УДК 681.5.62–5+681.5.681.3

ПРОГРАММНАЯ СИСТЕМА ДЛЯ ПОСТРОЕНИЯ РАЗБИЕНИЙ ПАРАЛЛЕЛЬНЫХ УПРАВЛЯЮЩИХ АЛГОРИТМОВ

Э.И. Ватутин
Курский государственный технический университет
Россия, 305040, Курск, ул. 50 лет Октября, 94
E-mail: evatutin@rambler.ru

И.В. Зотов
Курский государственный технический университет
Россия, 305040, Курск, ул. 50 лет Октября, 94
E-mail: zotov@nm.ru

Ключевые слова: параллельный алгоритм логического управления, разбиение, оптимизация
Key words: parallel logic control algorithm, separation, optimization

В работе рассматривается инструментальная программная система, предназначенная для решения задачи разбиения параллельных управляющих алгоритмов на параллельно работающие последовательные подалгоритмы (блоки), возникающая при проектировании параллельной системы логического управления. Приводится подробное описание открытой архитектуры программной системы.
A SOFTWARE SYSTEM FOR THE CONSTRUCTION OF SEPARATIONS OF PARALLEL LOGIC CONTROL ALGORITHMS / E.I. Vatutin (Kursk State Technical University, 94 50 Let Oktyabrya, Kursk 305040, Russia, E-mail: evatutin@rambler.ru), I.V. Zotov (Kursk State Technical University, 94 50 Let Oktyabrya, Kursk 305040, Russia, E-mail: zotov@nm.ru). The work addresses a software tool system designed for the solution of the problem of separating parallel logic control algorithms into a set of concurrent sequential subalgorithms (modules) that appears in the design of a parallel logic control system. Detailed description of the open system architecture is given.

1. Введение

Построение систем логического управления (СЛУ) остается одной из важных проблем современной науки, техники и технологии в силу продолжающегося активного внедрения дискретных процессов управления в различные сферы деятельности человека. Проблемы анализа и синтеза СЛУ на протяжении нескольких десятков лет остаются в центре внимания отечественных и зарубежных ученых. Их решению посвящено значительное число теоретических исследований, получен ряд важных для науки и практики результатов.

Анализ показывает, что традиционные подходы к построению СЛУ в условиях быстро растущей сложности современных объектов управления, распараллеливания задач управления, появления особых требований к гибкости СЛУ сталкиваются с серьезными трудностями. Однако успехи современной микроэлектронной технологии позволяют по-новому подойти к этой задаче с учетом возможности реализации полнофункциональной СЛУ в масштабах нескольких

специализированных СБИС. Подобные системы из СБИС (мультимикроконтроллеры) обладают быстродействием СЛУ с «жесткой» логикой и в то же время могут быть оперативно настроены (или перенастроены) на требуемый алгоритм управления подобно микропроцессорам. Они способны эффективно решать задачи параллельного логического управления и выполнять управляющие алгоритмы теоретически неограниченной сложности за счет их разбиения на параллельно работающие последовательные подалгоритмы.

В работе рассматривается задача выбора подобных разбиений с учетом заданных технологических и структурных ограничений мультимикроконтроллера. Для ее автоматизированного решения предлагается визуальная программная система, позволяющая находить разбиения с заданными свойствами различными методами.

2. Постановка и особенности решаемой задачи

Задача выбора разбиения имеет ярко выраженный комбинаторный характер и заключается в оптимальном представлении исходного параллельного алгоритма в виде множества взаимосвязанных последовательных подалгоритмов [1]. Каждый такой подалгоритм в общем случае представляет собой набор нескольких несвязанных между собой фрагментов исходного алгоритма. Множество подалгоритмов в совокупности с соответствующими связями образуют сеть взаимосвязанных алгоритмов.

К получаемым подалгоритмам, а также к структуре сети предъявляется ряд требований, состав и содержание которых зависит от архитектуры СБИС микроконтроллера, технологических ограничений, структурной организации мультимикроконтроллера, дисциплины взаимодействия микроконтроллеров и т.д. К их числу относятся:

- минимальное число подалгоритмов;
- ограниченная сложность подалгоритмов, выражаемая максимально допустимым числом вершин (микрокоманд), микроопераций и логических условий;
- минимальная сложность сети межблочных связей;
- минимальная интенсивность межблочных взаимодействий.

Кроме того, к полученным разбиениям предъявляется ряд дополнительных требований [1], к которым относятся:

- отсутствие пустых блоков в составе разбиения;
- отсутствие вершин, входящих одновременно в состав нескольких блоков (ортогональность разбиения);
- отсутствие нераспределенных вершин;
- отсутствие параллельных вершин в составе одного блока.

Приведенная постановка задачи представляет собой наиболее общий случай и в зависимости от функционально-топологический организации мультимикроконтроллера может меняться. Так, при проектировании СЛУ с полносвязной организацией можно исключить из рассмотрения учет интенствности межблочного взаимодействия, т.к. полносвязная организация предусматривает попарное взаимодействие и не предполагает транзитных передач информации. Напротив, при проектировании СЛУ с шинной структурой интенсивность межблочных

взаимодействий играет ключевую роль (учитывая критичность таких СЛУ к росту интенсивности межблочного взаимодействия).

3. Характеристика методов и алгоритмов формирования разбиений

Одной из основных особенностей решаемой задачи является ее принадлежность к классу NP-полных. Как известно [2], задачи этого класса могут быть решены точно только детерминированными алгоритмами с экспоненциальной временной (емкостной) сложностью, либо недетерминированными алгоритмами. Ввиду больших временных затрат на построение оптимального разбиения применение точных алгоритмов существенно ограничено, и на практике ориентируются на поиск приближенных решений (субоптимальных разбиений) детерминированными алгоритмами, достижимых за время, ограниченное полиномиальной функцией от размерности решаемой задачи (количества вершин алгоритма управления). Следует отметить, что оптимальные разбиения представляют теоретический интерес как средство оценки качества разбиений, получаемых с использованием эвристических алгоритмов.

Спектр существующих алгоритмов весьма широк [1]. К их числу относятся:

- последовательные алгоритмы (например, [3]) – производят синтез единственного решения, характеризуются высоким быстродействием, невысокими требованиями к ресурсам ЭВМ (в случае программной реализации), но и невысоким качеством получаемых решений;
- итерационные алгоритмы (например, алгоритм случайного перебора) – производят выбор оптимального решения из некоторой совокупности решений, характеризуются высоким качеством получаемых решений, большей (по сравнению с последовательными) вычислительной сложностью, слабой распространенностью на практике;
- алгоритмы, сводящие задачу разбиения к другой (например, в [4] задача разбиения решается как задача минимальной раскраски специального графа), – качество решения в целом определяется качеством решения задачи, к которой производится сведение;
- алгоритмы, ориентированные на построения разбиений последовательных алгоритмов и адаптированные для параллельных алгоритмов (например, [3]) – характеризуются невысоким качеством получаемых решений;
- алгоритмы, предназначенные непосредственно для разбиения параллельных алгоритмов (например, [5]) – характеризуются высоким качеством получаемых решений и умеренными вычислительными затратами.

Актуальной является задача сравнения методов между собой с целью выявления наилучших из них по тем или иным критериям, к которым могут быть отнесены качество полученного решения, время нахождения решения и т.д. Для этого необходима организация процесса построения разбиений некоторой выборки параллельных алгоритмов управления различными методами. Увеличение объема выборки алгоритмов должно оказать положительное влияния на объективность сравнения, в связи с чем актуальна задача автоматизации построения разбиений путем унификации представления исходных (параллельных алгоритмов управления) и результирующих (искомых разбиений) данных, а

также реализаций методов построения разбиений и средств сравнения и анализа результатов.

Данная работа посвещена рассмотрению разработанной авторами программной системы PAE, предназначенной для решения указанных задач.

4. Требования к программной системе

К основным требованиям, предъявляемым к программной системе, следует отнести следующие:

- унификация форматов представления исходных и результирующих данных;
- унификация форматов методов построения разбиений;
- удобное для пользователя графическое представление исходных и результирующих данных, простота редактирования данных;
- реализация набора дополнительных функций (например, оценка степени параллелизма алгоритма, сохранение алгоритма управления в графическом виде и т.д.);
- автоматизация процесса построения разбиений;
- возможность расширения (масштабирования) путем добавления новых компонент (методов) без перекомпиляции системы, независимость от языка программирования.

С учетом обозначенных требований программная система была спроектирована и реализована.

5. Архитектура программной системы

Обобщенная структура программной системы представлена на рис. 1.

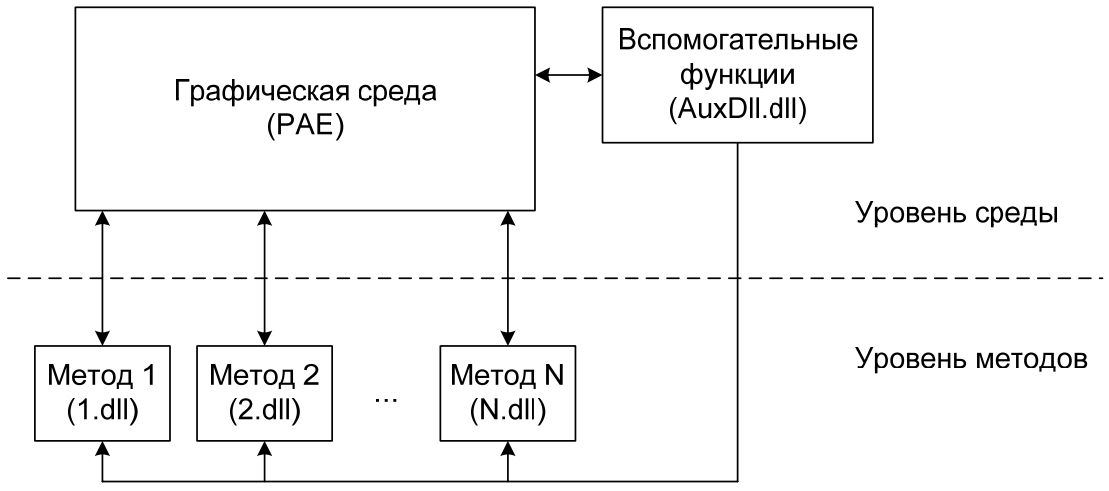

Рис. 1. Структурная схема программной системы

Как видно из рис. 1, система разбита на два уровня. На уровне среды пользователем производится выбор методов, ввод исходных данных, задание необходимых установок, построение разбиений. Уровень методов охватывает ин-

терфейсы взаимодействия библиотек методов с графической средой и библиотекой вспомогательных функций.

Под интерфейсом в данном случае (в отличие, например, от интерфейсов COM [6]) понимается набор экспортируемых библиотекой функций, которые обеспечивают требуемую функциональность (например, обмен исходными/результирующими данными). Интерфейсы подразделяются на обязательные и необязательные. Библиотека, содержащая метод, должна реализовать все обязательные интерфейсы (на данный момент это интерфейсы IDI и SMI).

В системе реализованы следующие интерфейсы:

- IDI (IDentification Interface) – интерфейс идентификации, служит для идентификации библиотеки как метода построения разбиений и получения ее основных параметров (версия, значения параметров метода по умолчанию, список поддерживаемых интерфейсов и т.д.). Если данный интерфейс не реализован в DLL, она не считается методом для построения разбиений.
- SMI (Separation Main Interface) – основной интерфейс построения разбиений. Включает в себя набор функций, отвечающих за передачу исходных данных (граф схема алгоритма управления, ограничения и параметры метода), прием результатов разбиения и освобождение выделенных в DLL ресурсов. Данный интерфейс также является обязательным к реализации.
- RMI (Relation Matrix Interface) – интерфейс обработки матрицы отношений [5]. Реализован в библиотеке вспомогательных функций (рис. 1) и служит для классификации отношений между вершинами.

Более детально интерфейсы представлены на рис. 2 на примере одного метода.

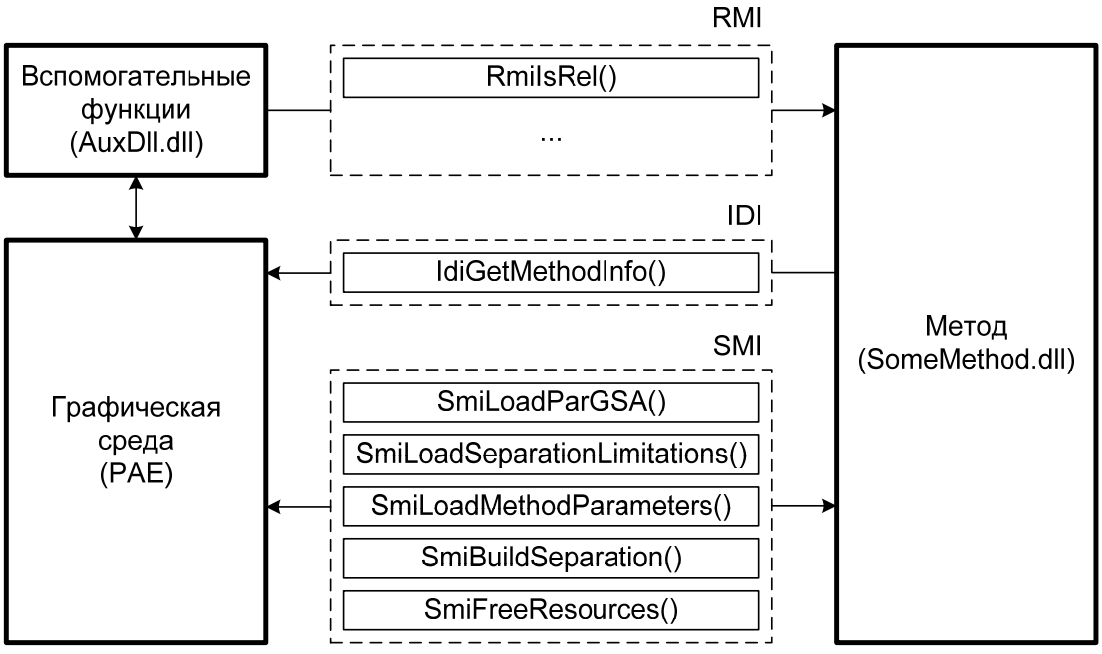

Рис. 2. Интерфейсы обмена данными

Функции интерфейсов являются экспортируемыми из DLL со стандартным соглашением о вызовах STDCALL, что позволяет добиться независимости от языка программирования. Главным требованием к языку/среде программирования является возможность создания DLL. Этому требованию отвечают все наи-

более распространенные средства разработки (например, Borland Delphi, Microsoft Visual C++, Borland C++ Builder). На этапе проектирования программной системы рассматривалась возможность использования технологии COM [6] для обеспечения языковой независимости, однако ни одно из преимуществ этого подхода не нашло применения. Кроме того, вызовы COM не являются быстрыми, поэтому в программной системе используются обычные DLL.

При сопряжении методов со средой основной объем передаваемых данных приходится на передачу различных множеств (блоки разбиения, наборы микроопераций и логических условий и т.д.). Поэтому с целью обеспечения языковой независимости, а также высокой скорости операций над множествами реализованы специальные классы, инкапсулирующие в своем составе набор необходимых методов. Множество представляется в виде битовой строки (наиболее эффективный с точки зрения затрат памяти способ), базовые операции (пересечение, дополнение, проверка принадлежности и т.д.) реализованы на ассемблере и являются высокоэффективными ввиду использования SIMD-расширений (MMX, SSE) и команд обработки битовых строк современных процессоров. Рассмотренные классы могут быть использованы не только при обмене данными между средой и методами, но и «внутри» составных частей программной системы.

Открытость архитектуры подразумевает возможность добавления компонентов (DLL методов) в систему, причем компоненты могут быть изготовлены сторонними разработчиками в соответствии с документацией (рис. 3).

Рис. 3. Фрагмент документации

При запуске программной системы производится сканирование каталога с библиотеками методов, опрос параметров и подключение методов к среде. Количество подключаемых методов ничем не ограничивается. После проведения процедуры подключения методов пользователь имеет возможность выбрать их них некоторое подмножество для построения разбиений (рис. 4).

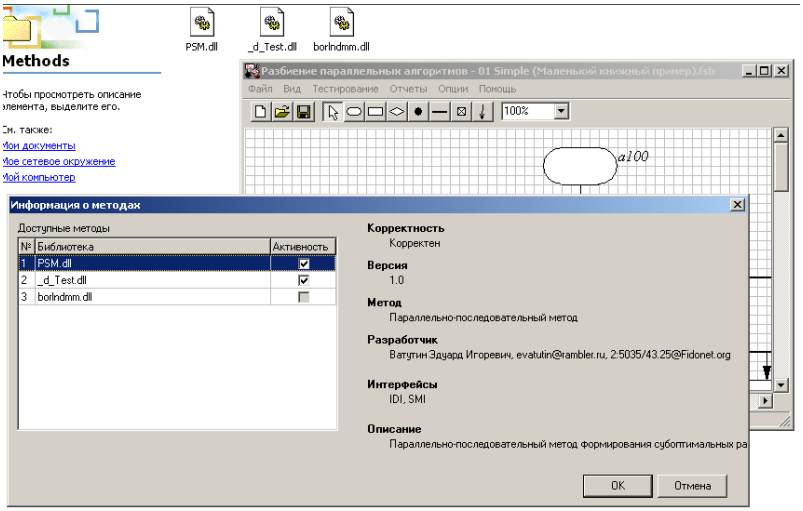

Рис. 4. Подключение и выбор методов построения разбиений

С помощью выбранных методов возможно построение разбиений алгоритмов управления, вводимых пользователем (рис. 5).

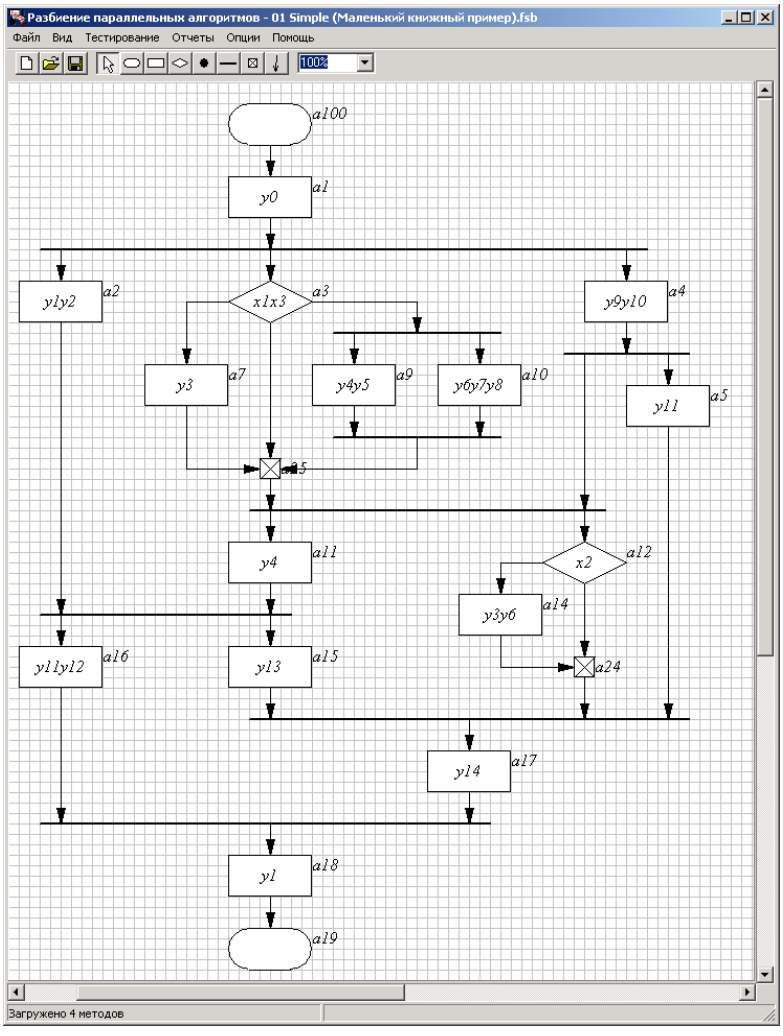

Рис. 5. Исходный алгоритм управления

Результат построения разбиений различными методами представлен на рис. 6 и 7.

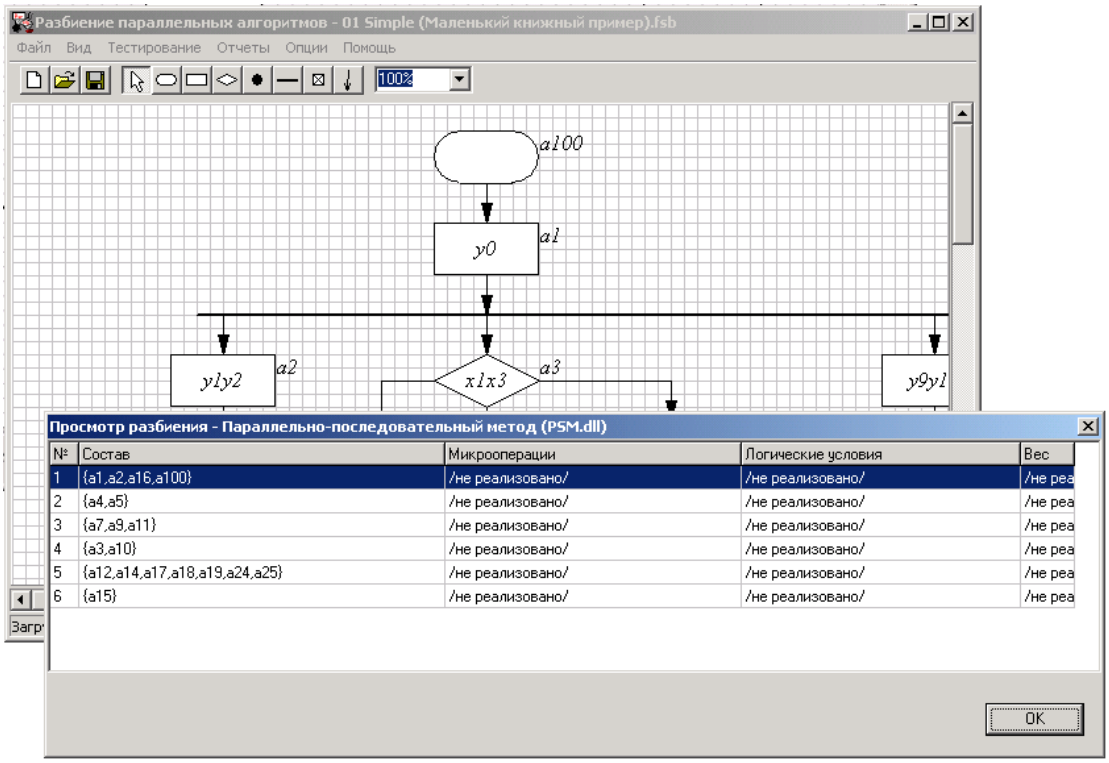

Рис. 6. Построение разбиения параллельно-последовательным методом

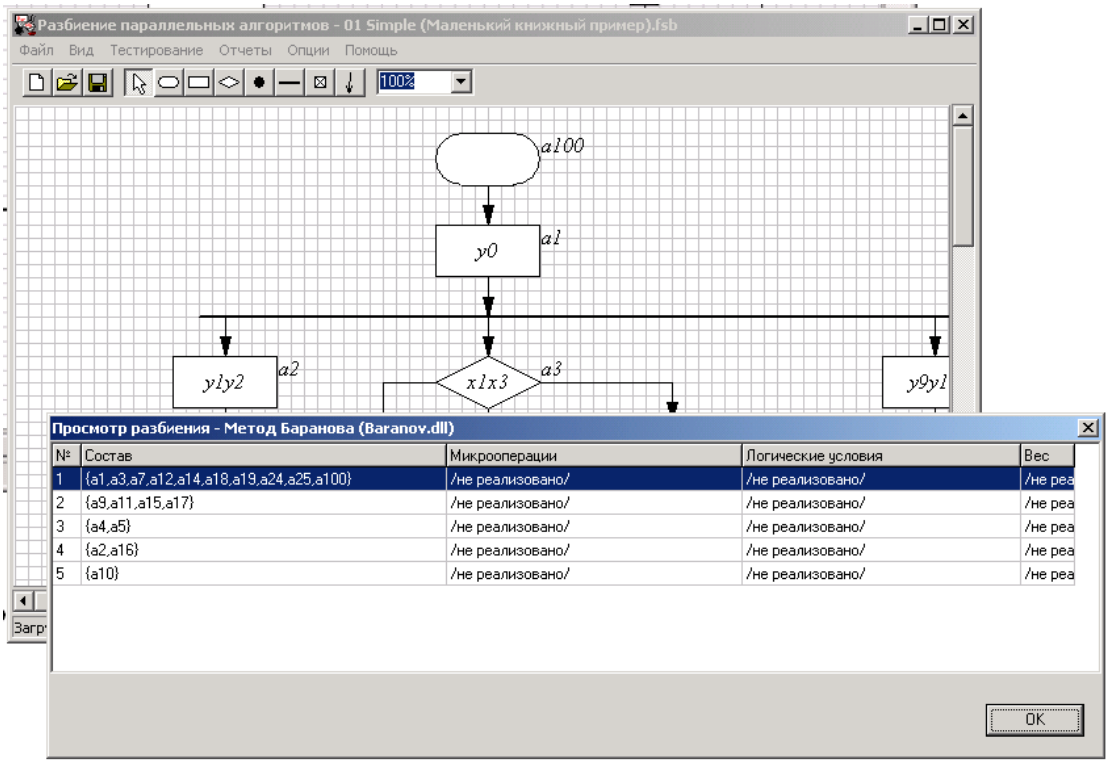

Рис. 7. Построение разбиения методом С.И. Баранова

Графическая среда PAE реализована с использованием Borland Delphi 7 по объектно-ориентированной технологии. Работоспособность интерфейсов обмена данными протестирована для DLL, разработанных с использованием всех перечисленных выше сред программирования. В настоящее время в стадии заключительного тестирования находится параллельно-последовательный метод [5] (Delphi, процедурно-модульный подход), в стадии разработки находятся методы С.И. Баранова [3] (Visual C++), А.Д. Закревского [4] (Visual C++), а также переборные методы (Delphi).

6. Дополнительные возможности программной системы

Кроме непосредственно построения разбиений в программной системе реализован набор дополнительных возможностей, облегчающих некоторые аспекты тестирования.

6.1. Унификация представления исходных данных

Исходные данные (граф-схемы алгоритмов управления) хранятся в .FSB-файлах (Figures Set Binary) в бинарном формате в виде набора т.н. фигур (фигурой называется графический примитив, отображаемый на поле рисования). FSB-файлы могут быть загружены средой PAE и при построении разбиения преобразованы в формат граф-схемы алгоритма управления с целью передачи методу построения разбиений в качестве исходных данных (рис. 8).

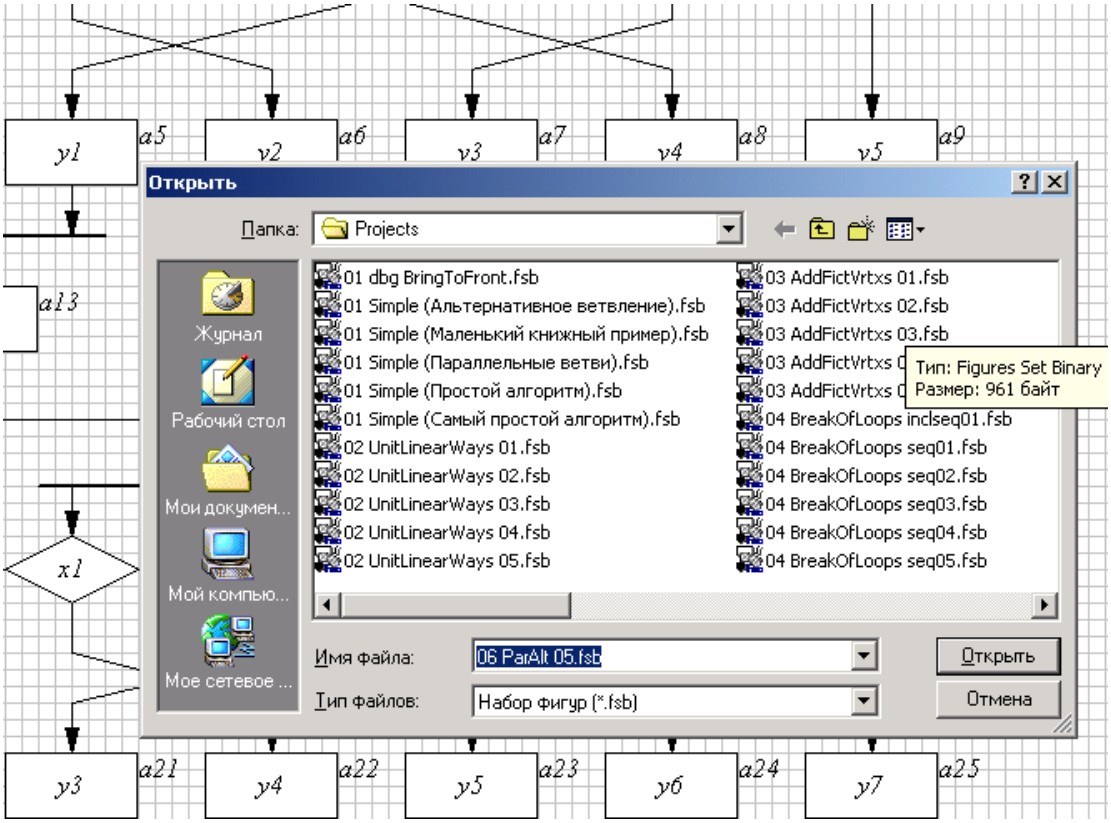

Рис. 8. Исходные данные

6.2. Возможности редактирования

Одной из ключевых особенностей является возможность ввода алгоритмов управления и редактирования их параметров в графическом виде (рис. 9).

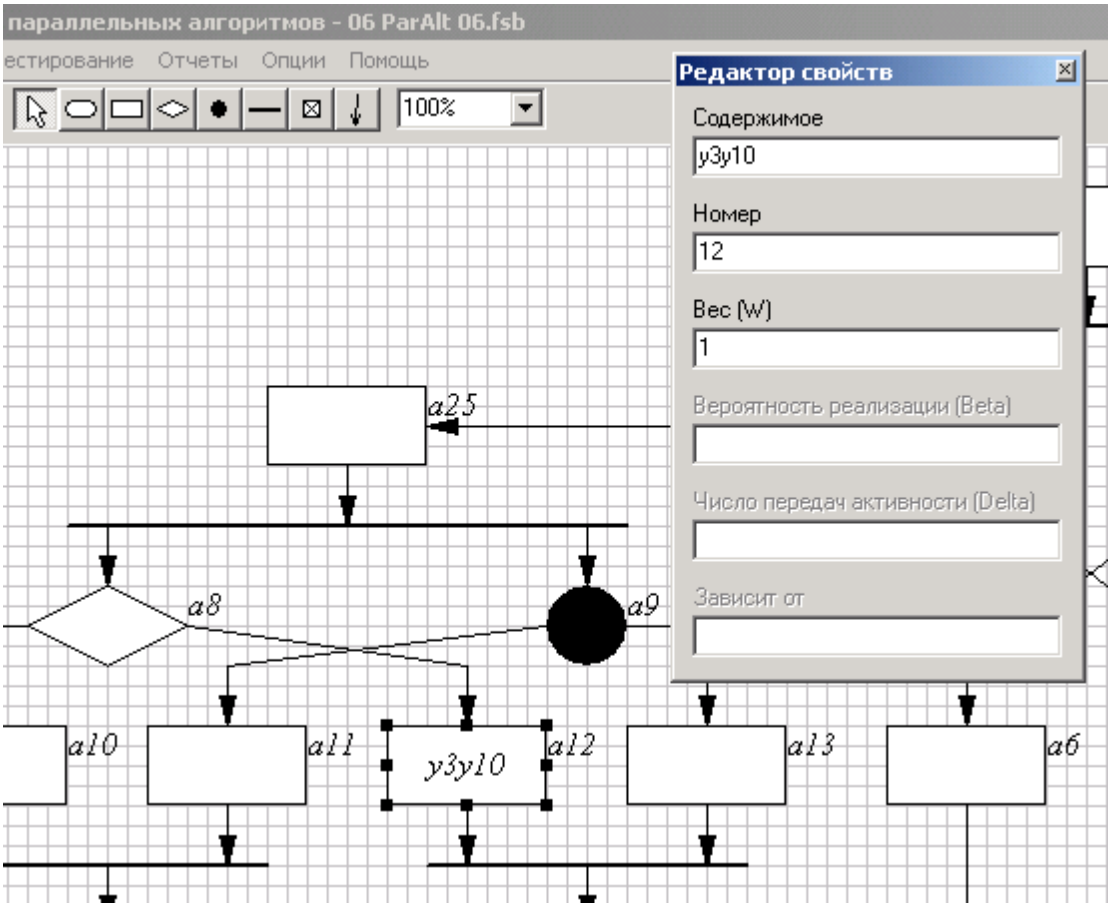

Рис. 9. Редактирование алгоритмов управления и их параметров

6.3. Автоматизация процесса построения разбиений

При построении разбиений большого числа алгоритмов управления (например, с целью сравнения качества разбиений различными методами) актуальной является задача автоматизации построения разбиений. С этой целью в программе PAE предусмотрена возможность сканирования каталога с файлыми исходных граф-схем и автоматического построения разбиений.

В настоящее время программная система находится в стадии разработки, а методы – в стадии тестирования, поэтому пока автоматическое построение разбиений используется только для выявления ошибок в методах (рис. 10).

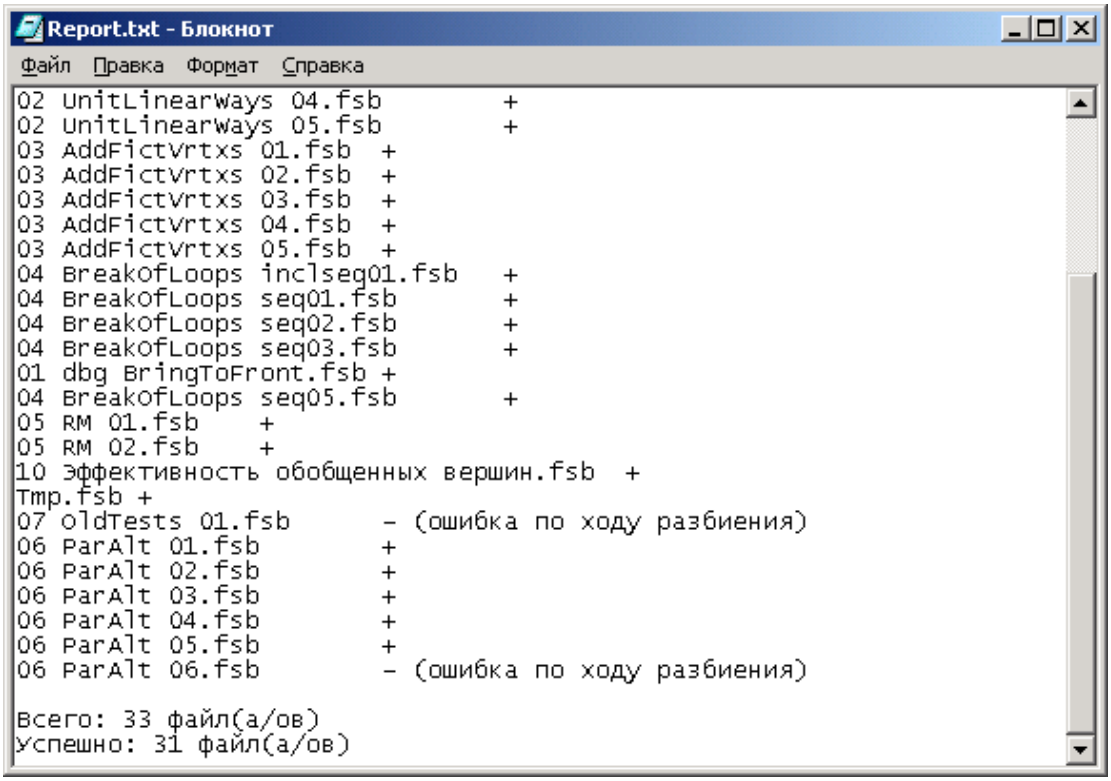

Рис. 10. Автоматическое построение разбиений с целью тестирования параллельно-последовательного метода

6.4. Нереализованные возможности

В программе PAE предусмотрен ряд возможностей, которые в данный момент еще не реализованы, однако их реализация запланирована. К ним относятся:

- генерация случайных алгоритмов на основании задаваемых пользователем параметров (вероятность появления типовых фрагментов, таких как альтернативные ветвления, линейные участки и т.д. [5]);
- проверка корректности полученных разбиений;
- оценка качества получаемых разбиений на основании весовой функции.

Список литературы

1. Организация и синтез микропрограммных мультимикроконтроллеров / Зотов И.В. и др. Курск: ГУИПП «Курск», 1999. 368 с.
2. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений, 2-е изд.: Пер. с англ. М.: Вильямс, 2002. 528 с.
3. Баранов С.И., Журавина Л.Н., Песчанский В.А. Метод представления параллельных граф-схем алгоритмов совокупностями последовательных граф-схем // А и ВТ. 1984. № 5. С. 74–81.
4. Закревский А.Д., Поттосин Ю.В. Декомпозиция параллельных алгоритмов логического управления по заданному разбиению множества предложений // А и ВТ. 1985. № 4. С. 65–72.
5. Ватутин Э.И., Зотов И.В. Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Параллельные вычисления и задачи управления. М.: Институт проблем управления, 2004. С. 884–917.
6. Н. Елманова, С. Трепалин, А. Тенцер. Delphi и технология COM. СПб.: Питер, 2003. 698 с.