015. – Курск, 2015. – С. 244–247.

22. Мартынов, И. А. Измерение реальной пропускной способности шины PCI Express с использованием видеокарт с поддержкой технологии CUDA в качестве периферийных устройств / И. А. Мартынов, Э. И. Ватутин // Распознавание – 2015. – Курск, 2015. – С. 242–244.

23. Мартынов, И. А. Многопортовое матричное запоминающее устройство для хранения битовых признаков / И. А. Мартынов, Э. И. Ватутин, В. С. Титов // Медико-экологические информационные технологии – 2014. – Курск, 2014. – С. 141–145.

24. Наджаджра, М. Х. Схемотехническая реализация операции умножения битовых векторов при классификации бинарных отношений граф-схем параллельных алгоритмов / М. Х. Наджаджра, И. А. Мартынов, Э. И. Ватутин // Распознавание – 2015. – Курск, 2015. – С. 275–277.

25. Martynov, I. A. Hardware oriented classification of binary relations of graph-schemes of parallel algorithms / I. A. Martynov, E. I. Vatutin, V. S. Titov // Eighth World Conference on Intelligent Systems for Industrial Automation (WCIS – 2014). – Tashkent, 2014. – P. 70–73.