

УДК 004.421

В. С. Князьков, Т. В. Князькова

СИНТЕЗ АЛГОРИТМОВ МАССИВНЫХ ВЫЧИСЛЕНИЙ ДЛЯ МНОГОЯДЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР

В статье рассматривается подход к синтезу алгоритмов массивных вычислений на базе пространственной алгебры (S-алгебры), который ориентирован на создание оптимальных с точки зрения временной и пространственной сложности реализации алгоритмов массивной обработки данных в многоядерных процессорных пространствах. Предлагаемый подход к синтезу позволяет формально построить оптимальную с точки зрения выбранного критерия сложности схему обработки данных для той процессорной среды, в которой будет выполняться данный алгоритм. Концептуально такие многоядерные процессорные среды представляют массивы битовых процессорных элементов, связанных между собой битовыми каналами обмена данными через коммутационные элементы.

Приводится пример синтеза алгоритма решения задачи поиска максимального числа в массиве положительных целых чисел M разрядности из N элементов. Показана оценка временной сложности синтезируемых алгоритмов, причем оценку временной сложности реализации процедуры поиска максимального числа можно было получить из выражения общей оценки операций данного класса до разработки конкретного алгоритма.

Ключевые слова: массивные вычисления, многоядерные вычислительные структуры, синтез алгоритмов, оценка сложности.

На первом этапе разработки общего алгоритма решения конкретной задачи разработчиком произвольно на основе собственного опыта и интуиции определяется последовательность шагов вычислений. Предлагаемый подход к синтезу позволяет формально построить оптимальную с точки зрения выбранного критерия сложности схему обработки данных для той процессорной среды, в которой будет выполняться данный алгоритм. Концептуально такие многоядерные процессорные среды представляют массивы битовых процессорных

элементов, связанных между собой битовыми каналами обмена данными через коммутационные элементы.

Можно выделить следующие основные этапы синтеза алгоритмов рассматриваемого класса [3].

1. *Определяются входные, внутренние и выходные структуры данных*, используемые в качестве аргументов и результатов выполнения операций. В качестве метрического пространства определения элементов структур данных принимается пространство процессорной среды, в которой будут выполняться вычисления. Причем, каждый вычислительный модуль среды интерпретируется как простейший элемент метрического пространства. Выбираются схемы размещения элементов структур данных в метрическом пространстве и выполняется их размещение. Определяется метод и общая схема решения поставленной задачи на уровне математического описания процедур преобразования данных. Разрабатывается общая алгоритмическая схема вычислений, определяющая порядок и тип операций над данными. Выполняется проверка алгоритмической корректности данной схемы.

2. *Выполняется описание каждой операции над данными* как функции или системы функций S -алгебры в форме аналитических выражений. Выполняется минимизация каждой функции.

3. *Выполняется замена описаний операций* в модулях базовой схемы алгоритма на их описания в виде S -функций или систем S -функций. Выделяются последовательности операторов, в которых в качестве аргументов операций используются результаты предыдущих операций. Выполняется описание таких участков в виде S -функции, аргументами которых являются S -функции предыдущих операций. При возможности выполняется минимизация полученных выражений.

4. *При необходимости определяется временная и пространственная сложность* алгоритма по методике, рассмотренной в работах [4, 5, 6]. При этом:

- каждая операция ориентированного сдвига Q рассматривается как последовательность M операций обмена через коммутационную сеть сложности 1,
- каждая одноместная или многоместная операция S -преобразования элемента данных, – как однократная вычислительная операция в процессорном элементе сложности 1,
- каждая операция присваивания одной переменной значения другой переменной или значения результата вычислений S -функции от нескольких переменных – как однократная операция доступа сложности 1,
- каждому действию над переменной в процессорном элементе предшествует операция выборки переменной сложности 1.

В случае не соответствия заданному критерию сложности выполняется изменение п.1 и синтез алгоритма повторяется. Следует отметить особенности. Во-первых, приведенные в [4, 5, 6] теоретические положения позволяют дать общую оценку вычислительной сложности реализации алгоритма уже на этапе 1 рассматриваемой методики. В связи с этим такую оценку уместно предварительно выполнить. Во-вторых, критерий сложности может быть выбран некорректно на этапе постановки задачи и, соответственно, получить алгоритм с требуемыми характеристиками сложности принципиально не возможно. При соблюдении приведенных положений полученная в результате синтеза формализованная в виде записей S -функций схема алгоритма решения задачи обладает следующими особенностями:

- количество обменных, вычислительных и операций доступа к данным при реализации алгоритма в вычислительной среде будет минимальным, так как в процессе синтеза алгоритма получены минимальные формы операторных модулей алгоритма в виде минимизированных S -функций;
- полученные представления операторов алгоритма в виде S -функций являются эффективной формой для преобразования последовательного представления алгоритма в целом или отдельных его участков в параллельно-битовую форму: каждая простая S -импликанта каждой S -функции может быть вычислена независимо от других простых импликант этой же функции;

• полученные описания операций алгоритма в виде аналитических записей **S**-функций практически являются готовыми алгебраическими описаниями функционирования операционных устройств процессорного элемента и позволяют при необходимости выполнить их формальный технический синтез.

Для демонстрации приведенных положений ниже рассматривается пример синтеза алгоритма решения задачи поиска максимального числа в массиве положительных целых чисел **M** разрядности из **N** элементов.

Пусть имеется вычислительная среда из $(D \times D) = N$ элементов. В качестве схемы размещения элементов обрабатываемой структуры данных примем матричную разрядно – последовательную схему размещения (Рис. 1). В данном случае разряды элементов структуры данных размещаются построчно в **X**-плоскостях физического пространства запоминающей среды вычислителя аналогично векторным разрядно-последовательным схемам. Таким образом, в **Z**-плоскостях битового куба размещаются D^2 однопозиционных разрядов различных данных (рис.1), а D^2 элементов вектора размещаются в **M** плоскостях. Соответственно, для размещения **N** элементов потребуется D^2 битовых модулей ОЗУ с емкостью – $H = M N / D^2$.

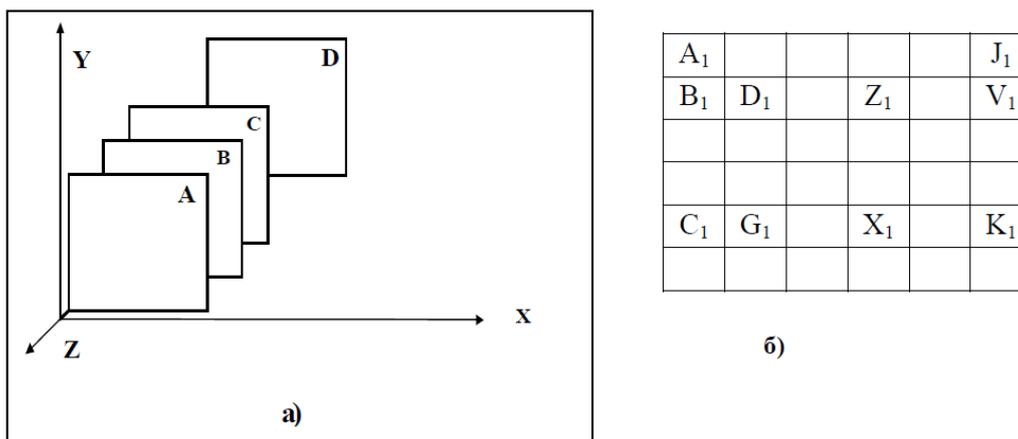


Рис. 1. Матричная разрядно-последовательная схема размещения данных

При таком размещении данных при выполнении каждой массовой операции чтения в процессорные модули вычислительной среды будут поступать од-

ноименные разряды всех элементов обрабатываемой структуры данных. Учитывая эту особенность, анализ элементов структуры данных построим по следующей схеме.

1. Пусть \mathbf{A} – множество элементов обрабатываемой структуры данных, размещенных в памяти итеративно-битовой процессорной среды по матричной разрядно – последовательной схеме. В этом случае $\mathbf{A} = \{ \mathbf{A}^1, \mathbf{A}^2, \mathbf{A}^3 \dots \mathbf{A}^M \}$ – упорядоченное множество битовых матриц из \mathbf{N} элементов. Пусть \mathbf{C} – битовая матрица из \mathbf{N} элементов, размещенная в памяти вычислительной среды также по матричной разрядно – последовательной схеме. Каждый элемент \mathbf{C} – равен значению логической 1.

2. Старший разряд каждого элемента структуры \mathbf{A} логически умножается на однопозиционный элемент структуры \mathbf{C} . Сформированная таким образом структура \mathbf{B} будет иметь такую же размерность, что и структура \mathbf{C} . Причем, если в структуре \mathbf{B} имеется хотя бы один элемент, равный по значению логической 1, то соответствующий позиционно данному элементу элемент \mathbf{A} является на данном этапе анализа максимальным по значению элементом из \mathbf{A} . В этом случае каждый элемент структуры \mathbf{C} логически умножается на однопозиционный элемент структуры \mathbf{B} , тем самым формируется модификация \mathbf{C} для следующего шага анализа. Если в \mathbf{B} нет ни одного элемента равного 1, то структура \mathbf{C} без изменений используется на следующем шаге анализа.

3. Выбираются следующие по порядку разряды структуры \mathbf{A} и операции п.2 повторяются. Процесс анализа заканчивается после обработки младших разрядов структуры \mathbf{A} . В результате выполненных операций в структуре \mathbf{C} будут сформированы единичные значения только для тех элементов, которые позиционно совпадают с элементами \mathbf{A} , имеющими максимальные значения.

Алгоритм выполнения описанной процедуры в графической форме приведен на рис. 2а. Преобразованный алгоритм с использованием операций \mathbf{S} -алгебры показан на рис. 2б. Преобразование выражений булевой алгебры в выражения

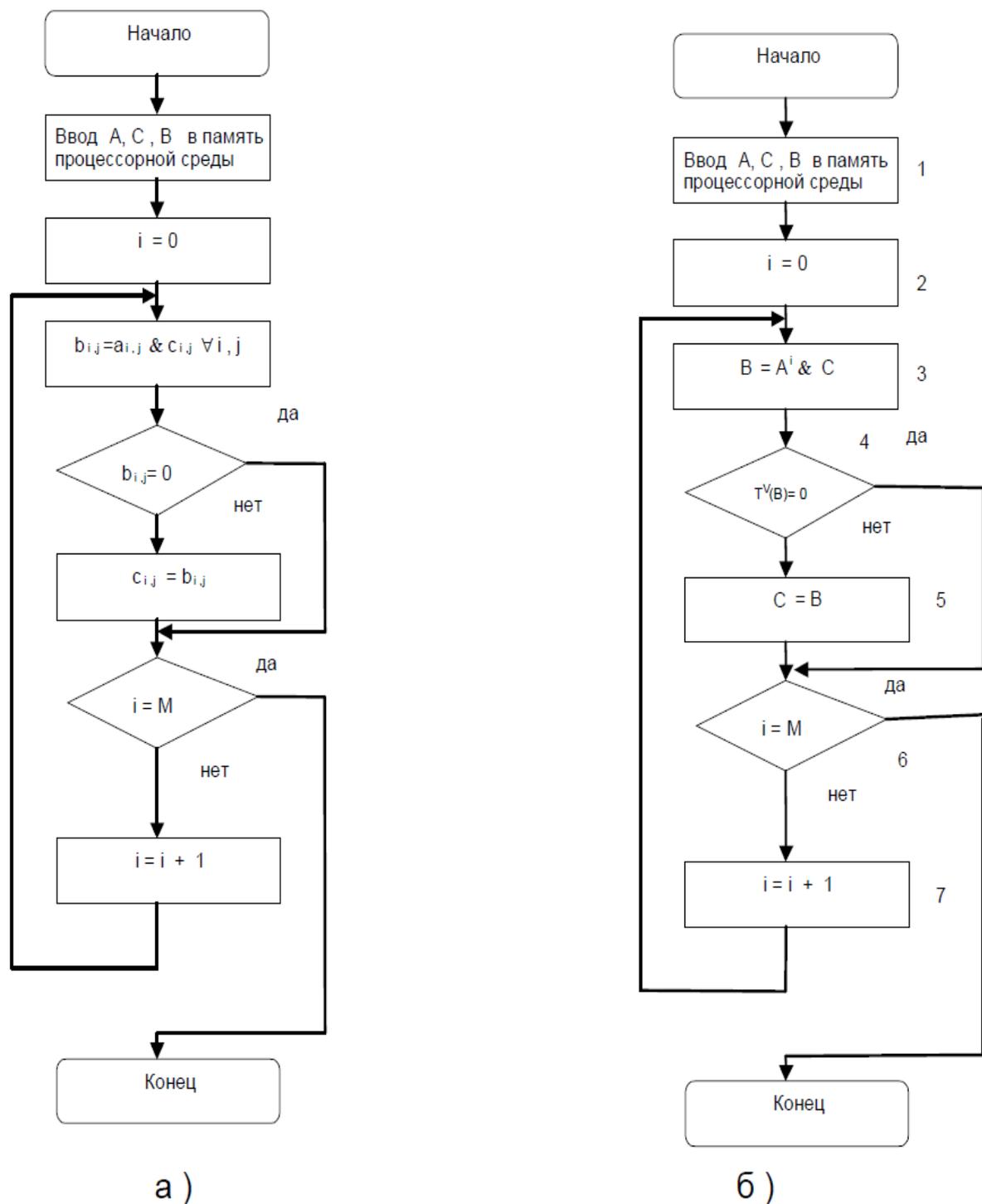


Рис. 2. Алгоритм выполнения процедуры

S - алгебры выполнено по методике, приведенной в работе [3]. В данном случае преобразования сведены к формальной замене обозначений аргументов функций алгебры логики на аргументы функций S -алгебры при сохранении со-

става и порядка выполнения операций. Временная сложность вычислений в итеративно-битовой вычислительной среде при организации процесса по алгоритму рис.2.б. будет равна $\text{TIME} (F) = 4 M$, где M – разрядность элементов обрабатываемой структуры данных, F – функция поиска максимального элемента в структуре A . Действительно, вычислительный процесс в итеративно-битовой среде из (D^2) элементов представим графом, показанном на рис. 3.

На рис.3. соответственно S_H , S_K – начальное и конечное состояние процесса, S_1 и S_2 – отображает блок 3 алгоритма по рис.2. (S_1 -ввод на обработку структуры A^i , S_2 – выполнение операции конъюнкции), S_3 – блок 4, S_5 – блок 5. Данный граф отображает только вычислительные процессы в итеративно-битовой среде и не учитывает процессы ввода и вывода данных в (из) среды, временные затраты по общему управлению и построен с учетом хранения переменных множества B в течении всего процесса вычислений во внутренней памяти процессорных элементов.

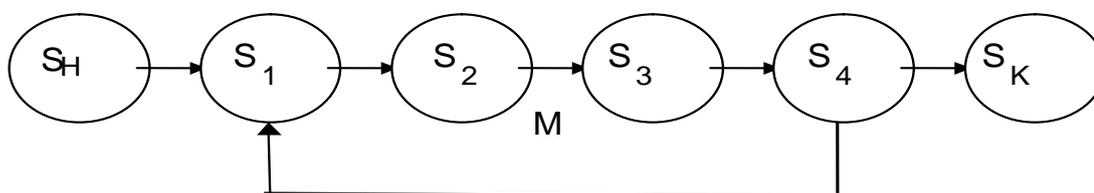


Рис. 3. Граф вычислительного процесса

В заключение отметим тот факт, что оценку временной сложности реализации процедуры поиска максимального числа можно было получить из выражения общей оценки операций данного класса до разработки алгоритма. При данном типе размещения элементов структуры A в вычислительной среде реализуется схема массивно-параллельных разрядно-последовательных вычислений. Выбранный способ реализации операции поиска максимума – итерационно-разрядный. Отсюда временная сложность таких вычислений относится к классу $\text{DTIME} (4 M N P / D^2)$. Так как количество итерационно-разрядных операций при выполнении алгоритма $P=1$ и при разработке алгоритма принято,

что $\mathbf{N} = (\mathbf{D} \times \mathbf{D})$, то подставляя данные значения \mathbf{P} и \mathbf{N} в общую формулу получим оценку временной сложности совпадающую с приведенной выше оценкой временной сложности алгоритма. Этот факт свидетельствует о том, что разработанный алгоритм оптимален для выбранной схемы вычислений.

Список литературы

1. *Князьков В. С.* Алгебраические основы массивных вычислений в итеративно-битовых СБИС-структурах // Логико-математические методы в технике, экономике и социологии: тез. докл. IV Междунар. науч.-техн. конф., октябрь, 1999 г., г. Пенза. Пенза, 1999. С. 24.
2. *Knyazkov V. S., Volchenskaya T. V.* Algebraic Fundamentals of Massive Computation // Proceeding of the Fourth Intern. Conf. Of Science and Technology “New Information Technologies and Systems”, Penza, Russia, December. Penza, 2000. P. 88–89.
3. *Волченская Т. В., Князьков В. С.* Алгебраический метод синтеза алгоритмов массивных вычислений // Логико-математические методы в технике, экономике и социологии: тез. докл. IV Междунар. науч.-техн. конф., октябрь, 1999 г., г. Пенза. Пенза, 1999. С. 27.
4. *Knyazkov V. S., Volchenskaya T. V.* Method of Organization and Complexity of Sequentially – bit Mass Processing // Proceeding of the Fourth Intern. Conf. Of Science and Technology “New Information Technologies and Systems”, Penza, Russia, December, 2000. P. 87–88.
5. *Князьков В. С.* Организация и сложность массивно-конвейерных итерационно-разрядных вычислений в одномерных вычислительных структурах // Известия ТРТУ. 1997. № 3. С. 197.
6. *Князьков В. С.* Способы организации и сложность массивных вычислений в одномерных итеративно-битовых процессорных средах // Вычислительная техника в автоматизированных системах контроля и управления: межвуз. сб. науч. тр. Пенза, 1999. С. 3–9.

КНЯЗЬКОВ Владимир Сергеевич – доктор технических наук, профессор кафедры Электронные вычислительные машины, Вятский государственный университет. 610000, г. Киров, ул. Московская, 36.

E-mail: kniazkov@list.ru

КНЯЗЬКОВА Тамара Викторовна – кандидат технических наук, доцент кафедры Прикладная информатика, Вятский государственный университет. 610000, г. Киров, ул. Московская, 36.

E-mail: glon_2014@mail.ru