

УДК 004.021:004.75

ПОСТРОЕНИЕ ГРАФИЧЕСКОЙ ОБОЛОЧКИ ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТОВ НА БАЗЕ СИСТЕМЫ ЛОГИЧЕСКОЙ ОПТИМИЗАЦИИ ФУНКЦИОНАЛЬНО-СТРУКТУРНЫХ ОПИСАНИЙ ДИСКРЕТНЫХ УСТРОЙСТВ



И.П. Логинова

*Старший научный сотрудник ОИПИ НАНБ,
кандидат технических наук, доцент
irilog@mail.ru*

И.П. Логинова

*Окончила Белорусский государственный университет, кандидат технических наук, доцент.
Работает в ОИПИ НАН Беларуси в должности старшего научного сотрудника и в Белорусском
государственном университете информатики и радиоэлектроники в должности доцента.*

*Круг научных интересов: программирование, логическое проектирование и верификация цифровых
схем, виртуализация, реализация параллельных алгоритмов.*

Аннотация. В статье описывается функционирование графического интерфейса, предназначенного для визуального формирования схемы вычислительного эксперимента в виде цепочек действий, предлагаемых элементами управления интерфейса. Визуальная среда программного средства позволяет проводить разнообразные эксперименты с использованием набора проектных операций системы логической оптимизации функционально-структурных описаний дискретных устройств FLC-2. Средствами графического интерфейса формулируется схема эксперимента, в процессе составления которой проводится выбор: программ и объектов, участвующих в эксперименте. Затем пользователь запускает составленную схему эксперимента, отслеживает результат встроенными инструментами визуализации и верификации. В состав программного средства включена также подсистема пакетирования, в которой средствами интерфейса формируются задания для параллельного выполнения логической оптимизации в объектах, собранными в пакеты.

Ключевые слова: многоядерный процессор, параллельные процессы, планировщик задач, логическая оптимизация, визуализация.

Введение. Важным этапом в автоматизированных системах проектирования является этап проектирования, в котором проводится синтез логических схем, реализующих функциональные комбинационные блоки цифровых СБИС в различных технологических базисах. Первым и особенно трудоемким этапом синтеза схемы является технологически независимая оптимизация, которая в дальнейшем и определяет такие характеристики логической схемы: площадь (число транзисторов), быстродействие, энергопотребление.

Система логической оптимизации функционально-структурных описаний цифровых устройств FLC-2 была описана в работах [1]. В качестве исходных данных для этой системы выступает функциональное либо иерархическое функционально-структурное описание проектируемой логической схемы. Результатом является оптимизированное описание, на основе которого можно получить логическую схему меньшей сложности и большего быстродействия. В основе системной организации FLC-2 лежит принцип модульной

организации выполнения проектных процедур. Взаимодействие модулей осуществляется на языке управления консольными приложениями, выполняющими проектные процедуры [1]. На разных этапах проведения оптимизации логическая схема представляется в виде различных описаний, преобразование которых осуществляется методами детализации, экстракции, оптимизации. При проведении этих преобразований учитываются характеристики и состояние логической схемы. С учетом этих атрибутов формируются цепочки проектных процедур и операций, представляющих маршрут проектирования, где программные модули реализации проектных процедур унифицированы по способу вызова и возврата, что позволяет организовывать их автоматический вызов. Рост размерности СБИС, уменьшение технологических норм приводит к усложнению процесса проектирования и подводит к необходимости организации параллельных вычислений в САПР, как одного из основных способов уменьшения времени проектирования. Возможность распараллеливания прикладных задач логического проектирования определяется рядом факторов: реализацией параллельных алгоритмов для решения прикладных задач; возможностью построения эффективных параллельных программ, реализующих эти алгоритмы; наличием технологий для реализации параллельных программ на многопроцессорной (многоядерной) архитектуре. Как показали многочисленные исследования [2-4], разработка параллельных алгоритмов для большинства логико-комбинаторных задач проектирования цифровых схем, в принципе, невозможна. Тем не менее, один из альтернативных вариантов организации параллельных вычислений на многоядерной вычислительной системе, не требующий разработки параллельных алгоритмов, реализован в системе логической оптимизации функционально-структурных описаний дискретных устройств FLC-2 [5].

В рамках системы FLC-2 реализована интеллектуальная поддержка проектирования, основанная на применении продукционно-фреймовой модели представления знаний [6], имеются свои механизмы управления процессами проектирования (механизмы продукции и стратегий), которые позволяют сократить число однотипных проектных действий, выполняемых проектировщиком, в ряде случаев значительно улучшить качество решений за счет применения комбинированных алгоритмов оптимизации. Расширение возможностей системы FLC-2 путем включения в состав проектных процедур оптимизации, которые выполняются с использованием параллельных вычислений, позволяет существенно сократить время проектирования, и, таким образом, повысить размерность решаемых задач [6].

В основу организации параллельных вычислений в FLC-2 лежит одновременный запуск на ядрах, так называемых, процессов исполнения. В каждом из процессов выполняется процедура (например, проводится оптимизация) с отдельными частями описания схемы, представленной в исходном проекте. Разбиение на части исходного проекта производится применением ряда операций. При необходимости проводятся проектные операции по выявлению и устранению информационных зависимостей между полученными в результате разбиения частями исходного объекта, что приводит к их преобразованию в независимые объекты. Независимость частей проекта, в контексте параллельной реализации означает, что входные аргументы для работы с одной частью проекта не определяются результатами вычислений с другими частями проекта. В итоге, появляется возможность проведения параллельных вычислений с частями проекта при выполнении некоторых проектных операций без блокировок и синхронизации [7].

Организация параллельной работы в системе FLC-2 представлена в работах [5-8], выполнена с использованием технологии OpenMP, которая является стандартом для реализации многопоточных приложений и содержит набор различных директив компилятора, функций соответствующей библиотеки и переменных окружения [9].

Следует отметить, что в системе FLC-2 на всех этапах проектирования проводятся преобразования только с одним проектом, который представлен описанием одной логической схемы. В этой системе отсутствует возможность одновременного проведения

однотипных проектных операций с несколькими описаниями логических схем. Потребность в массовом проведении однотипных проектных операций одновременно для нескольких описаний логических схем привело к запросу на разработку программной среды, в которой можно было бы проводить вычислительные эксперименты с разными проектами, с разными программами, представляющими проектные операции оптимизации на разных операционных платформах. При этом такая проектная среда должна предоставлять средства верификации и визуализации результатов вычислительных экспериментов и обладать удобной графической оболочкой с интеллектуальной поддержкой организации экспериментов.

Архитектура графического интерфейса визуальной среды. Поэтому для системы FLC-2 был разработано автономное программное средство, графическая оболочка, которого представляет визуальную среду с многофункциональным графическим интерфейсом, предназначенную для проведения экспериментов. Возможности интерфейса позволяют проводить эксперименты для потока проектов, формировать пакеты, проводить в параллельные вычисления. Программное средство (далее, ПС PLUS) позволяет исследовать временные показатели выполнения различных программ оптимизации, получать оценки загрузки вычислительных ядер процессора при проведении параллельных вычислений, проводить верификацию. На рисунке 1 представлен состав функциональных модулей, управлением которыми осуществляют средства ПС PLUS.

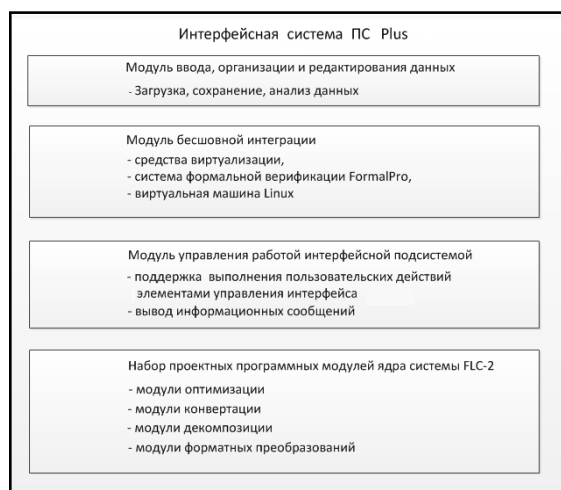


Рисунок 1. Общая структура программного средства Plus

Интерфейс ПС PLUS предоставляет инструментарий для организации пакетирования, когда пакеты представляют собой группы заданий, каждое из которых реализует свою операцию оптимизации с одним из проектов, включенных в пакет. Пакет формируется в виде коллекции независимых объектов, получаемых в результате декомпозиции описаний логических схем, представленных в каждом проекте пакета. Преобразование в независимые объекты проводится посредством процедур, входящих в ядро системы FLC-2. В качестве исходных данных, как для системы FLC-2, так и для визуальной среды PLUS выступают функциональные либо иерархические функционально-структурные описания проектируемых логических схем на языке SF [3]. Схема на языке SF определяется иерархической последовательностью функционально-структурных описаний подсхем (блоков). Функциональные описания листовых блоков дерева иерархии описания представляют собой либо логические уравнения (LOG-формат), либо матричные формы представления систем булевых функций в виде ДНФ (SDF-формат). Изменение иерархии SF-описаний и форматные преобразования позволяет переходить к реализации описания схемы в виде одноуровневой структуры описаний подсхем (2-Connect формат) [10]. При

проведении такого преобразования устраняются (элиминируются) все промежуточные внутренние переменные.

Графическая оболочка программного средства PLUS представлена экранной формой в виде окна, обеспечивает пользователю возможность проводить эксперименты, при этом использует функциональные и проектные модули системы FLC-2, которые включены в состав ядра программного среды. На рисунке 2 показан вид окна интерфейса ПС PLUS.

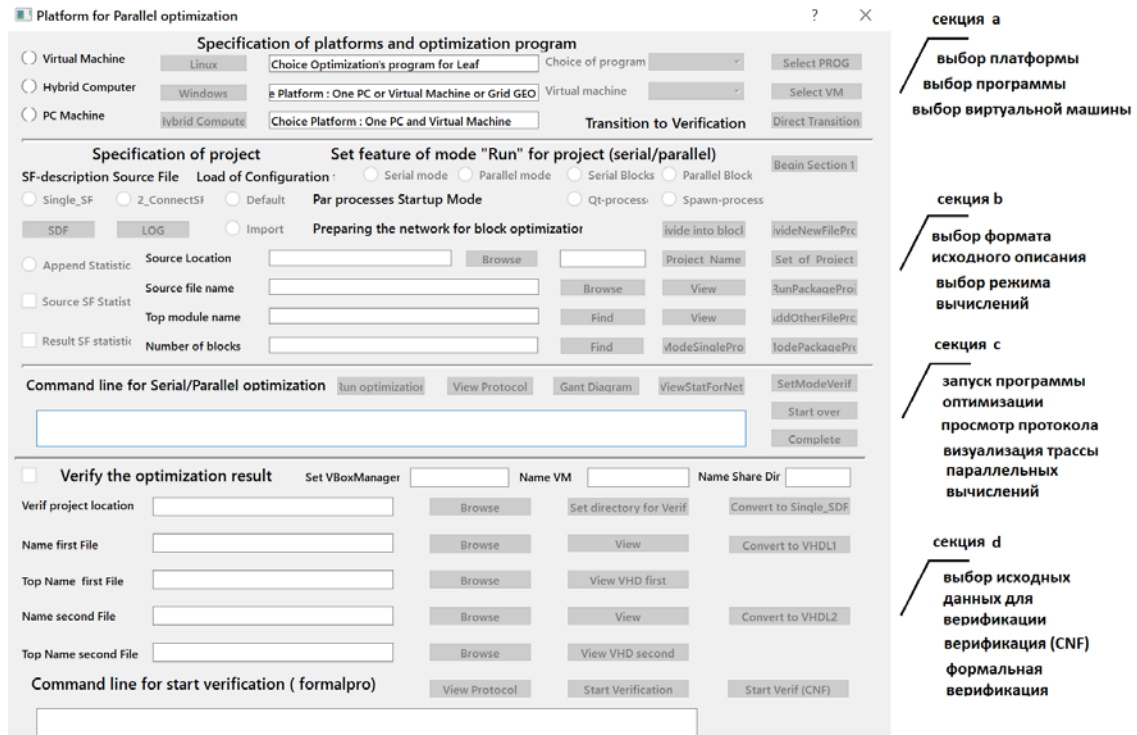


Рисунок 2. Главное окно графического интерфейса программного средства PLUS

Многофункциональный графический интерфейс позволяет существенно упрощать, и, в некоторой степени, автоматизировать проведение экспериментов с программами оптимизации путем визуально-интерактивного формирования элементами управления интерфейса цепочек действий. Одни цепочки действий позволяют конструировать пакет из разных исходных описаний логических схем. Другие цепочки действий определяют работу с набором функциональных модулей, управляющими преобразованиями и форматами данных. В окне интерфейса, представленном на рисунке 2, элементы управления сгруппированы по четырем секциям (a, b, c, d). Цепочки действий в каждой секции далее рассматриваются по отдельности.

Организация визуально-интерактивной навигации в графическом интерфейсе.

Пользователю предоставляется следующий функционал: выбор ОС (Windows или Linux), выбор различных программ логической оптимизации, формирование пакета, выполнение задания для одного проекта и заданий для всех проектов пакета в параллельном и последовательном режимах, отслеживание результатов оптимизации с запуском программ сбора статистики, визуализации и верификации. При выполнении параллельных вычислений визуализация трассы параллельных процессов, позволяет увидеть порядок выполнения процессов и распределение вычислительной нагрузки по ядрам процессора.

Навигация, осуществляемая посредством последовательной активацией элементов управления графического интерфейса ПС PLUS, предоставляет следующие возможности:

- позволяет проводить эксперименты с программами оптимизации в двух ОС;
- доступ к различным форматам и конвертерам функционально-структурных описаний дискретных устройств системы FLC-2;

- запускать исполнение программ оптимизации в параллельных процессах, в том числе из коллекции программ оптимизации, находящихся в открытом доступе;
- не отвлекаться на задачи чтения и сохранения результатов;
- получать оценку времени выполнения операций оптимизации в протоколе или в трассе параллельных процессов, в том числе получить визуальное представление этой оценки;
- сравнить эффективности программ оптимизации, выполняемых в различных ОС;
- импортировать и использовать файл конфигурации, системы FLC-2 в котором хранятся параметры конфигурации, например, параметры программ оптимизации;
- формировать структуру и порядок выполнения пакета заданий.

На рисунке 3 приведена схема организации работы в каждой секции интерфейса.

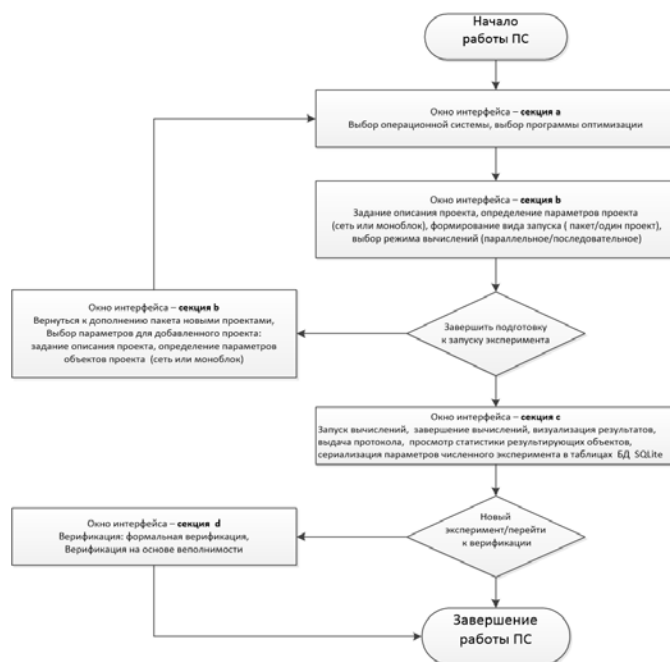


Рисунок 3. Общая схема работы с данными в окне интерфейса PLUS

На рисунке 2 представлено состояние графического интерфейса ПК PLUS, когда ни один из элементов управления (радиокнопки, кнопки, выпадающие списки) еще не был выбран (не актуализирован). Актуализация в контексте навигации по интерфейсу означает, что при выборе одного из элементов предлагаются возможные варианты последующих действий путем подсвечивания элементов, обозначающих последующие действия. Подобным образом осуществляется визуальная навигация при составлении той или иной схемы проводимого эксперимента. Работа в среде PLUS начинается с актуализации элементов управления в секции *а*. Соответствующий фрагмент окна представлен на рисунке 4. На фрагменте элементы управления обозначены номерами, что позволяет формальным образом описать порядок действий с элементами. Переход от одного элемента управления интерфейса к другому можно записать схематически так: если за выбором элемента управления *i* следует выбор актуализированного (доступного) элемента управления *j*, то навигация от *i* к *j* записывается как $i \rightarrow j$.

Подготовка к проведению эксперимента начинается с выбора операционной системы – PC (Windows), Virtual Machine (Linux), Hибриd Computer (GRID cluster). В секции *а* есть элементы, которые задают два выпадающих списка: *а6* – список доступных программ оптимизации для Windows и для Linux; *а7* – список имен виртуальных машин (BM) Linux. Выбор программы оптимизации подтверждается нажатием кнопки *а8*. Выбор виртуальной машины подтверждается нажатием кнопки *а9*. После этого становятся доступными элементы: список выбора программы и список выбора BM. На рисунке 4 также представлены три варианта навигации в этой секции полученных посредством актуализации элементов

управления для: 1) проведения эксперимента на платформе Windows; 2) проведения эксперимента на платформе Linux (виртуальная машина); 3) Переход к секции *d* для проведения верификации результатов проведенных экспериментов.

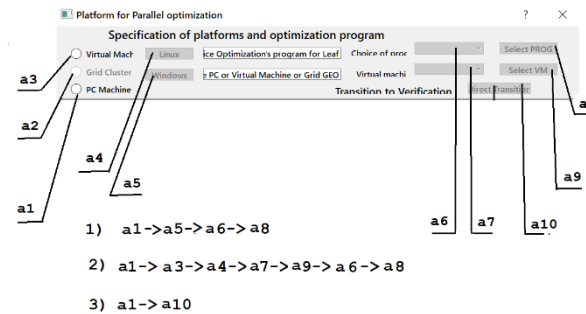


Рисунок 4. Элементы управления и варианты навигации по элементам управления в секции *a*

После выбора ОС и конкретной программы оптимизации становятся доступными элементы управления во второй секции (секции *b*). Навигация по элементам этой секции определяет: место расположения папки проекта, структуру sf-описания исходного проекта, режим вычислений, предоставление статистики об объектах проекта, подключение файла конфигурации (по умолчанию или импорт из системы FLC-2). При проведении эксперимента, как для одного проекта, так и для пакета, местом хранения исходной и результирующей информации является папка проекта. В папке проекта собираются исходные и оптимизированные описания логических схем, протоколы с результатами оптимизации, протоколы верификации, скрипты заданий для пакетирования. Актуализация элементов управления в этой секции начинается с выбора места расположения проекта. Место размещения проекта в файловой системе и имя папки проекта определяется пользователем. Фрагмент окна с секцией *b* и обозначением порядкового номера каждого элемента управления представлен на рисунке 5.

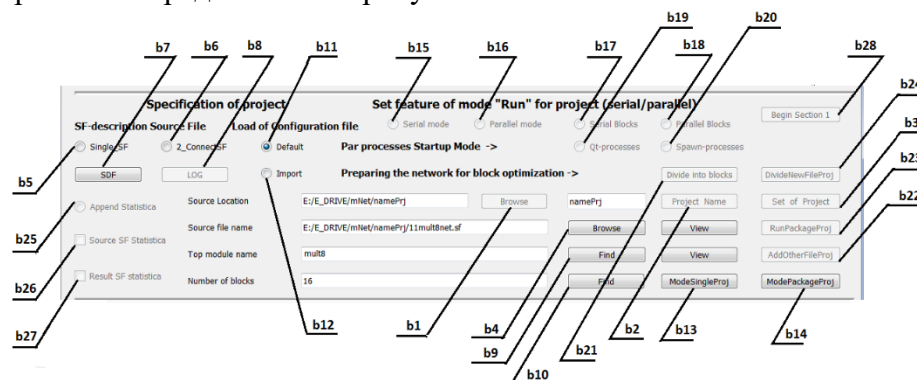


Рисунок 5. Обозначение элементов управления во второй секции интерфейса

Место размещения проекта и имя папки проекта определяются пользователем, посредством последовательным прохождением элементов: *b1* (выбор места размещения проекта), *b2* (активация окна для ввода имени папки проекта) и *b3* (фиксация имени папки проекта). После выбора места расположения и задания имени папки проекта (действия *b1*→*b2*→*b3*), при актуализации элемента *b4* (кнопка «Browse») осуществляется поиск исходного sf-описания проекта. Имя файла с этим описанием и путь к месту его расположения отображается в соответствующем окне просмотра. После нажатия элемента *b4* актуализируется группа элементов управления в левой части секции *b*, последовательным выбором которых уточняется структура и формат исходного sf-описания проекта.

Таким образом, начало навигации по элементам секции *b* выглядит следующим образом: $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4$. Далее, выбором элементов $b5 \rightarrow b7$ уточняется структура исходного проекта – sf-описание схемы представлено одним блоком в формате SDF. Выбор элементов $b5 \rightarrow b8$ определяет следующую структуру проекта – sf-описание схемы представлено одним блоком в формате LOG. Если провести выбор элементов $b6 \rightarrow b8$, то он определяет структуру описания схемы как 2-Connect сеть в формате LOG. Выбор $b6 \rightarrow b7$ определяет структуру описания схемы как 2-Connect сеть в формате SDF. Исходное описание проекта можно просмотреть (имеются соответствующие кнопки «View»). Только после уточнения структуры исходного проекта актуализируются кнопки $b9$ и $b10$. Нажатие этих кнопок позволяет в соответствующих полях просмотра визуализировать – имя головного блока и число независимых блоков, которым представлено описание исходного проекта. По завершении действий с кнопками $b9$ и $b10$ актуализируются элементы $b11$ и $b12$, которые определяют способ подключения файла конфигурации config.ini ($b11$ – по умолчанию) или ($b12$ – импорт из системы FLC-2). После определения файла конфигурации появляется возможность выбрать: продолжать эксперимент с одним проектом или организовать пакет с заданным числом проектов. Эту возможность предоставляет актуализация элементов $b13$ («ModeSingleProj» – запуск оптимизации для одного проекта) и $b14$ («ModePackageProj» – формирование пакета для проведения оптимизации с несколькими проектами). Задание и уточнение структуры описания при работе с одним проектом определяются следующими вариантами навигации по элементам управления секции *b*, которые представлены на рисунке 6.

- 1) $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b5 \rightarrow b7 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13$ – блок, формат SDF
- 2) $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b5 \rightarrow b8 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13$ – блок, формат LOG
- 3) $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b6 \rightarrow b7 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13$ – сеть, формат SDF
- 4) $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b6 \rightarrow b8 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13$ – сеть, формат LOG

Рисунок 6 . Варианты навигации по секции *b* для подготовки данных при запуске одного проекта

Порядок действий при проведении эксперимента с одним проектом. При нажатии $b13$ актуализируется подмножество из четырех радиокнопок, определяющих режим вычислений. Выбор одной радиокнопки из этого подмножества осуществляет подготовку к запуску выполнения логической оптимизации:

1 $b15$ («serial mode») – выполнение программы оптимизации в последовательном режиме для sf-описания исходного проекта, представленного в виде одного блока;

2 $b16$ («parallel mode») – выполнение программы оптимизации в параллельном режиме для sf-описания исходного проекта, представленного в виде 2-Connect сети. В противном случае выдается предупреждение о недоступности этой кнопки для программы оптимизации, которая была выбрана в секции *a*, при представлении исходного проекта в виде одного блока;

3 $b17$ («serial block») – выполнение оптимизации в обычном режиме для sf-описания исходного проекта, представленного описанием в 2-Connect формате;

4 $b18$ («parallel block») – выполнение оптимизации в параллельном режиме для sf-описания исходного проекта, представленного описанием сети в 2-Connect формате. Фрагмент третьей секции (секции *c*) окна интерфейса представлен на рисунке 7.

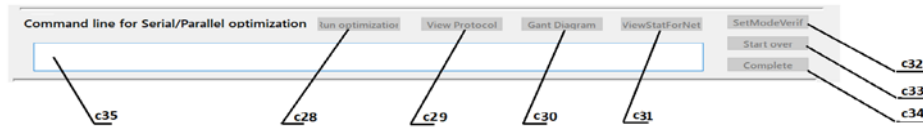


Рисунок 7. Назначение элементов управления в третьей секции (с) интерфейса

На рисунке 7 указаны номера для обозначения каждого элемента управления данной секции. В секции расположено окно просмотра, в котором отображается командная строка запуска выбранной программы оптимизации. После завершения действий по уточнению структуры объекта (один проект), которые были доступны при выборе *b13*, и выбора одного из режимов вычислений (*b15|b16|b17|b18*) актуализируется элемент в третьей секции *c28* («Run optimization»), нажатие которого приводит к запуску программы оптимизации. В окне просмотра (*c35*) отображается командная строка с параметрами запускаемой программы оптимизации. При завершении вычислений актуализируются элементы *c29* («View protocol»), *c30* («Gant Diagram») и *c31* («ViewStatNew») – выбор просмотра протокола, визуализации и статистики. В этой секции предусмотрен элемент *c35* («Complete») – завершение работы PLUS. На рисунке 8 приведены варианты навигации в секции *b* для проведения эксперимента с одним проектом, представленном или в виде одного блока или в виде 2-Connect-сети.

- 1 $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b5 \rightarrow b7 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13 \rightarrow (b15|b16) \rightarrow c28$ – блок, формат SDF
- 2 $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b5 \rightarrow b8 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13 \rightarrow (b15|b16) \rightarrow c28$ – блок, формат LOG
- 3 $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b6 \rightarrow b7 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13 \rightarrow (b17|b18) \rightarrow c28$ – сеть, формат SDF
- 4 $b1 \rightarrow b2 \rightarrow b3 \rightarrow b4 \rightarrow b6 \rightarrow b8 \rightarrow b9 \rightarrow b10 \rightarrow b11(b12) \rightarrow b13 \rightarrow (b17|b18) \rightarrow c28$ – сеть, формат LOG

Рисунок 8. Завершение навигации действием, запускающим эксперимент с одним проектом

Если необходимо провести вычисления с той же программой оптимизации, но с исходными данными другого проекта, проводится выбор элемента *c33* («Start over»). В таком случае при сохранении установок первой секции проводится заново актуализация элементов в секции *b*, согласно маршрутам навигации, приведенным на рисунке 8, начиная с определения места расположения нового проекта. Если надо провести верификацию результатов эксперимента, то выбор элемента *c32* («SetModeVerif») актуализирует элементы четвертой секции (секции *d*) для работы с программами верификации.

В вычислительном ядре визуальной среды PLUS представлены: набор модулей, реализующих процедуры логической оптимизации в системе FLC-2, параллельные версии программ оптимизации (Minim, bdd_builder, BoolNetOpt), исполняемые коды таких известных программ оптимизации, как Espresso и ABC [5]. Выбор элемента *c28* запускает выполнение модулей оптимизации с ключевыми параметрами, формируемыми структурой исходного описания и условиями запуска конкретной программы.

Замечание 1. Если исходное описание схемы представлено сетью в 2-Connect формате, кнопки *b17* и *b18* не будут актуализированы, пока не будет проведено выбором

элемента *b21* («Divide into block») разделение сети на отдельные файлы с sf-описаниями независимых блоков (3-й маршрут на рисунке 8).

Замечание 2. Кнопки *b25*, *b26*, *b27* актуализируются по завершению задания исходных данных, и могут быть выбраны при необходимости получения полной статистики для исходного и оптимизированного sf-описаний проекта. Статистика отображается в протоколе при завершении процесса оптимизации.

Замечание 3. Программы, включенные в вычислительное ядро визуальной среды PLUS и выполняемые в ОС Windows, программы, выполняемые в ВМ с ОС Linux (на гостевой виртуальной машине), находятся в общем выпадающем списке секции *a*, который открывается при нажатии элемента *a8*.

Замечание 4. Актуализация группы элементов *b15*, *b16*, *b17*, *b18* определяется применимостью выбранной программы оптимизации для описания схемы с конкретной структурой и форматом. Например, если при выборе распараллеленной версии программы раздельной минимизации *MinimSepPar* ошибочно было указано действие *b15* («serial mode»), выдается предупреждение с подсказкой выбора нужного для этой программы действия, как показано на рисунке 9. Совершив действие, указанное в этом предупреждении, проведение эксперимента с этой программой можно продолжить.

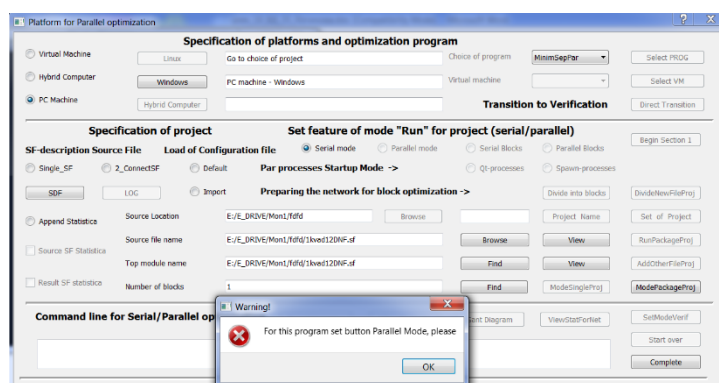


Рисунок 9. Предупреждение о неправильном выборе режима вычислений

Порядок действий при проведении эксперимента с пакетом проектов. При выборе элемента *b14* («ModePackageProj») осуществляется подготовка по организации экспериментов в подсистеме пакетирования. Формирование пакета запускается, ставшим доступным, элементом *b22* («AddOtherProj»). При этом в секции *b* очищаются все поля с данными предыдущего проекта, затем проводятся действия согласно маршрутам навигации, представленными на рисунке 8, но с формированием тех параметров исходных данных и режимов вычислений, которые соответствуют проекту, присоединяемому к пакету. Для добавленного проекта можно поменять и программу, которая будет работать с данными этого проекта, так как становится доступным элемент *a8* в первой секции.

Каждый раз добавление очередного проекта в пакет предшествует выбор элемента *b22* («AddOtherProj»). Формирование нужного числа проектов в пакете завершается действием *b23* («RunPackageProj»), которая была доступна на всем этапе формирования пакета. В результате действия *b23* актуализируется подмножество из четырех элементов, определяющих режим вычислений и параметры запускающей программы для пакета:

1 *b17* («serial block») – выполнение в обычном режиме программ оптимизации, назначенных для проектов, которые собраны в сформированный пакет;

2 b18 («parallel block») – выполнение в параллельном режиме программ оптимизации назначенных для проектов, которые собраны в сформированный пакет;

3 b19 («Qt-processes») – запуск программ оптимизации, которые назначены для проектов в пакете, командой «start»;

4 b20 («Spawn-processes») – запуск программ оптимизации, которые назначены для проектов в пакете, командой «spawn».

Замечание 5. Если описание проекта, добавляемого в пакет, представлено 2-Connect сетью, то элементы b22 (добавление нового проекта в пакет) и b23 (завершение формирования пакета) не будут актуализированы, до тех пор, пока не будет проведено разделение 2-Connect сети предыдущего проекта на файлы с описаниями отдельных блоков. Таким образом, при представлении описания нового проекта в виде 2-Connect сети только после выбора элемента b24 («Divide into block») актуализируются кнопки b22 и b23.

Замечание 6. Выбор элементов управления b24 (при работе в пакетном режиме) и b21 (при работе с одним проектом) заключается в выполнении следующих действий:

- разделение исходной 2-Connect- сети на блоки, проведение анализа блоков, которые имеют промежуточные переменные, исключение промежуточные переменные в этих блоках;

- подготовка и преобразование структуры блоков, унификации имен блоков;

- занесение sf-описаний блоков в отдельные файлы, перенос файлов в папку проекта.

На рисунке 10 представлена схема всех связей элементов управления в главном окне графического интерфейса визуальной среды PLUS с указанием порядка действий и маршрутов навигации (работа с секцией верификации, в данной статье не рассматривается). Если провести анализ формирования всех возможных маршрутов навигации по элементам интерфейса, можно прийти к следующим выводам:

1 При работе в визуальной среде PLUS требуется лишь понимание того, с какими проектами проводятся эксперименты, и какие программы оптимизации для этих проектов надо запустить. Навигация по элементам интерфейса с интеллектуальной поддержкой требует одного – следовать указаниям доступных элементов управления интерфейса, согласно выбранному пользователем цели эксперимента.

2 Вся логика взаимодействия системных, программных модулей визуальной среды, функциональных модулей процедур оптимизации системы FLC-2, средств виртуализации и верификации, отделено от пользователя графической оболочкой с многофункциональным интерфейсом. Особенностью интерфейса является: «умное» управление действиями пользователя, сериализация параметров численного эксперимента, формирование таблиц базы данных в формате SQLite [14] с информацией о прошедших экспериментах, проведение параллельных вычислений с возможностью визуального анализа результатов параллельных процессов, как в экспериментах с одним проектом, так и с проектами, собранными в пакет.

Таким образом, реализации вычислительных экспериментов с программами логической оптимизации согласно рассмотренным выше схемам, в определенной мере представляют собой потоки выполнения (workflow) в нотации сеанса эксперимента, которые позволяют компоновать различные вычислительные задачи в последовательные и параллельные структуры, направленные на достижение цели проводимого эксперимента. Потоки выполнения может быть описаны графом, где вершины являются действиями, выполняемые над данными, а ребра – потоками данных. Такая модель позволяет абстрагироваться от тонкостей запуска отдельных приложений и функциональных модулей. В оболочке PLUS в качестве действий выступают элементы управления, описанные в предыдущих разделах. Визуальная оболочка, встроенная в PLUS, позволяет

пользователю проектировать будущий workflow и задавать характеристики его выполнения.

Таким образом, пользователь может проводить вычисления, создавая конструкции, которые отражают семантику вычислительных процессов. К тому же, одной из главных особенностей PLUS является то, что программы оптимизации могут запускаться в параллельном режиме, что позволяет пользователю в полной мере задействовать возможности многоядерной вычислительной архитектуры. В случае запроса на параллельное выполнение, происходит автоматическая декомпозиция исходных данных, распределение звдвч кооперативным планировщиком по ядрам процессора, выполнение оптимизации, сбор и слияние в оптимизированное описание логической схемы, в соответствии с форматами системы FLC-2.

Из всех многочисленных связей программных модулей и действий, представленных на рисунке 10, помечены цветом те сущности, которые необходимы для достижения цели проводимых вычислений и которые должны четко определяться действиями пользователя, как показано ниже на рисунке 11.

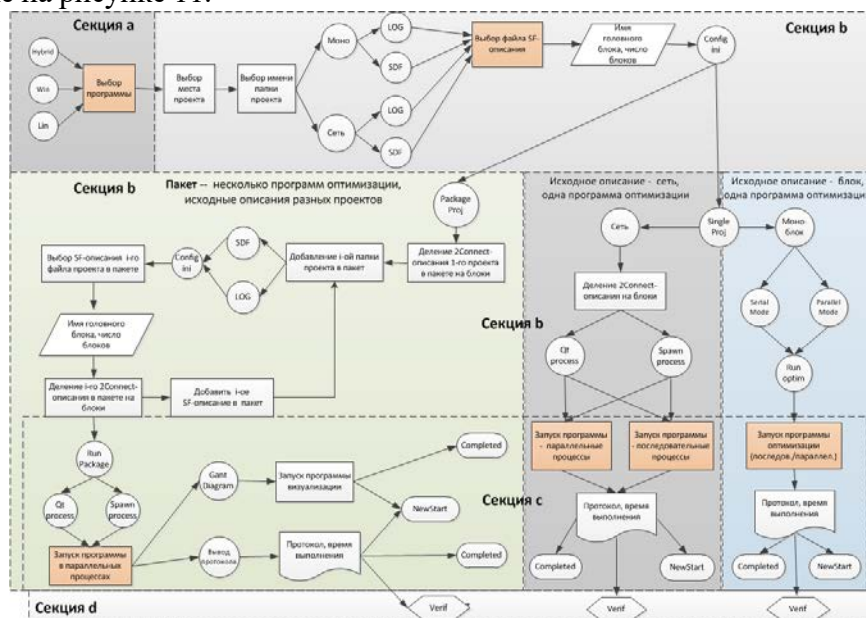


Рисунок 10. Графовая модель workflow эксперимента в нотации визуальной оболочки PLUS

Остальные элементы workflow, входящие в маршрут навигации формируются с обеспечением интеллектуальной поддержки программными средствами интерфейса PLUS в зависимости от структуры проекта, типа оптимизации и типа вычислений.



Рисунок 11. Действия пользователя при проведении вычислительного эксперимента

Организация визуализации при проведении параллельных вычислений в ПС Plus. Средства интерфейса визуальной среды PLUS предоставляют возможность визуализации трассы параллельных вычислений при проведении эксперимента с пакетами или с проектом, в котором описание логической схемы представлено в виде сети в 2-Connect формате. Визуальный анализ позволяет оценить эффективность распределения параллельных процессов

по ядрам вычислительной системы для получения характеристик производительности, оценить оптимальность процедур разбиения исходных проектов на независимые блоки. Присутствие такого средства в системе PLUS можно обосновать следующими доводами. Эффективность параллельной программы зависит от большого числа параметров: как от архитектуры вычислительной системы, алгоритмов прикладной задачи, так и от библиотек поддержки различных технологий параллелизма. Помимо вопросов, связанных с решением вышеперечисленных проблем, существует проблема, заложенная в самой природе применения параллельных вычислений для решения логико-комбинаторной задач, – недетерминированность выполнения [12]. Параллельная программа на одних и тех же входных данных может давать разные результаты. Поэтому для валидации полученных параллельных решений в PLUS включена подсистема верификации результатов оптимизации [13].

В третьей секции графического интерфейса предоставляется возможность действием элемента *c30* получить визуальное представление метрик, которые в процессе выполнения параллельной программы записываются в файлы, называемые трассами (*trace file*). Данные метрики отображаются на визуальные абстракции, заимствованные из статистической графики, в частности, в виде горизонтальных диаграмм с накоплением [15]. Визуализация трассы, в которой зафиксированы последовательность событий и их временные отметки, возможна после завершения программы, когда актуализируется элемент управления *c30*.

Особенности параллельных вычислений в визуальной среде PLUS. В основе организации параллельной вычислений в визуальной среде PLUS, также как и в системе FLC-2, положено выполнение действий: разделение (декомпозиция) объекта проектирования на части, выявление и устранение информационных зависимостей между частями, распределение работы с этими частями по вычислительным элементам (по ядрам процессора). Поскольку любая выполняемая программа образует процесс, то для запуска определенного числа независимых процессов (определяется числом декомпозируемых блоков объекта проектирования) возможна достаточно простая организация параллельного выполнения процессов, если между ними нет межпроцессных коммуникаций. Это осуществляется путем вставки директив OpenMP в код программы, которая производит запуск выполнение процессов. В программе, запускающей параллельные процессы, создается множество так называемых «задач» путем вставки соответствующей директивы OpenMP [9]. Каждая задача производит запуск одного процесса. Задачи динамическим образом распределяются по ядрам (включая логические процессоры, если они есть). Данная схема удобна в том смысле, что: а) она не зависит от численного метода, реализованного в проектной операции; б) в процессах, выполняемых параллельно можно запускать программы, представленные только исполняемыми кодами [8]. При таком подходе организация параллельных вычислений возможна по принципу: один процесс исполнения (код какой-либо программы, разные входные данные) – одно ядро процессора.

Примеры визуализации, полученные при проведении разных экспериментов. На рисунке 12 приведены фрагменты файлов конфигурации для четырех проведенных экспериментов, в которых описаны: имя и место расположения исходных проектов, имя программ, работающих с каждым из проектов. Файл конфигурации подается как один из аргументов командной строки для программы запуска параллельных вычислений. Для оптимизации в экспериментах 1-4 использовались программы, которые указаны в файле конфигурации.

Файлы конфигурации для экспериментов 1-4			
Эксперимент 1	Эксперимент 2	Эксперимент 3	Эксперимент 4
E:/All_Exper2/Net/prim1 MinimSepPar 24	E:/AllExper2/maNet/v111 MinimABC 28 E:/AllExper2/maNet/v222 MinimABC 28 E:/AllExper2/maNet/v333 MinimABC 26 E:/AllExper2/maNet/v444 MinimABC 16	E:/AllExper2/maNet/ff111 ForEsp 28 E:/AllExper2/maNet/ff222 ForEsp 28 E:/AllExper2/maNet/ff333 ForEsp 26 E:/AllExper2/maNet/ff444 ForEsp 16	E:/AllExper2/maNet/vv111 MinimABC 28 E:/AllExper2/maNet/vv222 ForEsp 28 E:/AllExper2/maNet/vv333 MinimSepPar 26 E:/AllExper2/maNet/vv444 MinimABC 16

Рисунок 12. Файлы конфигурации для запуска параллельных процессов в экспериментах 1-4

Файл конфигурации подается как один из аргументов командной строки для программы запуска параллельных вычислений. Для оптимизации в экспериментах 1-4 использовались программы, которые указаны в файле конфигурации.

Эксперимент 1. Визуализация результатов параллельного выполнения программы MINIM [5] для 2-Connect-сети, которая разделена на 24 блока.

Эксперимент 2-3. Визуализация результатов параллельного выполнения двумя программами (ABC и Espresso) четырех проектов, собранных в пакет.

Эксперимент 4. Визуализация результатов параллельного проведения оптимизации разными программами в каждом из четырех проектов, собранных в одном пакет.

Вычисления проводились на могоядерном процессоре (8 ядер плюс 8 логических ядер при включении гипертрейдинга). Таким образом, в первом эксперименте выполнение программы MINIM проходит в 24-х процессах, которые распределяются по 16-ти ядрам процессора. После завершения работы получена трасса, в которой представлены значения метрик, показывающих время начала и завершения процесса, длительность выполнения программы, запускаемой в данном процессе. В трассе для каждого процесса приводятся: номер потока исполнения (номер ядра), номер блока данных с которым проводится вычисления. Для визуального оценки распределения вычислений между ядрами процессора проводится построение графического отображения трассы в виде горизонтальной столбиковой диаграммы с накоплением (модифицированной диаграммы Ганта) [15].

Использование этого инструмента визуализации в контексте отображения результатов параллельных вычислений, можно использовать не только для оценки загрузки ядер процессора, но также позволяет увидеть возможные проблемные места при организации параллельной работы с использованием различных алгоритмов разбиения и оптимизации. При построении диаграммы сначала проводится фильтрация событий по потокам исполнения, по результатам которой формируется подмножества событий, относящихся к каждому из ядер. Затем в каждом из полученных подмножеств, проводится сортировка процессов по возрастанию времени от начала и до окончания последнего процесса в данном подмножестве. Поскольку в трассе отдельного процесса указывается время выполнения программы, именно эти значения будут определять размеры интервалов на диаграмме. На рисунках 13 и 14 приведены диаграммы Ганта, построенные по трассам экспериментов 1 и 2.



Рисунок 13. Диаграмма распределения процессов по ядрам для одного проекта

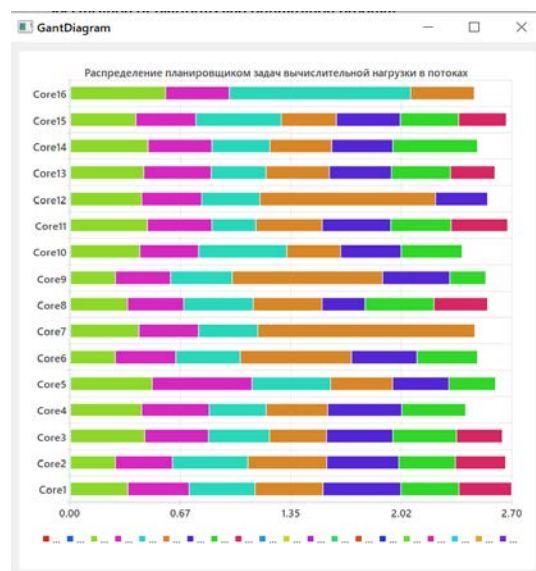


Рисунок 14. Программа ABC. Диаграмма распределения процессов пакета по ядрам

Эксперимент 2. Параллельное выполнение четырех проектов, собранных в пакет. Для всех 98-ми блоков выполнялась программа ABC, алгоритмы которой эффективно работают при минимизации ДНФ булевых функций большой размерности [5]. По завершению всех параллельных процессов получен файл трассы, визуализация которого приведена на рисунке 14.

Длина каждого интервала на диаграмме определяется временем работы программы оптимизации в одном процессе. Разные значения этой величины определяется параметрами исходных данных, которые представляют содержимое каждого блока. Нумерация блоков в пакете определяется порядком расположения исходных объектов в файле конфигурации,

Эксперимент 3. При проведении эксперимента использована программа Espresso [5]. Параллельное выполнение организовано для пакета, собранного в том же порядке, как и для эксперимента 2. В 98-ми процессах проходило выполнение программы минимизации Espresso для каждого блока. По завершению всех параллельных процессов, получены файл трассы с указанием порядка выполнения процессов в ядрах. На рисунке 15 приведена диаграмма Ганта, построенная по данным файла трассы, полученной в эксперименте 3.

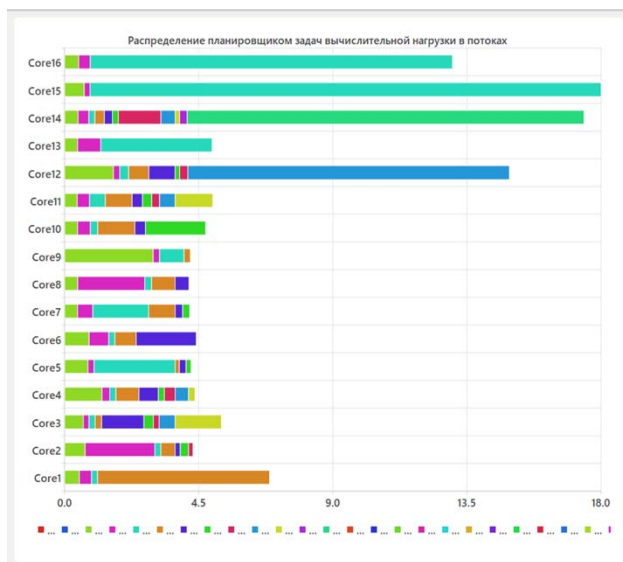


Рисунок 15. Программа Espresso.
Диаграмма распределения процессов пакета
по ядрам

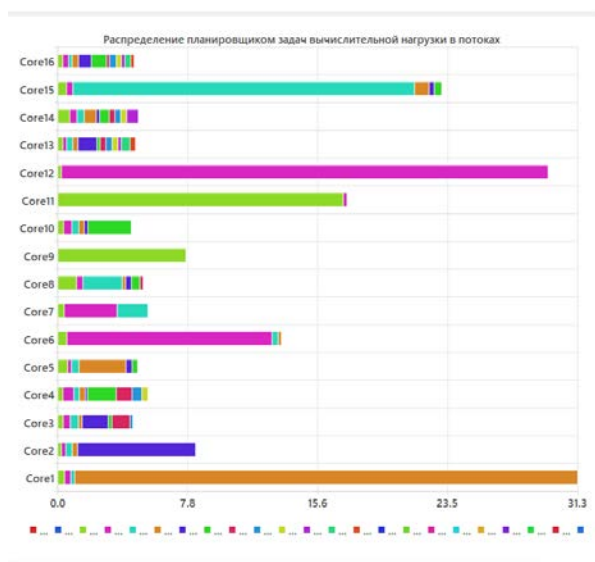


Рисунок 16. Диаграмма распределения
процессов пакета по ядрам при
выполнении разных программ

Эксперимент 4. Визуализация результатов параллельного проведения вычислений несколькими программами (ABC, Espresso, MINIM) для четырех проектов представлена на рисунке 16. Во время работы 98-ми параллельных процессов проходило выполнение оптимизации с той группой блоков, для которой назначена каждая программа оптимизации.

Заключение. Графические интерфейсы являются важнейшим элементом человеко-компьютерного взаимодействия. Они обеспечивают формирование входных данных для программ и визуализацию результатов. Система FLC-2 предлагает удобный интерфейс для разработчика, но при этом лишена встроенных средств одновременной организации вычислительных экспериментов с различными описаниями логических схем. Построение автономной визуальной среды PLUS, интегрированной по данным и функциональным решениям расширяет возможности системы FLC-2. В настоящей работе представлен поэтапный процесс работы в графической оболочке для постановки численных экспериментов. Особенности интерфейса визуальной среды расширяют и дополняют возможности системы FLC-2, повышают уровень автоматизации при выполнении численных экспериментов, позволяют увидеть схему эксперимента в целом, снижает количество ошибок и повышает оперативность принятия проектных решений.

Список литературы

- [1] Бибило, П. Н. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов // Проблемы разработки перспективных микро- и наноэлектронных систем. – 2020. – Вып. 4. – С. 9–16.
- [2] Banerjee, P. Parallel Algorithms For VLSI Computer-Aided Design / P.Banerjee – Prentice Hall, Englewoods Cliffs, NJ, 1994. – 699 p.
- [3] Черемисинов, Д.И. Проектирование и анализ параллелизма в процессах и программах / Д.И. Черемисинов. – Минск: Беларус. навука, 2011. – 300 с.
- [4] Черемисинов, Д.И. Использование параллельных вычислений при автоматизированном проектировании СБИС / Д.И. Черемисинов, Л.Д. Черемисникова // Проблемы разработки перспективных микро- и наноэлектронных систем – 2016. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. – М.: ИППМ РАН, 2016. Часть – С. 2-8.
- [5] Бибило, П.Н. Экспериментальное сравнение эффективности программ минимизации систем булевых функций в классе дизъюнктивных нормальных форм / П.Н. Бибило, И.П. Логинова // Информатика. – 2022. – № 2(19). – С. 26–55.

[6] Бибило, П. Н. Система логической оптимизации функционально-структурных описаний цифровых устройств на основе продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов // Проблемы разработки перспективных микро- и наноэлектронных систем. – 2020. – Вып. 4. – С. 9–16.

[7] Логинова, И. П. Построение и визуальный анализ параллельных решений в системе логической оптимизации функционально-структурных описаний дискретных устройств / И. П. Логинова // Компьютерное проектирование в электронике = Electronic Design Automation: сб. трудов Междунар. науч.-практ. конф. (Республика Беларусь, г. Минск, 28 ноября 2024 г.) / редкол.: В. Р. Стемпицкий [и др.]. – Минск: БГУИР, 2024. – С. 108–111.

[8] Логинова, И.П. Анализ быстродействия при реализации параллельных вычислений для решения задач оптимизации в системе логического проектирования / И. П. Логинова // IX Международная научно-практическая конференция «BIG DATA and Advanced Analytics» (BIG DATA 2023): Сборник научных статей в 2 ч. Ч. 1, (Республика Беларусь, Минск, 17-18 мая 2023 г.). – Минск: БГУИР, 2023. – С. 365–378.

[9] E Ayguadé, E. The design of Open MP tasks / E. Ayguade [et al.] // IEEE Transactions on Parallel and Distributed Systems. – 2009. – Vol. 20, no. 3. – P. 404–418.

[10] Торопов, Н. Р. Преобразование многоярусной комбинационной сети в двухъярусную / Н. Р. Торопов // Логическое проектирование : сб. науч. тр. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2000. – Вып. 5. – С. 4–14.

[11] Логинова И.П. Организация параллельных вычислений на базе программного комплекса PLUS /И.П. Логинова // XII Международная научно-техническая конференция «Информационные технологии в промышленности, логистике и социальной сфере» (ITI*2023): тезисы докладов, 21-22 сентября 2023 г. – Минск : ОИПИ НАН Беларуси, 2023. – С. 103–107.

[12] Черемисинова, Л.Д. Программные средства верификации описаний комбинационных устройств в процессе логического проектирования / Л.Д. Черемисинова, Д.Я. Новиков // Программная инженерия, 2013, № 7, с. 8–15.

[13] Логинова, И.П. Формальная верификация результатов выполнения параллельных программ логической оптимизации / И. П. Логинова // Информационные технологии и системы 2022 (ИТС 2022) – Information Technologies and Systems 2022 (ITS 2022): материалы междунар. науч. конф. (Республика Беларусь, Минск, 30 октября 2021 года). – Минск: БГУИР, 2021. – С. 85-86.

[14] Шлее, М. Qt 5.10. Профессиональное программирование на C++ / М. Шлее. – СПб.: БХВ-Петербург, 2018. – 1072 с.

[15] Логинова, И. П. Визуализация трассы параллельной программы в системе проектирования / И. П. Логинова // Информационные технологии и системы 2024 (ИТС 2024) – Information Technologies and Systems 2024 (ITS 2024): материалы междунар. науч. конф., Минск, 20 нояб. / редкол. : Л. Ю. Шилин [и др.]. – С. 89 – 90

THE BUILDING A GRAPHICAL SHELL FOR PERFORMING EXPERIMENTS BASED ON A SYSTEM FOR LOGICAL OPTIMIZATION OF FUNCTIONAL AND STRUCTURAL DESCRIPTIONS OF DISCRETE DEVICES

I.P. Loginova

*Senior researcher, United Institute of Informatics Problems of the National Academy of Sciences of
Belarus, PhD, associate professor*

Abstract. The article describes the functioning of a graphical interface designed to visually form the scheme of a computational experiment in the form of chains of actions proposed by the interface control elements. The visual environment of the software allows making for a variety of experiments using a set of design operations of a system for logical optimization of functional and structural descriptions of discrete devices FLC-2. By means of the graphical interface, an experimental scheme is formulated, in the process of which a choice is made: programs and objects participating in the experiment, User runs the experiment scheme, monitors the result with built-in visualization and verification tools. The software includes a bundling subsystem in which interface tools generate parallel tasks for logical optimization in objects assembled into packages.

Keywords: multicore system, parallel process, task scheduler, speedup of performance, logical optimization, visualisation.