

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ  
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN 0135-1710

**АВТОМАТИЗИРОВАННЫЕ  
СИСТЕМЫ УПРАВЛЕНИЯ И  
ПРИБОРЫ АВТОМАТИКИ**

**Всеукраинский межведомственный  
научно-технический сборник**

**Основан в 1965 г.**

**Выпуск 166**

Харьков  
2014

## СОДЕРЖАНИЕ

<b>МИХЕЕВ И.А., СИЗОВА Н.Д.</b> КРОССПЛАТФОРМЕННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ТЕХНОЛОГИИ СТРОИТЕЛЬНЫХ МАТЕРИАЛОВ.....	4
<b>ЛАРЧЕНКО Л.В., ВАРЕЦА В.В., ЛАРЧЕНКО Б.Д., МАКАРЕНКО А.Н.</b> СРАВНИТЕЛЬНАЯ ОЦЕНКА БЫСТРОДЕЙСТВИЯ РЕГИСТРОВ-КОМПАКТОРОВ СИНХРОННОГО И АСИНХРОННОГО ТИПОВ.....	9
<b>КАНЦЕДАЛ С.А., КОСТИКОВА М.В.</b> ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ ДЛЯ ЗАДАЧИ КОММИВОВАЖЕРА.....	15
<b>МЕЛЬНИК О.С., КОЗАРЕВИЧ В.О., РОМАНЮК В.Ю.</b> ПРОЕКТУВАННЯ ЛОГІЧНО-АРИФМЕТИЧНИХ НАНОПРИСТРОЇВ.....	21
<b>ЧЕРНА Т.І.</b> ВИКОРИСТАННЯ АДАПТИВНИХ ОНТОЛОГІЙ В ЗАДАЧІ КВАЗІРЕФЕРУВАННЯ ТЕКСТОВИХ ДОКУМЕНТІВ.....	27
<b>ФОМОВСКИЙ Ф.В.</b> МОДЕЛИРОВАНИЕ ПРОЦЕССА РАЗРЯДА ПЕРЕЗАРЯЖАЕМЫХ ИСТОЧНИКОВ ЭНЕРГИИ С ПОСТОЯННОЙ МОЩНОСТЬЮ.....	32
<b>ЧАЛЫЙ С.Ф., ДЕМЧЕНКО А.А.</b> ИСКЛЮЧЕНИЕ ЭФФЕКТА ОРЕОЛА ПРИ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ СВЕТОФОРНЫХ СИГНАЛОВ.....	38
<b>БАРАННИК В.В., ОТМАН ШАДИ О.Ю., ХАХАНОВА А.В.</b> МЕТОД РЕКОНСТРУКЦИИ ТРАНСФОРМАНТ В ДИАГОНАЛЬНО-НЕРАВНОМЕРНОМ БАЗИСЕ ОСНОВАНИЙ НА ОСНОВЕ ДЕКОДИРОВАНИЯ НЕРАВНОМЕРНЫХ КОДОГРАММ.....	42
<b>БОБРОВА Н.Л.</b> ОБОСНОВАНИЕ ИСПОЛЬЗОВАНИЯ КОМПЛЕКСА ДИАГНОСТИЧЕСКИХ МЕТОДИК ДЛЯ ОЦЕНКИ ПСИХОФИЗИОЛОГИЧЕСКОГО СОСТОЯНИЯ ЧЕЛОВЕКА.....	48
<b>БОНДАРЕНКО Н.А.</b> ПОДСИСТЕМА КОНТРОЛЯ ЗНАНИЙ ПО ТЕМЕ «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ DELPHI. РАБОТА С ЗАПИСЯМИ».....	53
<b>BAGHDADI AMMAR AWNI ABBAS (БАГХДАДИ АММАР АВНИ АББАС), ХАХАНОВ В.И., ЛИТВИНОВА Е.И.</b> МЕТОДЫ АНАЛИЗА И ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ (АНАЛИТИЧЕСКИЙ ОБЗОР).....	59
<b>РЕФЕРАТИ</b> .....	75
<b>ПРАВИЛА</b> ОФОРМЛЕНИЯ РУКОПИСЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО СБОРНИКА.....	79

## **КРОССПЛАТФОРМЕННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ТЕХНОЛОГИИ СТРОИТЕЛЬНЫХ МАТЕРИАЛОВ**

---

Представляются результаты разработки программного обеспечения для решения задач технологии строительных материалов. В качестве примера выбирается задача проектирования состава бетона. Обосновывается выбор языка программирования и используемых модулей. Описывается структура, элементы кроссплатформенного интерфейса, а также результаты тестирования компьютерной программы в различных операционных системах. Апробируются технологии разработки кроссплатформенного программного обеспечения для решения задач в технологии строительных материалов.

### **1. Введение**

Рынок информационных технологий сегодня – это динамически развивающаяся система с постоянным ростом количества ключевых игроков. В первую очередь это касается операционных систем. Продукты корпораций Microsoft хоть и продолжают господствовать, в том числе и на рынке операционных систем, но конкуренты с каждым годом расширяют свою аудиторию. В первую очередь это MacOS X и Ubuntu. Исходя из этого, актуальным вопросом в области разработки программного обеспечения является кроссплатформенное программирование – технология разработки программного обеспечения, работающего более чем на одной аппаратной платформе и/или операционной системе. Кроссплатформенное программирование позволяет реализовывать возможности компьютерных программ и демонстрировать графический интерфейс пользователя вне зависимости от используемого аппаратного и операционного обеспечения, а разработка таких программ при практическом полном отсутствии литературы является актуальной и сложной задачей.

Цель исследования – разработка структуры интеллектуальной системы, а также кроссплатформенного программного обеспечения для проектирования состава бетона – важной технологической задачи строительного материаловедения.

### **2. Основной материал**

Технологии строительных материалов с точки зрения использования информационных систем можно описать следующими характеристиками:

- сложная формализация моделируемых процессов;
- огромное количество данных, разнообразных по своим типам и структуре: характеристики используемых материалов; алгоритмы методологий; нормативные регламентирующие документы;
- разрозненность аппаратного и программного обеспечения;
- различная квалификация конечных пользователей.

Для реализации задачи проектирования состава бетона (одной из ключевых задач технологии бетона – основного конструкционного строительного материала) коллективом авторов разработан алгоритм принятия рецептурных решений, учитывающий множество характеристик исходных материалов и других конструктивно-технологических факторов [1]. Для автоматизации расчетов разработана интеллектуальная система управления составом и свойствами бетона.

Важным вопросом при разработке кроссплатформенного программного обеспечения является выбор языка программирования. При решении этого вопроса необходимо учитывать множество факторов: компиляция или интерпретация, скорость работы и скорость разработки, открытость исходного кода или проприетарность.

В ходе разработки интеллектуальной системы выделены основные требования к программному обеспечению: возможность адаптации алгоритмов под конкретные производственные условия; наличие дружелюбного графического интерфейса; полная фриварность используемых технологий.

### 3. Результаты исследования

Исходя из выдвигаемых требований, в качестве базового языка программирования был выбран Python – многопарадигмальный высокоуровневый язык программирования, ориентированный на повышение производительности разработчика и читаемости кода [2]. В качестве дополнительных библиотек использованы Tkinter (стандартная кросс-платформенная графическая библиотека Python на основе средств Tcl/Tk), PIL (библиотека для работы с растровой графикой), ReportLab (библиотека для форматирования и генерации отчетов) [3].

Интеллектуальная система имеет блочную структуру, что нашло отражение в модульной структуре программного обеспечения. Каждая методология проектирования состава бетона описана как класс в отдельном функциональном модуле (рис. 1). Интерпретируемость языка позволяет адаптировать алгоритм и вносить необходимые коррективы [4].

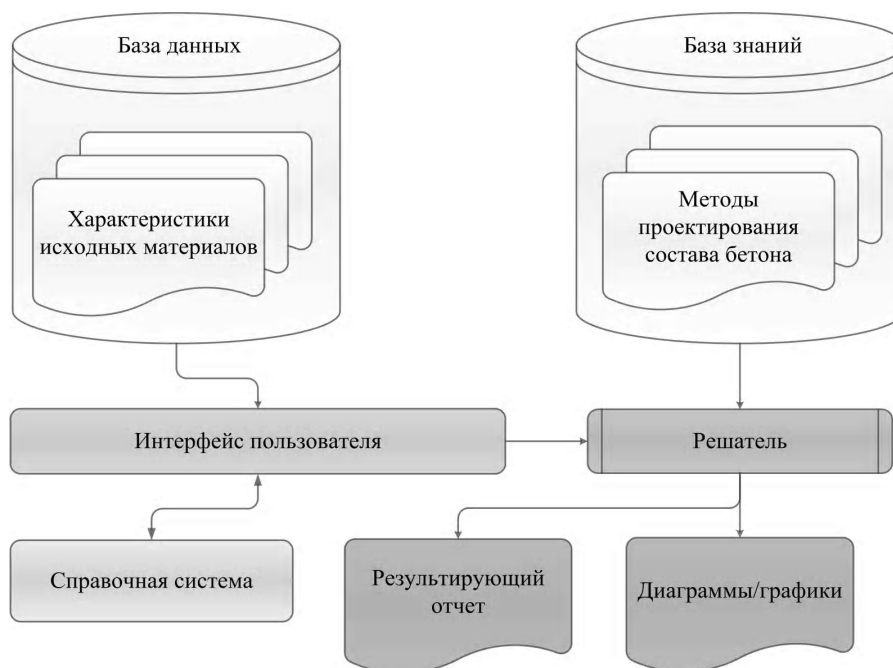


Рис. 1. Структура интеллектуальной системы

Интерфейс пользователя разрабатывался с обеспечением принципов комфортности применения, высокой функциональности и поддержки пользователя на всех этапах решения задачи проектирования состава бетона.

На главной форме, на вкладке «Исходные данные» с помощью имеющихся элементов управления вводятся данные, описывающие характеристики составляющих бетон материалов, проектные требования к бетонной смеси и бетону, тип и условия эксплуатации конструкции, а также указывается основной критерий (рис. 2). Результаты расчетов выводятся на одноименной вкладке в виде одностраничного отчета, экспорт которого возможен в один из трех форматов: \*.txt; \*.csv; \*.pdf (рис. 3). На третьей вкладке представлена диаграмма набора прочности (рис. 4) – важная технологическая информация, позволяющая принимать решения о снятии опалубки, нагрузке конструкции и т.д. Интерфейс пользователя, результирующий отчет, справочная информация реализованы на трех языках: русском, украинском и английском.

Проведено тестирование программного обеспечения в операционных системах Windows 7 (см. рис. 2), Ubuntu 14.04 (рис. 5), MacOS X (рис. 6).

Программное обеспечение в каждой из платформ сохраняет полную функциональность, но имеет свои особенности, связанные в основном с отображением элементов графического интерфейса. Из видимых недостатков – кегль шрифтов отображается по-разному – этот параметр необходимо вынести в модуль настроек программы. Необходимо отметить обязательную установку всех дополнительных модулей для каждой из платформ.

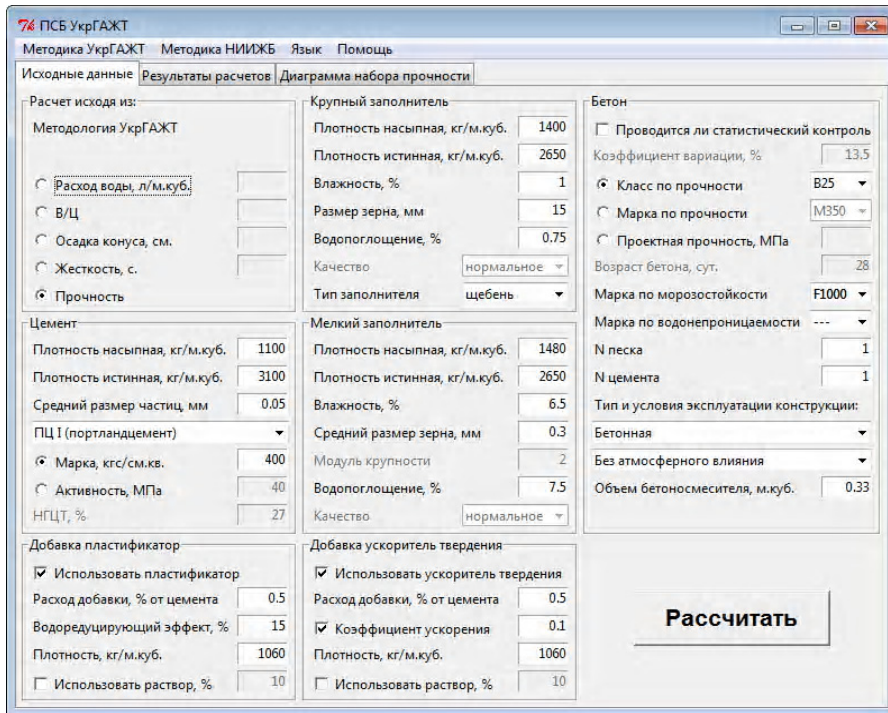


Рис. 2. Вкладка «Исходные данные» в Windows 7

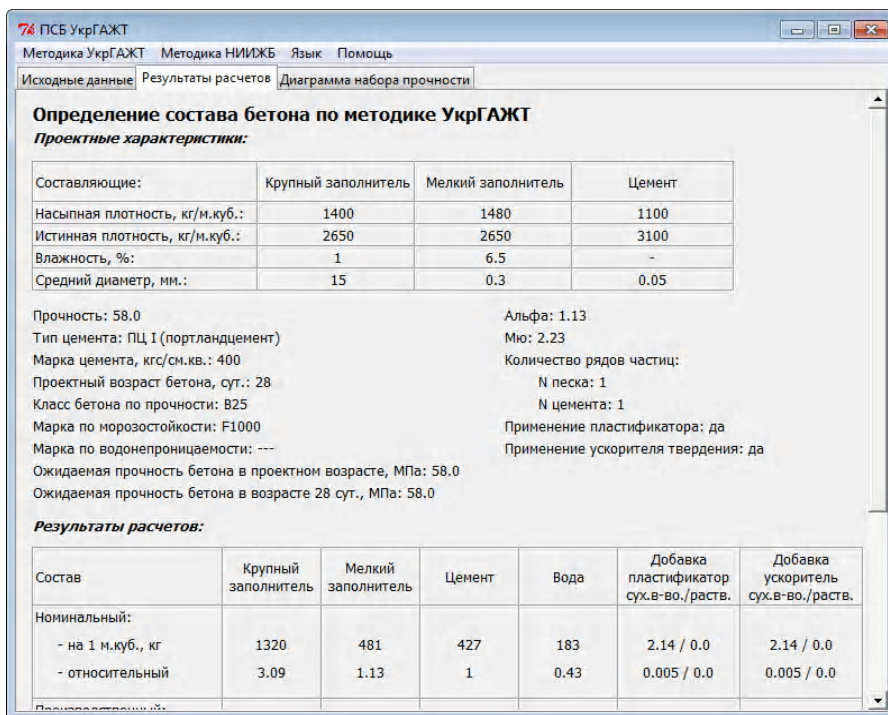


Рис. 3. Вкладка «Результаты расчетов»

Трудоемкость процесса разработки обусловлена отсутствием привычных для разработчиков «drag-and-drop» билдеров для разработки графического интерфейса – необходимо «вручную» описывать свойства каждого элемента управления, его поведение и реакцию на события.

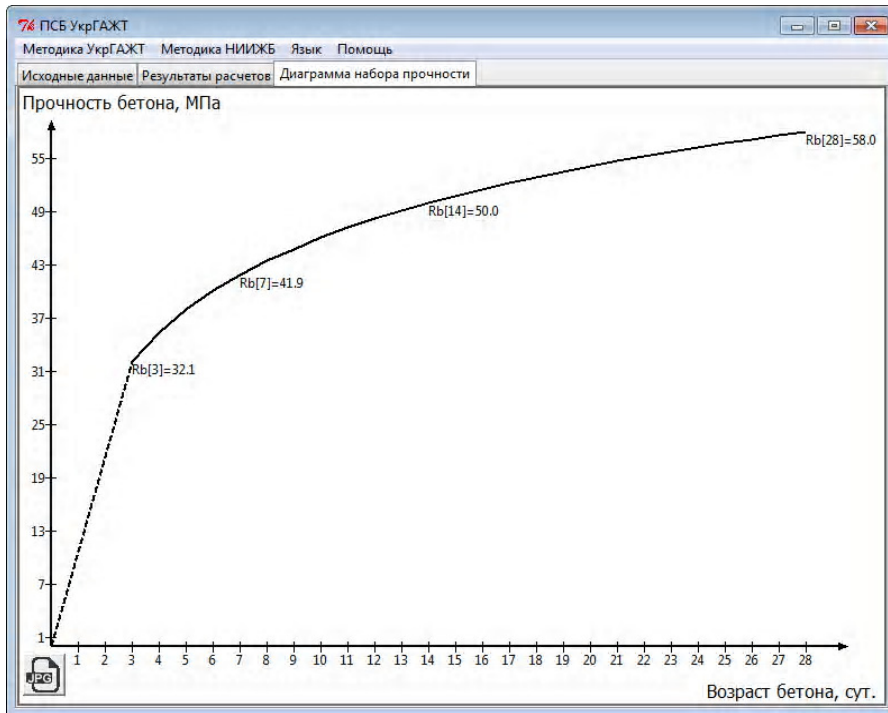


Рис. 4. Вкладка «Диаграмма набора прочности»

Section	Parameter	Value
Расчет исходя из:	Методология	УкрГАЖТ
	Расход воды, л/м.куб.	
	В/Ц	
	Осадка конуса, см.	
Цемент	Плотность насыпная, кг/м.ку	1100
	Плотность истинная, кг/м.ку	3100
	Средний размер частиц, мм	0.05
	Марка, кгс/см.кв.	400
Крупный заполнитель	Плотность насыпная, кг/м.ку	1400
	Плотность истинная, кг/м.ку	2650
	Влажность, %	1
	Размер зерна, мм	15
Мелкий заполнитель	Плотность насыпная, кг/м.ку	1480
	Плотность истинная, кг/м.ку	2650
	Влажность, %	6.5
	Средний размер зерна, мм	0.3
Бетон	Класс по прочности	B25
	Марка по прочности	M350
	Проектная прочность, МПа	
	Возраст бетона, сут.	28

Рис. 5. Вкладка «Исходные данные» в Ubuntu 14.04

Несмотря на перечисленные неудобства, весь код программы (более 5000 строк) остается неизменным, что позволяет существенно экономить время при разработке различных версий программы под определенные платформы.

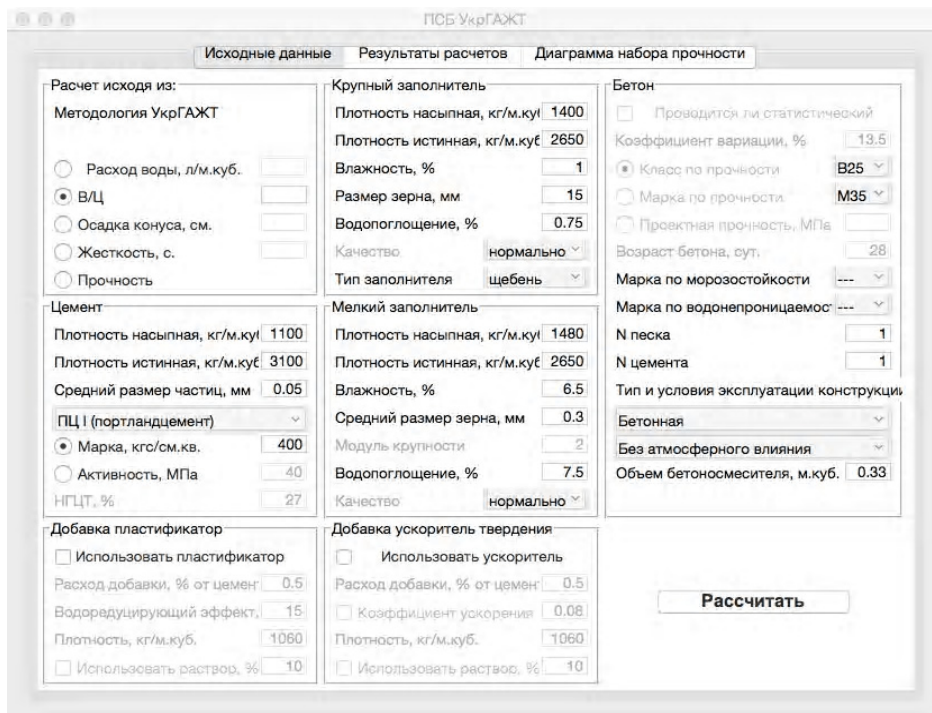


Рис. 6. Вкладка «Исходные данные» в MacOS X

#### 4. Выводы

В ходе создания прототипа интеллектуальной системы управления составом и свойствами бетона были использованы современные технологии и подходы разработки программного обеспечения, реализованы авторские алгоритмы и обеспечена кроссплатформенность интерфейса пользователя.

С точки зрения технологии строительных материалов компьютерная программа для решения задачи проектирования состава бетона с кроссплатформенным интерфейсом является пионерным и уникальным проектом. Апробированные технологии разработки кроссплатформенного программного обеспечения могут быть применены для решения многих задач строительного материаловедения.

**Список литературы:** 1. *Плугин А.А.* Программное обеспечение системы проектирования состава бетона для конструкций и сооружений железных дорог // А.А. Плугин, О.А. Калинин, Н.Д. Сизова, И.А. Михеев // Технологический аудит. 2013, № 6/1(14). С. 38-40. 2. *Россум Г., Дрейк Ф.Л.Дж., Откидач Д.С.* и др. Язык программирования Python. 2001. 454 с. [электронный ресурс]. Режим доступа: [www. URL: http://rus-linux.net/MyLDP/BOOKS/python.pdf](http://rus-linux.net/MyLDP/BOOKS/python.pdf) 3. *Grayson J.E.* Python and Tkinter Programming. Manning Publications Co, 2000. 688 p. 4. *Сизова Н.Д.* Особенности создания программного обеспечения для проектирования состава бетона // Н.Д. Сизова, И.А. Михеев // Восточно-европейский журнал передовых технологий. 2013. № 6/2(66). С. 27-31.

*Поступила в редколлегию 22.02.2014*

**Михеев Иван Андреевич**, канд. техн. наук, доцент кафедры экономической кибернетики и информационных технологий ХНУСА. Научные интересы: математическое моделирование сложных процессов и систем, методы оптимизации строительного материаловедения, современные информационные технологии в строительной отрасли. Адрес: Украина, 61002, ул. Сумская, 40, тел. (057) 706-20-49, E-mail: [i.a.mikheev@gmail.com](mailto:i.a.mikheev@gmail.com)

**Сизова Наталия Дмитриевна**, д-р физ.-мат. наук, профессор кафедры экономической кибернетики и информационных технологий ХНУСА. Научные интересы: вычислительная информатика, математическое и компьютерное моделирование, информационные технологии. Адрес: Украина, 61002, ул. Сумская, 40, тел. (057) 706-20-49, E-mail: [sizova@ukr.net](mailto:sizova@ukr.net)

## СРАВНИТЕЛЬНАЯ ОЦЕНКА БЫСТРОДЕЙСТВИЯ РЕГИСТРОВ-КОМПАКТОРОВ СИНХРОННОГО И АСИНХРОННОГО ТИПОВ

Рассматривается структура и функционирование многоразрядных регистров-компакторов двоичных кодов синхронного и асинхронного типов. Анализируется схемотехническая реализация различных типов регистров-компакторов и выполняется сравнительная оценка их быстродействия по числу тактов, необходимых для сжатия кода.

### 1. Постановка задачи

Известны и широко применяются в ВТ и автоматике регистры сдвига на RS или JK-триггерах, причем прямой и инверсный выходы каждого JK-триггера, кроме последнего, соединены соответственно с J- и K- входами следующего JK-триггера [1]. Такие регистры сдвига могут работать в трех режимах: режиме сдвига последовательного кода на заданное число тактов, режиме хранения кода и режиме приема кода за  $n$  – тактов, где  $n$  – разрядность регистра сдвига.

Недостатком известных регистров сдвига являются их ограниченные функциональные возможности, обусловленные выполнением только одной основной функции – сдвига информации. В ряде случаев в устройствах обработки и контроля цифровой статистической и диагностической информации требуется получение сведений о числе единиц в кодовой комбинации.

Подобное расширение функциональных возможностей достигнуто в [2,3] за счет введения дополнительной функции – сжатия всех единиц кодовой комбинации вправо без их потери. Такой регистр будем называть регистром-компактором.

Основными задачами работы являются:

- рассмотрение структуры и функционирования различных типов регистров-компакторов;
- сравнительная оценка времени сжатия кода по максимальному числу тактов в функции от разрядности  $n$  регистра - компактора;
- формулировка рекомендаций и задач по использованию регистров с функцией сжатия кода.

### 2. Регистр-компактор синхронного типа

Схема синхронного регистра-компактора двоичных кодов из [2] представлена на рис.1.

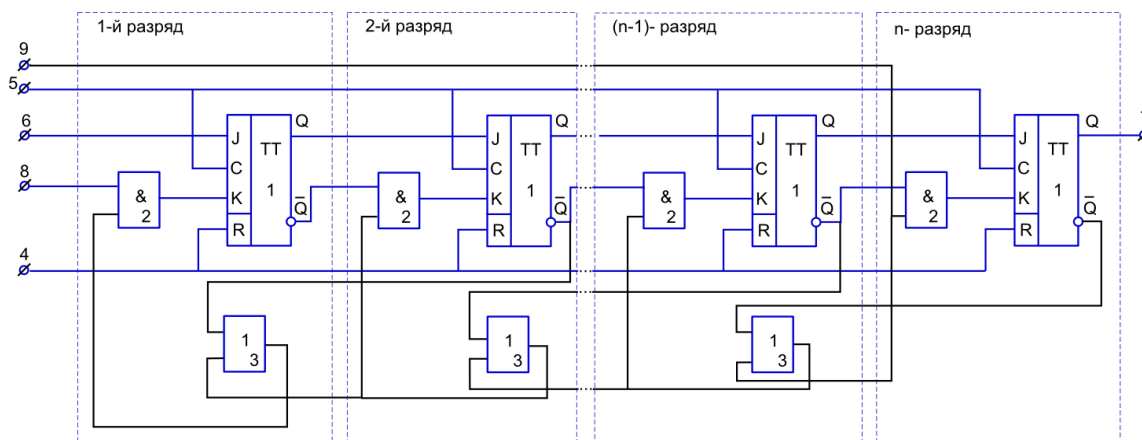


Рис.1. Регистр-компактор двоичных кодов

Регистр содержит в каждом разряде двухступенчатый JK - триггер 1, элемент И 2 и в каждом разряде, кроме последнего, элемент ИЛИ 3. Прямой выход Q JK – триггера

каждого разряда, кроме последнего, соединен с J- входом разряда, R- и C- входы JK – триггеров всех разрядов соответственно объединены и являются входами установки в 0 4 и синхронизации 5 регистра – компактора, J- вход JK – триггера первого разряда 6 является прямым входом регистра – компактора, а прямой выход JK – триггера последнего разряда 7 является прямым выходом регистра - компактора. Выход элемента И 2 каждого разряда соединен с K- входом JK – триггера данного разряда, первый вход элемента И первого разряда является инверсным входом 8 регистра-компактора, а первый элемента И каждого из остальных разрядов соединен с инверсным входом JK – триггера предыдущего разряда, второй вход элемента И каждого разряда, кроме последнего, соединен с выходом элемента ИЛИ 3 данного разряда, а второй вход элемента И последнего разряда является управляющим входом 9. Первый и второй входы элемента ИЛИ каждого разряда соединены соответственно с инверсным выходом JK – триггера и со вторым входом элемента И последующего разряда.

Рассмотрим работу регистра в различных режимах. При единичном значении сигнала управления на входе 9 регистр работает в режиме сдвига кода вправо. При этом на вторых входах каждого из элементов И 2 устанавливается единичное значение сигнала. Входная информация в виде парафазного кода поступает на входы 6 и 8. Заполнение n- разрядного регистра информацией происходит за время действия n импульсов сдвига на входе синхронизации 5.

При нулевом значении сигнала управления на входе 9 регистр работает в режиме сжатия информации вправо. Под действием импульсов сдвига, подаваемых на вход синхронизации 5, происходит сдвиг кода вправо до тех пор, пока не запишется единица в последний n-й разряд. После этого происходит сдвиг кода только в первых (n-1)-м разрядах до заполнения единицей (n-1)-го разряда. Записанные в n-м, (n-1)-м и т.д. разрядах единицы остаются в соответствующих разрядах, поскольку на K - входах JK – триггеров этих разрядов устанавливается нулевое значение. Таким образом, после подачи в общем случае (n-1)-го импульса сдвига (когда исходная информация представляет собой одну единицу в крайнем левом разряде) все единицы исходной информации располагаются плотно справа.

Например, если при единичном значении сигнала на входе 9 после подачи восьми импульсов сдвига в восьмиразрядном регистре запишется код 10100100, то при нулевом значении сигнала на входе управления 9 после подачи очередных пяти импульсов сдвига получим такие последовательности кодов: 01010010, 00101001, 00010101, 00001011, 00000111. При дальнейшей подаче импульсов сдвига состояние регистра не меняется. При установке единичного значения сигнала на входе управления 9 сжатая информация выводится из регистра.

Из рис. 1 найдем, что  $K_3 = z \cdot \bar{Q}_2$ ;  $J_3 = Q_2$ ;  $K_3 = z \cdot \bar{Q}_1 \vee \bar{Q}_1 \cdot Q_3 = \bar{Q}_1(z \vee Q_3)$ ;  $J_2 = Q_1$ .

При рассмотрении регистра-компактора (см. рис.1) мы ограничились числом разрядов n=3. Если обобщить полученные формулы для случая n разрядов, то получим для j=0,...,(n-1) следующие формулы (1):

$$\begin{aligned} J_n &= Q_{n-1}; K_n = z \cdot \bar{Q}_{n-1}; J_{n-1} = Q_{n-2}; K_{n-1} = \bar{Q}_{n-2} \cdot (z \vee \bar{Q}_n); J_{n-2} = Q_{n-1}; \\ K_{n-2} &= \bar{Q}_{n-1} \cdot (z \vee \bar{Q}_n \vee \bar{Q}_{n-1}) = \bar{Q}_{n-3} \cdot ((z \vee \bar{Q}_n) \vee \bar{Q}_{n-1}); J_{n-3} = Q_{n-4}; \\ K_{n-3} &= \bar{Q}_{n-4} \cdot (z \vee \bar{Q}_n \vee \bar{Q}_{n-1} \vee \bar{Q}_{n-2}) = \bar{Q}_{n-4} \cdot (((z \vee \bar{Q}_n) \vee \bar{Q}_{n-1}) \vee \bar{Q}_{n-2}); \\ K_{n-j} &= \bar{Q}_{n-j-1} \cdot (...(z \vee \bar{Q}_n) \vee \bar{Q}_{n-1}) \vee ... \vee \bar{Q}_{n-j+2}); J_{n-j} = Q_{n-j-1}. \end{aligned} \quad (1)$$

Существуют и другие схемные решения регистров компакторов, отличающиеся друг от друга возможностью уплотнения не только единиц, но и нулей, возможностью сдвига как вправо, так и влево, а также возможностью уплотнения в заданном направлении [4-6].

### 3. Регистр-компактор асинхронного типа

Схема асинхронного регистра-компактора двоичного кода из [3] приведена на рис.2.

Регистр содержит группу из n-1 двухвходовых элементов И 1, элемент ИЛИ 2, элемент И 3, в каждом разряде двухступенчатый JK-триггер 4, элемент И 5 и в каждом разряде,

кроме последнего, элемент ИЛИ 6. Прямой выход JK-триггера каждого разряда, кроме последнего, соединен с J-входом JK-триггера последующего разряда, R- и С-входы JK-триггеров всех разрядов соответственно объединены и являются входами установки в 0 7 и синхронизации 8 регистра сдвига, J-вход 9 JK-триггера первого разряда является прямым входом регистра сдвига, а прямой выход 10 JK-триггера последнего разряда является прямым выходом регистра сдвига. Регистр сдвига в зависимости от значения сигнала управления, подаваемого на управляющий вход 12, может работать в двух режимах - сдвига и сжатия информации. При единичном значении сигнала управления на входе 12 регистр работает в режиме сдвига информации. При этом на вторых входах каждого из элементов И 5 устанавливается единичное значение сигнала. Входная информация в виде парафазного кода поступает на входы 9 и 11. Единичный сигнал с выхода элемента ИЛИ 2 разрешит прохождение тактовых сигналов с входа 13 элемента И 3 на вход синхронизации 8 регистра сдвига. Заполнение n – разрядного регистра информацией происходит за время действия n импульсов сдвига на входе синхронизации 8.

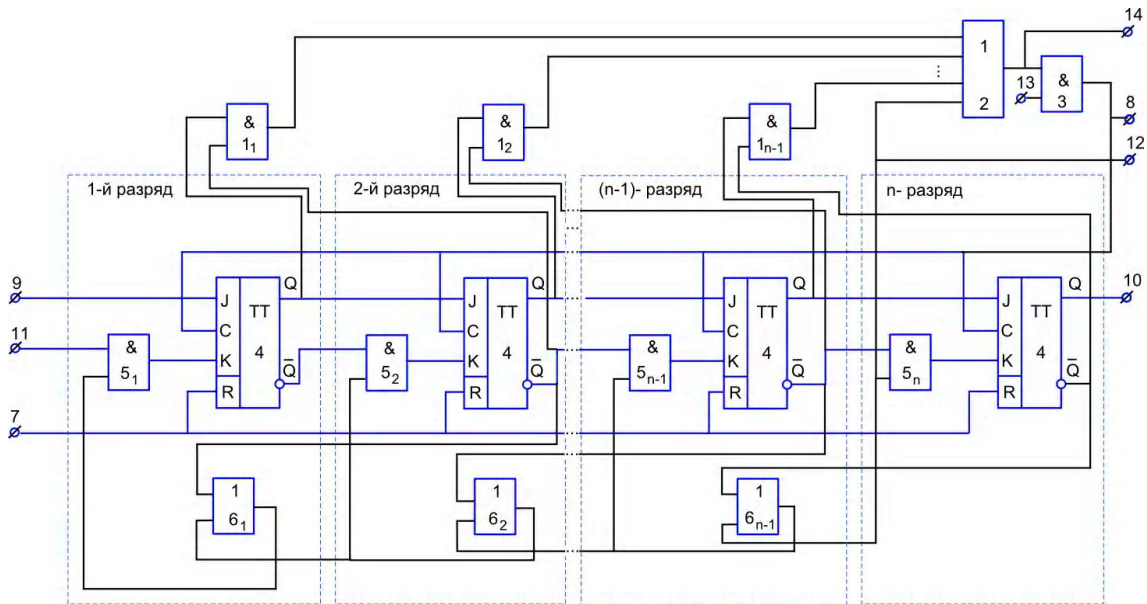


Рис. 2. Регистр-компактор асинхронного типа

При нулевом значении сигнала управления на входе 12 регистр работает в режиме сжатия информации. Если содержимое регистра не соответствует сжатой кодовой комбинации, то с выхода одного из элементов И группы 1 элементов И сформированный единичный сигнал через элемент ИЛИ 2 разрешит прохождение тактовых импульсов через элемент И 3 на вход синхронизации 8 регистра. Под действием импульсов сдвига происходит сдвиг кода вправо до тех пор, пока не запишется единица в последний n-й разряд. После этого происходит сдвиг кода только в первых (n-1)-м разрядах до заполнения единицей (n-1)-го разряда. Записанные в n-м, (n-1)-м разрядах единицы остаются в соответствующих разрядах, поскольку на K-входах JK-триггеров этих разрядов устанавливается нулевое значение. Как только содержимое регистра будет соответствовать сжатой кодовой комбинации, т.е. когда все единицы исходной информации будут расположены плотно справа, на выходах элементов И группы 1 сформируется нулевой сигнал, который через элемент ИЛИ 2 запретит прохождение тактовых импульсов на вход синхронизации 8 регистра сдвига. Нулевой сигнал с выхода элемента ИЛИ 2 сигнализирует о готовности регистра сдвига к приему и сжатию следующей комбинации. Время сжатия информации в синхронном регистре-компакторе составляет (n-1) такт, а в рассмотренном асинхронном – (n- m) тактов, где m – число нулей, отстоящих справа от старшей единицы.

Для n=4 функции дешифрации неуплотненных комбинаций от 1-го до 3-го разрядов могут быть представлены в виде выражений:

$$y_1^1 = Q_1 \cdot \overline{Q_2}; \quad y_1^2 = Q_2 \cdot \overline{Q_3}; \quad y_1^3 = Q_3 \cdot \overline{Q_4}. \quad (2)$$

При  $n=8$  добавляются формулы:

$$y_1^4 = Q_4 \cdot \overline{Q_5}; \quad y_1^5 = Q_5 \cdot \overline{Q_6}; \quad y_1^6 = Q_6 \cdot \overline{Q_7}; \quad y_1^7 = Q_7 \cdot \overline{Q_8}. \quad (3)$$

Для общего случая числа разрядов  $n$  функции дешифрации неуплотненных комбинаций, реализуемые с помощью схем И 1, описываются выражением:

$$y_1^i = Q_i \cdot \overline{Q_{i+1}}, \quad i = \overline{1, n-1}, \quad (4)$$

а функция наличия неуплотненного кода в регистре-компакторе (см. рис.2), вычисляемая схемой ИЛИ 2, определяется по формуле:

$$y_2 = \bigcup_{i=1}^{i=n-1} Q_i \cdot \overline{Q_{i+1}} = \bigcup_{i=1}^{i=n-1} y_1^i. \quad (5)$$

Схема И 3, разрешающая прохождение синхроимпульсов с на вход сдвига регистра, реализует выражение:

$$y_3 = c \cdot y_2. \quad (6)$$

#### 4. Быстродействующий регистр-компактор синхронного типа

Схема быстродействующего регистра-компактора с временем уплотнения комбинации в один такт приведена на рис.3.

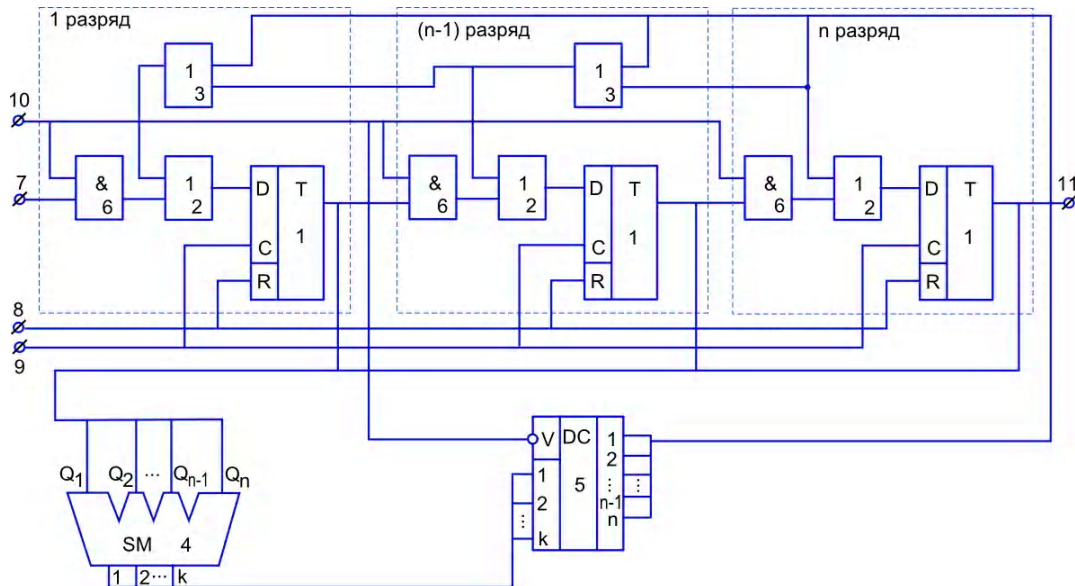


Рис. 3. Регистр-компактор с временем уплотнения в один такт

Регистр состоит из группы  $n$  CDR-триггеров 1 состояния, первого элемента ИЛИ 2 и элемента И 6 в каждом разряде, второго элемента ИЛИ 3 в каждом разряде, кроме последнего, комбинационного сумматора 4 количества единиц неуплотненного двоичного кода, дешифратора 5 для преобразования позиционного двоичного кода единиц в унитарный код единиц, второго элемента ИЛИ 3 в каждом разряде, кроме последнего. При единичном значении сигнала на входе управления режимом работы 10 и поочередной подаче тактовых сигналов на вход 9 и разрядов входного кода на информационный вход 7 за  $n$  тактов происходит запись последовательного кода в регистр.

При нулевом значении сигнала на входе управления режимом работы 10 регистр работает в режиме уплотнения кода. Логическая связь между прямым выходом CDR-триггера предыдущего разряда и информационным D-входом триггера разряда, который рассматривается, при этом отсутствует. Позиционный код количества единиц  $i$  с выхода комбинационного сумматора 4 с помощью дешифратора 5 преобразуется в сигнал единицы на выходе  $i$ , который далее подается на первый вход второй схемы ИЛИ 3 разряда  $i$  и далее по цепи вторых схем ИЛИ 3 на первые входы первых схем ИЛИ 2 в разрядах  $i, i-1, i-2, i-3,$

... ,2, 1. После подачи одного импульса на вход синхронизации 9 в регистр будет записан двоичный код, уплотненный влево. Например, после подачи в регистр - компактор при  $n=7$  кода 1001010, затем переключения регистра в режим уплотнения и подачи одного импульса на вход синхронизации 9 получим в регистре кодовую комбинацию 1110000, которая через выход 11 путем сдвига может быть выведена из регистра - компактора.

Быстродействие регистра – компактора в режиме уплотнения кода не зависит от разрядности и составляет один такт.

### 5. Сравнительная оценка быстродействия регистров-компакторов

Выполним сравнение синхронного и асинхронного регистров-компакторов по числу тактовых импульсов, необходимых для уплотнения всех  $n$ -разрядных комбинаций. Так как для уплотнения любой комбинации требуется  $n-1$  такт, то суммарное число тактов на уплотнение всех комбинаций в синхронном регистре определяется выражением:

$$N_c = (n - 1) \cdot 2^n . \quad (7)$$

В асинхронном регистре-компакторе число тактов, затрачиваемых на уплотнение кодовой комбинации, равно числу нулей  $m$ , находящихся справа от старшей единицы. Выражение для суммарного числа тактов уплотнения всех кодовых комбинаций описывается формулой:

$$N_a = \sum_{i=1}^{i=2^n} m_i . \quad (8)$$

Результаты вычислений по формуле (7) и статистического анализа для разрядностей  $n=2-6$  приведены в таблице. Здесь также приведены расчетные данные абсолютной экономии  $\varepsilon$  (по числу тактов сжатия), вычисленные по формуле (9), и относительной (в процентах) экономии, рассчитанной по формуле (10) :

$$\varepsilon = N_c - N_a ; \quad (9)$$

$$\gamma = \frac{\varepsilon}{N_c} . \quad (10)$$

n	2	3	4	5	6
$N_c$	4	16	48	128	320
$N_a$	1	5	17	49	120
$\varepsilon$	3	11	31	79	200
$\gamma, \%$	75	68,75	64,58	61,7	62,5

Анализ таблицы и формул показывает, что асинхронный регистр-компактор по сравнению с синхронным обеспечивает существенную экономию числа тактов на сжатие кодовых комбинаций (от 60 до 70 процентов), а быстродействующий регистр-компактор (см. рис.3) уже при  $n=4$ ;  $n=5$  и  $n=6$  требует для сжатия всего множества кодов соответственно 16, 32 и 64 такта, что указывает на его более высокое быстродействие, но для его реализации требуются и более высокие аппаратные затраты. Для использования тех или иных вариантов регистров-компакторов в различных устройствах обработки и контроля цифровой информации также необходима разработка VHDL-моделей указанных регистров, что позволит при наличии моделей автоматизировать этапы системного проектирования соответствующего регистра-компактора и ускорить его проектирование и реализацию на СБИС.

### Выводы

1. Рассмотрена структура и функционирование трех различных схемных реализаций регистров-компакторов двоичных кодов синхронного и асинхронного типов на JK- и DC-триггерах .

2. Получены аналитические выражения и статистические данные затрат времени (по числу тактов сдвига) на уплотнение кодовых комбинаций для различных вариантов регистра-компактора в функции от числа его разрядов.

3. Выполнена сравнительная оценка временных затрат на сжатие (компрессию) кодов, что позволяет в зависимости от требований к быстродействию и аппаратурным затратам выбрать соответствующую схемную реализацию регистра-компактора.

*Научная новизна* работы состоит в получении аналитических выражений для детального анализа числа тактов сдвига на компрессию двоичных кодов в каждой из схем регистра-компактора, что позволило выполнить сравнительную оценку быстродействия различных его реализаций.

*Практическая значимость* результатов заключается в возможности проектирования регистра-компактора с учетом основных критериев оптимизации (быстродействия и аппаратурных затрат), что делает также целесообразным получение его моделей на языке VHDL в целях ускорения их проектирования и реализации на СБИС.

**Список литературы:** 1. Майоров С.А., Новиков Г.И. Структура электронных вычислительных машин. Л., 1979, 384 с. 2. А.С. 1049978 G11C 19/00. Регистр сдвига / Н.Я.Какурин, Ю.К. Кирьяков // Открытия, изобретения. 1983. №39. С.167. 3. А.С. 1149978 G11C 19/00 Регистр сдвига. /Н.Я.Какурин, Ю.К. Кирьяков // Открытия, изобретения. 1985. №6. С.250. 4. Макаренко А.Н. Регистры для сжатия двоичных кодовых комбинаций / АСУ и приборы автоматики. 1989. Вып.89. С.32-38. 5. А.С. 1206783 G11C 19/00 Устройство контроля параллельного двоичного кода на нечетность / Н.Я. Какурин, Ю.К. Кирьяков // Открытия, изобретения. 1986. №3.С.250. 6. Еремеев И. С. Устройства сжатия информации. М., 1980. 160с.

*Поступила в редколлегию 18.02.2014*

**Ларченко Лина Викторовна**, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: компьютерные системы автоматического управления, проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

**Вареца Виталий Викторович**, аспирант кафедры автоматизации проектирования вычислительной техники ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

**Ларченко Богдан Дмитриевич**, студент группы КИ-10-5 ХНУРЭ. Научные интересы: проектирование программного обеспечения, языки описания аппаратуры. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

**Макаренко Анна Николаевна**, канд. техн. наук, доцент кафедры информационных технологий Харьковского банковского института. Научные интересы: информационные технологии, анализ и синтез цифровых устройств. Адрес: Украина, 61074, Харьков, пр. Победы, 55, тел. 336-05-64.

## ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ ДЛЯ ЗАДАЧИ КОММИВОЯЖЕРА

---

Описывается методика применения динамического программирования для решения задачи коммивояжера. Излагается алгоритм, а также приводятся теоретические и экспериментальные оценки его временной сложности и требований к памяти ЭВМ.

### 1. Постановка задачи

Метод динамического программирования как способ глобальной оптимизации аддитивных и сводящихся к ним функционалов, автором которого является американский математик Р. Беллман, для решения задачи коммивояжера практически одновременно был предложен Р. Беллманом и М. Хелдом и Р. Карпом [1]. В дальнейшем мы следуем изложению Р. Беллмана.

Как известно, идея динамического программирования, опирающаяся на принцип оптимальности, состоит в том, что процесс оптимизации функционала рассматривается как последовательный  $n$ -шаговый процесс принятия решений, на каждом шаге которого принимается оптимальное решение. Для этого в рассмотрение вводится понятие состояний системы (задачи), относительно каждого из которых вычисляется оптимальное значение функционала, часто называемое значением функции Беллмана.

Смысл задачи коммивояжера состоит в том, что коммивояжер, выезжая из города  $n$ , в определенной последовательности посещает остальные  $1, 2, \dots, n-1$  городов и возвращается в пункт выезда  $n$ . При этом требуется определить такой маршрут, который дает наименьшую стоимость объезда городов при условии, что коммивояжер не заезжает ни в один из них более одного раза, что в терминологии графов означает найти гамильтонов цикл наименьшей длины.

Применительно к этой задаче метод динамического программирования интерпретируется как  $n$ -шаговый процесс принятия решений, на каждом шаге которого коммивояжер должен определить оптимальный маршрут объезда оставшихся городов в зависимости от того, в каком из них он находится.

Предположим, коммивояжеру по пути наименьшей стоимости удалось достичь города  $i$  и до возвращения в город  $n$  осталось посетить  $k$  городов:  $j_1, j_2, \dots, j_k$ . Путь наименьшей стоимости из города  $i$  в город  $n$  через оставшиеся города  $j_1, j_2, \dots, j_k$  обозначим  $V(n; j_1, j_2, \dots, j_k)$ . Иными словами, так определим функцию Беллмана, значение которой устанавливается городом  $i$  и оптимальным порядком объезда оставшихся городов  $j_1, j_2, \dots, j_k$ . Следовательно, если будет найдено значение  $V(n; j_1, j_2, \dots, j_{n-1})$ , т. е. наименьшая стоимость объезда всех городов, задача коммивояжера будет решена. При этом очевидно, что состояниями в  $n$ -шаговом процессе принятия решений будут города  $1, 2, \dots, n-1$ . Например, в задаче с 5-ю городами коммивояжер из города 5 может выехать в любой из оставшихся  $1, 2, \dots, 4$  городов, для которых и должны быть вычислены функции Беллмана.

*Цель исследования* – выяснение возможности использования на современных компьютерах метода динамического программирования для решения задачи коммивояжера.

*Задача для достижения цели* – представление функции Беллмана для решения указанной задачи.

## 2. Алгоритм решения

Действия алгоритма продемонстрируем на примере задачи с 5-ю городами и матрицей

$$\text{стоимости переездов из города в город: } C = \begin{vmatrix} \infty & 25 & 40 & 31 & 27 \\ 5 & \infty & 17 & 30 & 25 \\ 19 & 15 & \infty & 6 & 1 \\ 9 & 50 & 24 & \infty & 6 \\ 22 & 8 & 7 & 10 & \infty \end{vmatrix} [2].$$

Согласно концепции Р. Беллмана минимальная стоимость объезда городов окажется найденной, если на последнем 5-м шаге будет вычислено значение  $V(5; j_1, j_2, j_3, j_4)$ , т. е. определено направление наилучшего выезда из города 5. Так как из этого города коммивояжер может поехать в города 1, 2, 3, 4, а затем из каждого из них по оптимальному маршруту вернуться в город выезда, для 4-го шага принятия решений, рассматривая процедуру в обратном порядке (от 5-го к 1-му шагу), необходимо определить функции Беллмана для городов (состояний) 1, 2, 3, 4.

В связи с тем, что из города 1 коммивояжер в оптимальной последовательности должен проехать города  $j_2, j_3, j_4$ , из города 2 города  $j_1, j_3, j_4$ , из города 3 города  $j_1, j_2, j_4$  и из города 4 города  $j_1, j_2, j_3$ , функции Беллмана согласно  $V(i; j_1, j_2, \dots, j_k)$  для этих состояний будут иметь вид  $V(1; j_2, j_3, j_4)$ ,  $V(2; j_1, j_3, j_4)$ ,  $V(3; j_1, j_2, j_4)$ ,  $V(4; j_1, j_2, j_3)$ . На этом основании, учитывая, что  $V(5; j_1, j_2, j_3, j_4)$  – минимальная стоимость объезда, для вычисления функции Беллмана на 5-м шаге процесса принятия решений получаем выражение

$$V(5; j_1, j_2, j_3, j_4) = \min \{ [c_{51} + V(1; j_2, j_3, j_4)], [c_{52} + V(2; j_1, j_3, j_4)], [c_{53} + V(3; j_1, j_2, j_4)], [c_{54} + V(4; j_1, j_2, j_3)] \}, \quad (1)$$

в котором  $c_{51}, c_{52}, c_{53}, c_{54}$  – стоимость переездов из города 5 в города 1, 2, 3, 4, взятые из матрицы  $C$ .

В свою очередь значения функций Беллмана  $V(1; j_2, j_3, j_4)$ ,  $V(2; j_1, j_3, j_4)$ ,  $V(3; j_1, j_2, j_4)$ ,  $V(4; j_1, j_2, j_3)$ , которые должны вычисляться на 4-м шаге принятия решений, с учетом того, что коммивояжер, выезжая из городов 1, 2, 3, 4, по оптимальному пути должен проехать три остальные города, определяются по выражениям:

$$\begin{aligned} V(1; j_2, j_3, j_4) &= \min \{ [c_{12} + V(2; j_3, j_4)], [c_{13} + V(3; j_2, j_4)], [c_{14} + V(4; j_2, j_3)] \}, \\ V(2; j_1, j_3, j_4) &= \min \{ [c_{21} + V(1; j_3, j_4)], [c_{23} + V(3; j_1, j_4)], [c_{24} + V(4; j_1, j_3)] \}, \\ V(3; j_1, j_2, j_4) &= \min \{ [c_{31} + V(1; j_2, j_4)], [c_{32} + V(2; j_1, j_4)], [c_{34} + V(4; j_1, j_2)] \}, \\ V(4; j_1, j_2, j_3) &= \min \{ [c_{41} + V(1; j_2, j_3)], [c_{42} + V(2; j_1, j_3)], [c_{43} + V(3; j_1, j_2)] \}. \end{aligned} \quad (2)$$

Очевидно, что вычислить эти выражения можно в том случае, если на 3-м шаге принятия решений найти значения функций Беллмана  $V(1; j_2, j_3)$ ,  $V(1; j_2, j_4)$ ,  $V(1; j_3, j_4)$ ;  $V(2; j_1, j_3)$ ,  $V(2; j_1, j_4)$ ,  $V(2; j_3, j_4)$ ;  $V(3; j_1, j_2)$ ,  $V(3; j_1, j_4)$ ,  $V(3; j_2, j_4)$ ;  $V(4; j_1, j_2)$ ,  $V(4; j_1, j_3)$ ,  $V(4; j_2, j_3)$ . Так как приведенные записи этих функций означают, что, например, для города 1 они должны вычисляться на основании вариантов посещения коммивояжером городов в последовательности 2, 3 или 3, 2, городов 2, 4 или 4, 2 и, наконец, городов 3, 4 или 4, 3, выбор предпочтительного маршрута для каждого варианта определяется соответствующим выражением:

$$\begin{aligned} V(1; j_2, j_3) &= \min \{ [c_{12} + V(2; j_3)], [c_{13} + V(3; j_2)] \}, \\ V(1; j_2, j_4) &= \min \{ [c_{12} + V(2; j_4)], [c_{14} + V(4; j_2)] \}, \\ V(1; j_3, j_4) &= \min \{ [c_{13} + V(3; j_4)], [c_{14} + V(4; j_3)] \}. \end{aligned} \quad (3)$$

Рассуждая аналогично, получаем формулы (4), по которым вычисляются значения функций Беллмана для состояний 2, 3, 4:

$$\begin{aligned}
V(2; j_1, j_3) &= \min\{[c_{21} + B(1; j_3)], [c_{23} + B(3; j_1)]\}, \\
V(2; j_1, j_4) &= \min\{[c_{21} + B(1; j_4)], [c_{24} + B(4; j_1)]\}, \\
V(2; j_3, j_4) &= \min\{[c_{23} + B(3; j_4)], [c_{24} + B(4; j_3)]\}, \\
V(3; j_1, j_2) &= \min\{[c_{31} + B(1; j_2)], [c_{32} + B(2; j_1)]\}, \\
V(3; j_1, j_4) &= \min\{[c_{31} + B(1; j_4)], [c_{34} + B(4; j_1)]\}, \\
V(3; j_2, j_4) &= \min\{[c_{32} + B(2; j_4)], [c_{34} + B(4; j_2)]\}, \\
V(4; j_1, j_2) &= \min\{[c_{41} + B(1; j_2)], [c_{42} + B(2; j_1)]\}, \\
V(4; j_1, j_3) &= \min\{[c_{41} + B(1; j_3)], [c_{43} + B(3; j_1)]\}, \\
V(4; j_2, j_3) &= \min\{[c_{42} + B(2; j_3)], [c_{43} + B(3; j_2)]\}.
\end{aligned} \tag{4}$$

Из приведенных выражений следует, что их значения могут быть найдены, если на 2-м шаге принятия решений будут вычислены значения функций  $V(1; j_2)$ ,  $V(1; j_3)$ ,  $V(1; j_4)$ ;  $V(2; j_1)$ ,  $V(2; j_3)$ ,  $V(2; j_4)$ ;  $V(3; j_1)$ ,  $V(3; j_2)$ ,  $V(3; j_4)$ ;  $V(4; j_1)$ ,  $V(4; j_2)$ ,  $V(4; j_3)$ . В свою очередь эти функции могут быть найдены, если учесть, что, например,  $V(1; j_2)$ ,  $V(1; j_3)$ ,  $V(1; j_4)$  определяют оптимальные значения переезда коммивояжера из города 1 в города 2, 3, 4 и дальше из них в город 5, так как рассматривается 2-й шаг обратного процесса принятия решений.

Поэтому

$$\begin{aligned}
V(1; j_2) &= c_{12} + B(2; j_5), & V(1; j_3) &= c_{13} + B(3; j_5), & V(1; j_4) &= c_{14} + B(4; j_5); \\
V(2; j_1) &= c_{21} + B(1; j_5), & V(2; j_3) &= c_{23} + B(3; j_5), & V(2; j_4) &= c_{24} + B(4; j_5); \\
V(3; j_1) &= c_{31} + B(1; j_5), & V(3; j_2) &= c_{32} + B(2; j_5), & V(3; j_4) &= c_{34} + B(4; j_5); \\
V(4; j_1) &= c_{41} + B(1; j_5), & V(4; j_2) &= c_{42} + B(2; j_5), & V(4; j_3) &= c_{43} + B(3; j_5).
\end{aligned} \tag{5}$$

В связи с тем, что функции  $V(1; j_5)$ ,  $V(2; j_5)$ ,  $V(3; j_5)$ ,  $V(4; j_5)$ , фигурирующие в этих выражениях, представляют собой стоимость переездов  $c_{15}$ ,  $c_{25}$ ,  $c_{35}$ ,  $c_{45}$ , содержащихся в матрице  $C$ , процесс попятного принятия решений может быть начат с 1-го шага, сводящегося к перенесению указанных стоимостей в специальную таблицу, строки которой интерпретируются как номера шагов, а колонки определяют значения функций Беллмана.

На основании данных первой строки и выражений (5) получаем значения функций Беллмана для второй строки, по которым, используя выражения (3), (4), вычисляем значения функций третьей строки и т. д. Процесс вычислений и заполнения строк таблицы заканчиваем получением значения функции Беллмана для 5-го шага. Принятые обозначения и найденные значения приведены в табл. 1.

Из нее следует, что наименьшая стоимость объезда 5-и городов равна 62, что совпадает с результатом, полученным в [2] методом ветвей и границ, а также методом прямого перебора.

Для определения маршрута, обеспечивающего найденную стоимость, поступаем следующим образом. В выражении  $V(5; j_1, j_2, j_3, j_4)$  найдем элемент, минимизирующий это выражение. Таким является элемент  $c_{54} + V(4; j_1, j_2, j_3) = 10 + 52 = 62$ . Это означает, что из города 5 коммивояжер выезжает в город 4 (показывают индексы компоненты  $c_{54}$ ). Для того, чтобы определить, в какой город он направляется из города 4, необходимо в выражении для  $V(4; j_1, j_2, j_3)$  найти элемент, доставляющий минимум этому выражению. Так как это элемент  $c_{41} + V(1; j_2, j_3) = 9 + 43 = 52$ , то из города 4 коммивояжер направляется в город 1. Минимум выражению  $V(1; j_2, j_3)$  доставляет элемент  $c_{12} + V(2; j_3) = 25 + 18 = 43$ , из чего следует, что из города 1 коммивояжер выезжает в город 2. В свою очередь минимум  $V(2; j_3) = c_{23} + V(3; j_5) = 17 + 1 = 18$ . Следовательно, из города 2 коммивояжер

направляется в город 3, а из него возвращается в город 5. В результате получаем оптимальный маршрут объезда городов 5, 4, 1, 2, 3, 5, легко представляемый последовательностью дуг (5, 4), (4, 1), (1, 2), (2, 3), (3, 5) для удобной интерпретации на матрице стоимости переездов  $C$ . Дерево поиска решений и оптимальный маршрут на нем, выделенный жирными линиями со стрелками, показаны на рисунке.

Подводя итог изложенному выше, отметим, что алгоритм решения задачи коммивояжера методом динамического программирования, как и в большинстве случаев применения этого метода, состоит из двух этапов: вычисления оптимального значения функционала и на его основе определения последовательности оптимальных решений.

### 3. Оценки временной сложности алгоритма и требований к памяти ЭВМ

Теоретическую временную сложность алгоритма, т. е. зависимость времени решения задачи от ее размера  $n$ , определим традиционным путем: подсчитаем количество основных вычислительных операций  $N_n$ , которые должен выполнить алгоритм при решении задачи. Согласно совокупности его действий такими операциями являются операции сложения и сравнения. Подсчет суммарного количества этих операций выполним для каждого шага алгоритма отдельно.

На 2-м его шаге для получения значений функций Беллмана необходимо выполнить лишь операции сложения. Поэтому, поскольку число состояний задачи равно  $n - 1$ , всего согласно выражениям (5) для 2-го шага требуется выполнить  $(n - 1) \cdot (n - 2)$  сложения.

На 3-м шаге алгоритма для вычисления значений функций Беллмана по каждому из  $n - 1$  состояний согласно выражениям (3), (4) требуется вычислить последовательность минимальных значений для пар городов. При этом для каждой пары необходимо выполнить два сложения и одно сравнение. Общее же число операций, которое требуется выполнить для каждого состояния, определяется количеством пар городов, которое, как очевидно, зависит от размера задачи. Для определения этого числа, как функции  $n$ , обратимся к примерам.

В задаче из пяти городов 1, 2, 3, 4, 5, предполагая, что коммивояжер находится в городе 1, он может посетить города (2, 3) или (3, 2), города (2, 4) или (4, 2), города (3, 4) или (4, 3). Значения функций Беллмана для этого случая вычисляются по выражениям (3). Таким образом, размерность  $n = 5$  определяет три пары (2, 3), (2, 4), (3, 4), т. е. общее количество пар  $S = 1 + 2 = 3$ .

Рассмотрим теперь задачу с шестью городами 1, 2, 3, 4, 5, 6. В этом случае пары городов, которые может посетить коммивояжер, находясь в городе 1, такие: (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5). В результате  $S = 1 + 2 + 3 = 6$ . По аналогии для задачи с семью городами получим такую последовательность пар: (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 6), т. е.  $S = 1 + 2 + 3 + 4 = 10$ .

Как известно, общая формула для вычисления суммы ряда  $1 + 2 + \dots + n$  такая:  $S = \frac{n \cdot (n + 1)}{2}$ . Учитывая, что в задаче  $n$  интерпретируется как число состояний, равное

$n - 1$ , и отсчет пар начинается со второго элемента последовательностей городов, для числа пар по общей формуле получаем  $S = \frac{[(n - 1) - 2] \cdot [(n - 1) - 1]}{2} = \frac{(n - 3) \cdot (n - 2)}{2}$ . Учиты-

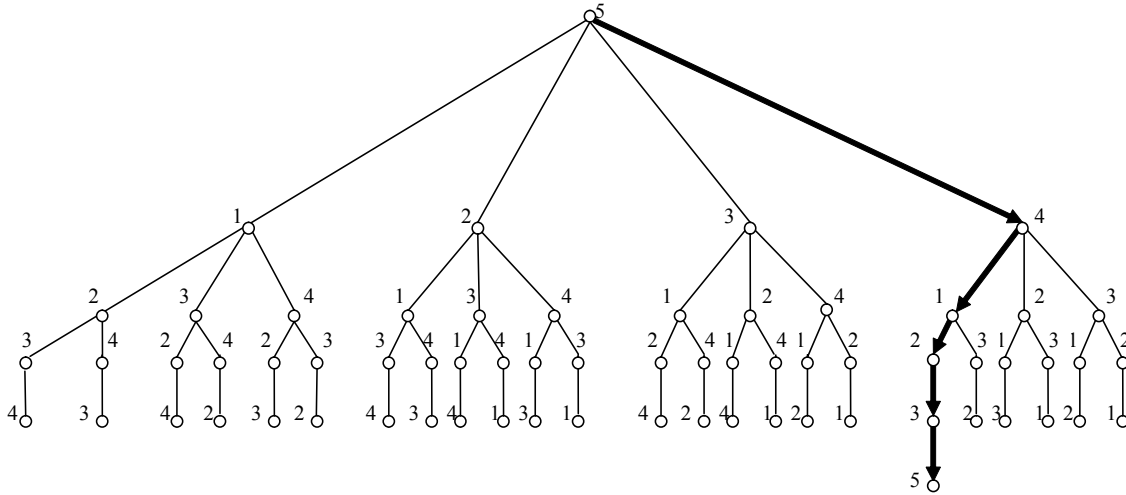
вая, что каждая пара требует выполнения трех основных операций и количество городов, для которых определяется  $S$ , равно  $n - 1$ , для числа основных операций третьего шага

получаем  $N_3 = 3 \cdot S \cdot (n - 1) = \frac{3 \cdot (n - 3) \cdot (n - 2) \cdot (n - 1)}{2}$ .

Как следует из примера, на четвертом шаге алгоритма по выражениям (2) требуется найти минимальные значения для последовательности троек городов, которые может посетить коммивояжер, находясь в каком-либо другом городе. Число таких троек, как и число пар, зависит от размера задачи.

Таблица 1. Обозначения и значения функции Беллмана

Обозначения и значения функции Беллмана													
$B(1;j_5)$	$B(2;j_5)$	$B(3;j_5)$	$B(4;j_5)$										
		1	6										
$B(1;j_2)$	$B(1;j_3)$	$B(1;j_4)$	$B(2;j_1)$	$B(2;j_3)$	$B(2;j_4)$	$B(3;j_1)$	$B(3;j_2)$	$B(3;j_4)$	$B(4;j_1)$	$B(4;j_2)$	$B(4;j_3)$		
				37	32	18	36	46	40	12	36	75	25
$B(1;j_2, j_3)$	$B(1;j_2, j_4)$	$B(1;j_3, j_4)$	$B(2;j_1, j_3)$	$B(2;j_1, j_4)$	$B(2;j_3, j_4)$	$B(3;j_1, j_2)$	$B(3;j_1, j_4)$	$B(3;j_2, j_4)$	$B(4;j_1, j_2)$	$B(4;j_1, j_3)$	$B(4;j_2, j_3)$		
				52	46	42	29	47	42	51	59	50	64
$B(1;j_2, j_3, j_4)$	$B(2;j_1, j_3, j_4)$			$B(3;j_1, j_2, j_4)$			$B(4;j_1, j_2, j_3)$						
				57			52						
$B(5;j_1, j_2, j_3, j_4)$	62												



Легко убедиться в том, что число пар городов определяется числом сочетаний  $C_{n-2}^2$ . Тогда по аналогии число их троек равно  $C_{n-2}^3$ , число четверок  $C_{n-2}^4$  и т. д. Учитывая, что на последнем шаге алгоритма ищется минимальное значение стоимости маршрута для последовательности из  $n-1$  городов, а на предпоследнем шаге – для последовательности из  $n-2$  городов, получаем, что для этого шага число сочетаний равно  $C_{n-2}^{n-2} = 1$ . С другой стороны, так как на 3-м шаге алгоритма при минимизации необходимо выполнить два сложения и одно сравнение, на четвертом – три сложения и два сравнения, получаем, что для  $C_{n-2}^m$  потребуется выполнить  $2 \cdot m - 1$  основных операций. На этом основании для  $C_{n-2}^{n-2}$  число таких операций равно  $2 \cdot (n-2) - 1 = 2 \cdot n - 5$ . Принимая во внимание, что на последнем шаге алгоритма необходимо найти минимум из  $n-1$  элементов, в каждом из которых имеется операция сложения, получаем, что для этого шага необходимо выполнить  $(n-2) + (n-1) = 2 \cdot n - 3$  основных операции.

Таким образом, на основании приведенных выкладок получаем, что общее число основных операций, которые должен выполнить алгоритм, решая задачу для  $n$  городов, может быть найдено по выражению:

$$\begin{aligned}
 H_n = & (n-1) \cdot (n-2) + (n-1) \cdot [(2 \cdot m - 1) \cdot C_{n-2}^2 + (2 \cdot m - 1) \cdot C_{n-2}^3 + \dots + \\
 & + (2 \cdot m - 1) \cdot C_{n-2}^{n-3} + 2 \cdot n - 5] + 2 \cdot n - 3 = (n-1) \cdot [(2 \cdot m - 1) \cdot C_{n-2}^2 + (2 \cdot m - 1) \cdot C_{n-2}^3 + \dots + \\
 & + (2 \cdot m - 1) \cdot C_{n-2}^{n-3} + 3 \cdot n - 7] + 2 \cdot n - 3,
 \end{aligned} \quad (6)$$

в котором  $m = 2, 3, \dots, n-3$ . К этому числу основных операций необходимо прибавить то их количество, которое потребуется выполнить при определении оптимального маршрута объезда городов на основе найденного значения функционала. Оно равно сумме, получаемой сложением числа операций каждого шага, заканчивая 2-м шагом и исключая  $n$ -й шаг, так как на этом шаге индекс элемента, доставляющий минимальное значение функционалу, легко фиксируется. Поэтому выражение для указанного числа операций имеет вид  $[(n-2) + (n-3)] + [(n-3) + (n-4)] + \dots + 3 + 1 = (2 \cdot n - 5) + (2 \cdot n - 7) + \dots + 4$ .

Таким образом, при двухэтапном процессе реализации алгоритма потребуется выполнить  $H_n = (n-1) \cdot [(2 \cdot m - 1) \cdot C_{n-2}^2 + (2 \cdot m - 1) \cdot C_{n-2}^3 + \dots + (2 \cdot m - 1) \cdot C_{n-2}^{n-3} + 3 \cdot n - 7] + 2 \cdot n - 3 + (2 \cdot n - 5) + (2 \cdot n - 7) + \dots + 4$  основных операций.

Полученное выражение позволяет экспериментальным путем определить вычислительную сложность алгоритма. С этой целью для ряда размеров задачи  $n = 5, 10, 15, 20, 25$  были вычислены значения  $H_n$ , представленные в табл. 2.

Таблица 2. Число основных вычислительных операций

n = 5	n = 10	n = 15	n = 20	n = 25
107	16190	1376803	106717205	4429186681

Данные табл. 2 приближены эмпирической формулой  $H_n(n) = 4,0282e^{0,3912n}$ , которая показывает, что при увеличении размера задачи число основных вычислительных операций, которые выполняет алгоритм, растет экспоненциально. При этом показатель экспоненты равен  $0,3912 \cdot n$ , а постоянный множитель –  $4,0282$ . Поэтому согласно этой формуле, например, при  $n = 60$  число операций окажется примерно таким  $H_n(60) = 4,0282 \cdot e^{0,3912 \cdot 60} = 4,0282 \cdot e^{23,472} = 62931937743$ , с которым компьютер с тактовой частотой центрального процессора 4 ГГц справится не более, чем за 20 с.

Память ЭВМ при увеличении размера задачи растет как количество столбцов табл. 1, умноженное на количество ее строк  $n - 1$ , предназначенных для хранения значений функций Беллмана при условии, что для этих целей в простейшем случае используется матрица. В свою очередь количество столбцов определяется числом состояний  $n - 1$ , умноженным на

число пар операций  $S = \frac{(n-2) \cdot (n-3)}{2}$ , которые необходимо выполнить на 3-м шаге алгоритма. Таким образом, требуемая память определяется формулой

$P(n) = (n-1) \cdot (n-1) \cdot S = \frac{(n-1)^2 \cdot (n-2) \cdot (n-3)}{2}$ , из которой следует, что при увеличении размера задачи  $n$  она растет как полином четвертой степени  $n$ .

Для  $n = 60$  значение этого полинома равно 5754093. Умножив его на 4, получим число байт памяти ЭВМ, равное 23016372, т. е. около 22 МБ, которые необходимы для хранения значений функции Беллмана при решении задачи указанного размера. Очевидно, что для компьютеров, располагающих оперативной памятью, например, 4 ГБ, указанное число достаточно просто помещается в этой памяти.

## Выводы

Таким образом, на основании анализа и расчетов получены новые результаты, показывающие объективную возможность решения задачи коммивояжера с числом городов 60 – 65 методом динамического программирования на современных персональных компьютерах за практически приемлемое время (не более 30 с).

*Научная новизна* работы – сконструирован алгоритм, получены оценки его временной сложности, которые показывают возможность его применения на современных компьютерах. *Практическая значимость* – алгоритм легко программируем, после чего он может быть применен на практике.

**Список литературы:** 1. Беллман Р. Применение динамического программирования к задаче о коммивояжере // Кибернетический сборник. М.: Мир, 1964. Т. 9. С. 219 – 228. 2. Гудман С., Хиднетиеми С. Введение в разработку и анализ алгоритмов. М.: Мир, 1981. 368 с.

Поступила в редколлегию 04.03.2014

**Канцедал Сергей Андреевич**, д-р техн. наук, профессор кафедры прикладной математики Западодонбасского института Межрегиональной академии управления персоналом. Научные интересы: математическое моделирование, теория расписаний и ее применение. Адрес: Украина, 51400, Днепропетровская обл., Павлоград, ул. Днепровская, 400/1.

**Костикова Марина Владимировна**, канд. техн. наук, доцент кафедры информационных технологий и мехатроники Харьковского национального автомобильно-дорожного университета. Научные интересы: математическое моделирование, теория расписаний и ее применение. Адрес: Украина, 61002, Харьков, ул. Петровского, 25, тел. 707-37-74.

## ПРОЕКТУВАННЯ ЛОГІЧНО-АРИФМЕТИЧНИХ НАНОПРИСТРОЇВ

Реалізується комп'ютерне проектування наноелектронних схем на квантових мажоритарних компонентах. Пропонуються методи побудови логічно-арифметичних обчислювальних пристроїв комбінаційного типу, які реалізують практично повну систему логічних функцій як в мажоритарному, так і в булевому базисах.

**Теза.** Квантові коміркові автомати – технологія, що виникла два десятиліття тому, в якій значенням логічних станів відповідають позиції окремих електронів. Квантові комірки використовуються для конструювання логічних нанoeлементів та арифметичних нанопристроїв. Робота присвячена комп'ютерному проектуванню сучасних логічно-арифметичних пристроїв, до складу яких входять універсальні мажоритарні елементи. В роботі виконується моделювання квантових нанопристроїв з використанням системи автоматизованого проектування (САПР) QCADesiner.

**Ключові слова:** квантовий комірковий автомат, мажоритарний елемент, автоматизоване проектування, логічно-арифметичні пристрої.

**Вступ.** Квантові коміркові автомати – це обчислювальна парадигма, згідно з якою інформація представляється певною конфігурацією електронів у комірці квантових автоматів, що складається з однієї чи двох окремих молекул [1].

Орієнтація пари квантових комірок така, що їхнє взаєморозміщення визначає взаємний вплив. Ця взаємодія зарядів між сусідніми комірками дає змогу обробляти і передавати інформацію. Такий метод є функціонально аналогічним, але структурно відрізняється від способу, в який комбінуються окремі вентиля у великих інтегральних мікросхемах для забезпечення виконання логічно-арифметичних операцій та створення схем з пам'яттю.

**1. Основи теорії квантових коміркових автоматів.** Прилади на квантових коміркових автоматах (КА) складаються з нанорозмірних діелектричних комірок, які мають чотири квантові напівпровідникові точки, розташовані у кутках, і два рухомі електрони. Окрема комірка забезпечує тунельні переходи з потенціальними бар'єрами. Переходи контролюються локальним електричним полем, величина якого збільшується для перешкоджання руху електронів, або зменшується для його спонукання. Таким чином, окрема комірка може перебувати в одному з трьох станів. Нульовий стан, або стан невизначеності настає, коли потенціальний бар'єр зменшений, і рухомий електрон може зайняти будь-яку з вакансій. Інші два стани – стани поляризації, виникають, коли величина потенціального бар'єру зростає і зберігає свій рівень, щоб мінімізувати енергетичний рівень комірки. Набір станів  $Q$  скінченний і типовий:  $Q = \{0,1\}$ . Ймовірність перебування комірки в одному із станів поляризації може бути співвіднесена з густиною заряду кожної окремої квантової точки і визначається формулою:

$$P = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{(\rho_1 + \rho_3) + (\rho_2 + \rho_4)} = \pm 1,$$

де  $\rho_i$  – густина електричного заряду кожної квантової точки комірки.

На рис. 1 зображена базова комірка КА, два способи її розміщення в просторі і поляризації електронів.

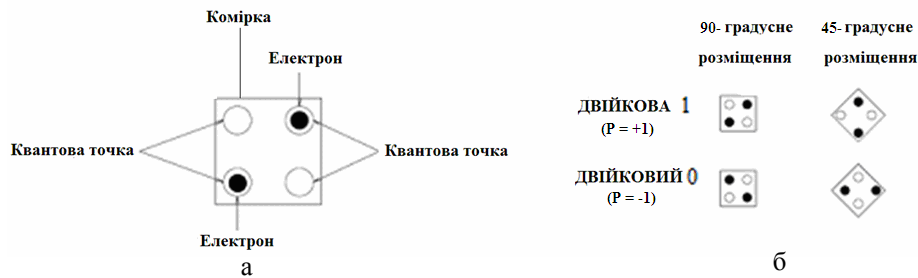


Рис. 1. Комірка квантового автомата (а), її два способи розміщення в просторі (б) і поляризації ( $P = \pm 1$ )

**2. Логічний мажоритарний елемент та інвертор.** Розміщуючи комірки в послідовності одна за одною і викликаючи їх взаємодію між собою, можливо забезпечити протікання інформації по такому провіднику. Теоретично існує два методи побудови провідника залежно від 45-градусної чи 90-градусної орієнтації комірок, але технологічно складно виготовити нанокмірки з різною орієнтацією.

За допомогою КА можуть бути сконструйовані різні елементи для виконання логічних і арифметичних операцій. Базовими логічними нанокмірками в теорії коміркових автоматів є мажоритарний елемент (МЕ) та інвертор (рис. 2).

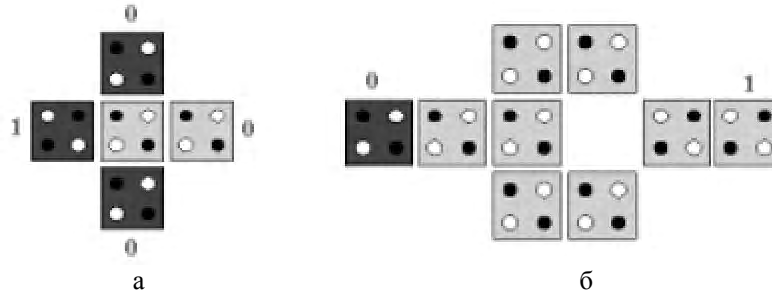


Рис. 2. Тривходовий мажоритарний елемент (а) та інвертор (б) на базі коміркових автоматів

Поляризація вихідної комірки МЕ збігається з поляризаціями більшості вхідних комірок. Булевий вираз для мажоритарної функції:

$$\text{maj}(x_2, x_1, x_0) = x_2 x_1 \vee x_2 x_0 \vee x_1 x_0,$$

де  $x_2, x_1$  та  $x_0$  – вхідні аргументи. Фіксація поляризації одного з входів мажоритарного елемента як логічний 0 чи логічна 1 дає змогу отримати елементи І чи АБО, відповідно  $\text{maj}(x_2, x_1, 0) = x_2 \cdot x_1$ ,  $\text{maj}(x_2, x_1, 1) = x_2 \vee x_1$ . Такі комірки можуть бути створені в процесі виробничого виготовлення, що усуває необхідність підтримувати постійний струм через схему.

**3. Синхронізація.** Синхронізація відіграє важливу роль в керуванні роботою квантово-коміркової логіки. Такий контроль реалізується в результаті приєднання КА до тактових зон таким чином, що вони замикаються в послідовність для бажаного напрямку передачі сигналу. Коли потенціал зони, де знаходиться КА, зменшується, електрони втрачають стабільність, в результаті чого поляризація комірки не визначається. Зростання потенціального бар'єру зменшує коефіцієнт тунелювання, в результаті чого електрон починає локалізуватися. Якщо електрон займає одне місце, то КА набуває визначеної поляризації. Коли потенціальний бар'єр досягає максимуму, говорять, що комірка закрита. Закриті комірки працюють як умовний вхід, причому дійсний вхід може набувати нового значення. Саме тому існує певна затримка в поширенні сигналу через КА. З метою забезпечення обчислень сигнал проходить через тактові зони, що представляють собою області, де відбуваються обчислення. Тактові зони фізично упорядковані, а це означає, що обчислення повинні проводитися в послідовному порядку. Сигнали повинні надходити до схеми обрахунку без затримок.

Квантові комірки МЕ повинні бути в тактовій зоні, відділеній від тактових зон вхідних і вихідної комірок, тому мажоритарний елемент розташовується на краях інших тактових зон.

**4. Суматори.** Науковці Харківського університету радіоелектроніки [2] та Університету Нотр Дам [1] першими запропонували мікро- та наносхеми суматорів на базі МЕ. Однобітний напівсуматор на КА може бути складений з чотирьох МЕ і двох інверторів (рис.3). Вирази для суми  $S$  і переповнення  $C$  цього суматора такі:

$$S = \text{maj}(\overline{xy}, \overline{xy}, 1) = x \oplus y, \quad C = \text{maj}(x, y, 0) = xy,$$

де  $x, y$  – вхідні доданки.

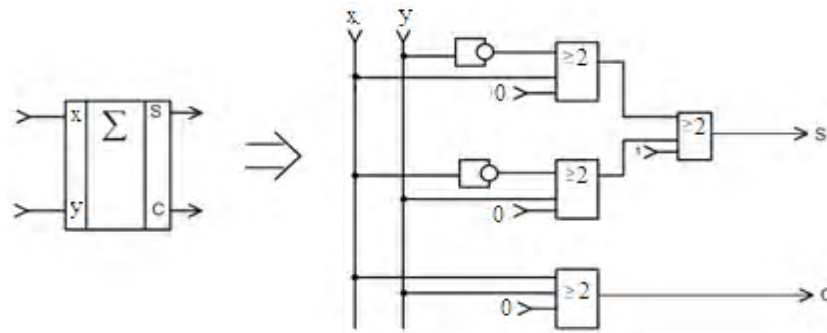


Рис. 3. Схема однобітного напівсуматора на тривходових мажоритарних елементах

На рис. 4 наведена наносхема на КА однобітного напівсуматора (а) та результати моделювання її часових діаграм на планшетному полі САПР QCADesiner [4, 5]. Для конструювання вхідних провідників використані комірки з 45 градусною орієнтацією. Використання такого способу розміщення комірок створює інвертуючий ланцюг, в якому кожна комірка протилежно поляризована до сусідніх.

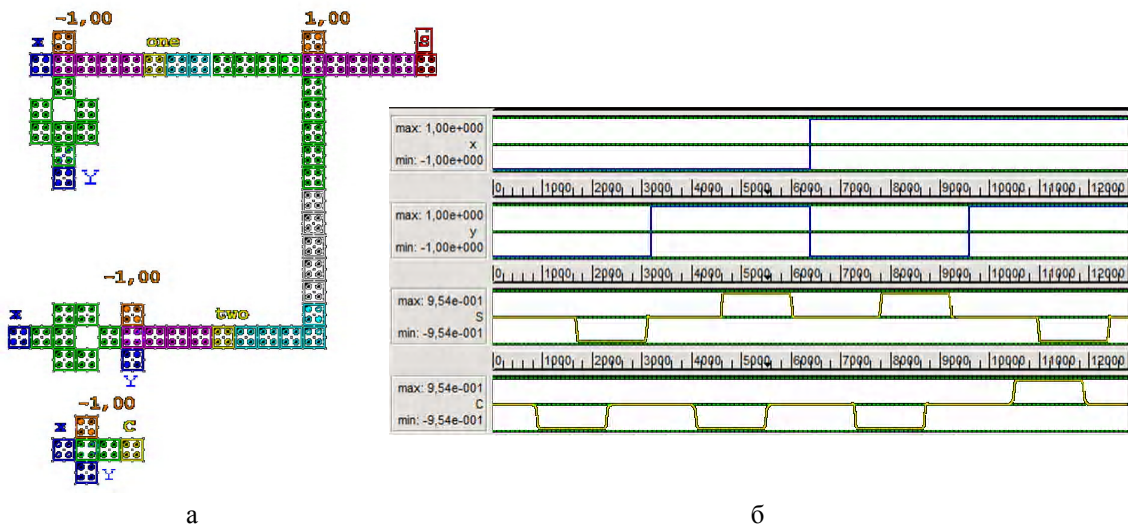


Рис. 4. Наносхема однобітного напівсуматора на КА (а) та результат моделювання його логічних діаграм (б)

Якщо провідник з однаково поляризованих комірок перетинає інвертуючий провідник, провідники не взаємодіють між собою, тому можуть працювати незалежно на одному рівні. Такий перетин має назву компланарний [3,5]. Проблемою компланарного перетину є те, що відстань між комірками окремого провідника призводить до зниження ймовірності проходження сигналу. Затримка такого суматора – один тактовий цикл, що складається з чотирьох тактових зон, представлених на рис. 4 різними ступенями градацій сірого.

Дворозрядний суматор реалізує арифметичне додавання  $S_2, S_1, S_0$  дворозрядних бінарних чисел  $x_1x_0$  та  $y_1y_0$  (рис.5).

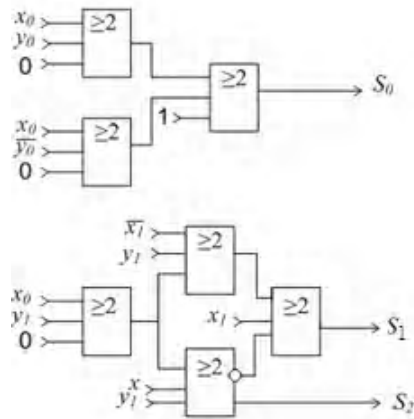


Рис.5. Наносхема дворозрядного суматора на тривходових мажоритарних елементах  
Логічні рівняння для трьох виходів дворозрядного суматора, реалізованого на семи МЕ і одному інверторі, мають вигляд:

$$S_0 = \text{maj}(\text{maj}(x_0, \bar{y}_0, 0), \text{maj}(x_0, y_0, 0), 1) = x_0 \bar{y}_0 \vee \bar{x}_0 y_0 = x_0 \oplus y_0,$$

$$C_0 = \text{maj}(x_0, y_0, 0) = x_0 y_0,$$

$$S_1 = \text{maj}(\text{maj}(x_1, y_1, C_0), x_1, \text{maj}(x_1, y_1, \bar{C}_0)) = \bar{x}_1 \bar{y}_1 C_0 \vee \bar{x}_1 y_1 \bar{C}_0 \vee x_1 \bar{y}_1 \bar{C}_0 \vee x_1 y_1 C_0 =$$

$$C_0 (\bar{x}_1 \bar{y}_1 \vee x_1 y_1) \vee \bar{C}_0 (\bar{x}_1 y_1 \vee x_1 \bar{y}_1) = (x_1 \oplus y_1) \oplus C_0,$$

$$S_2 = \text{maj}(C_0, x_1, y_1) = x_1 y_1 \vee x_1 C_0 \vee y_1 C_0.$$

Далі на рис. 6, 7 показано результати проектування цієї наносхеми за допомогою САПР QCAD.

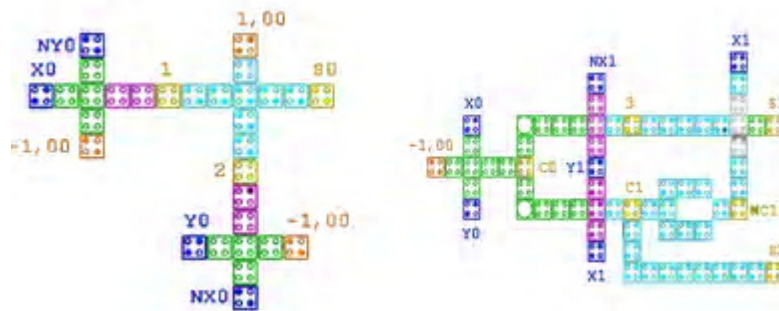


Рис. 6. Проект наносхеми дворозрядного суматора

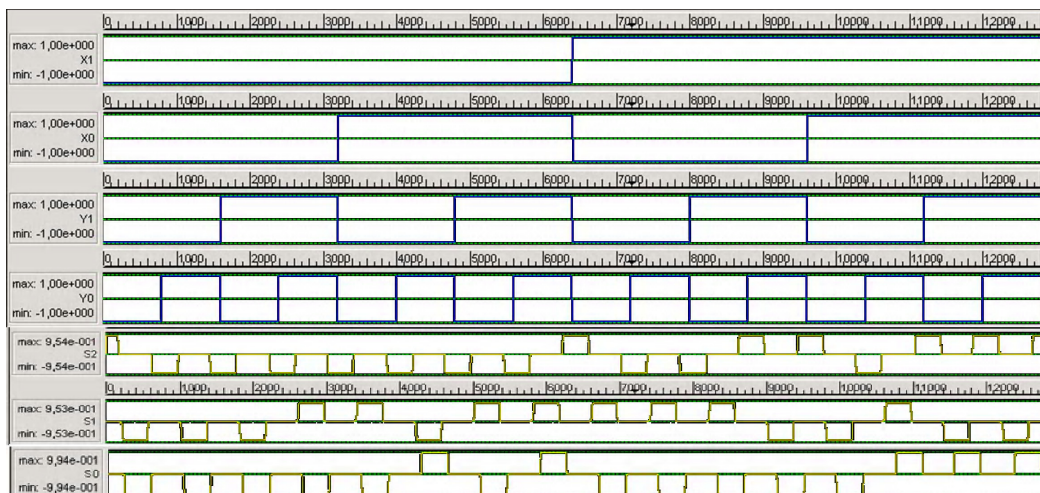


Рис. 7. Моделювання часових характеристик дворозрядного суматора

Спроектована таким чином наносхема дворозрядного суматора базується на 98 квантових коміркових автоматах, розмір яких (18x18) нм, з чотирма квантовими точками діаметром 5 нм, відстань між центрами сусідніх комірок складає 20 нм. Загальний розмір конструкції (360x414) нм. Існує 11 входів і три виходи, а чотири комірки мають фіксовану поляризацію.

При використанні п'ятивходових МЕ можна побудувати найбільш раціональну і просту схему однорозрядного суматора. Структурна схема однорозрядного суматора побудована на одному п'ятивходовому МЕ і одному тривходовому (рис. 8).

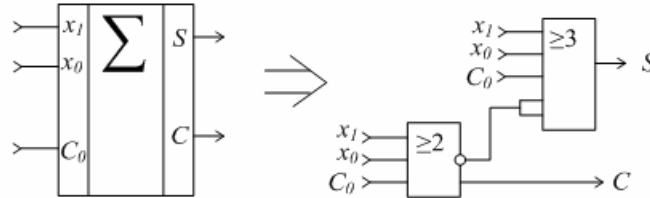


Рис. 8. Схема однорозрядного суматора на базі п'ятивходового МЕ

Результати комп'ютерного проектування цієї наносхеми за допомогою САПР QCAD наведені на рис. 9 та 10.

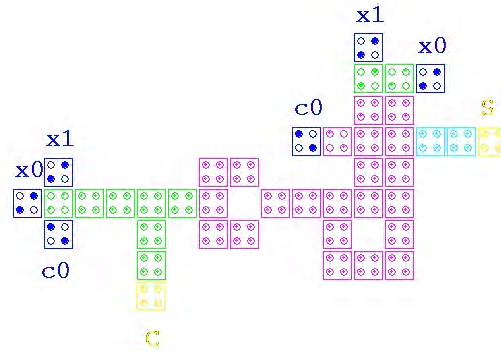


Рис. 9. Проект повного однорозрядного суматора на п'ятивходовому МЕ

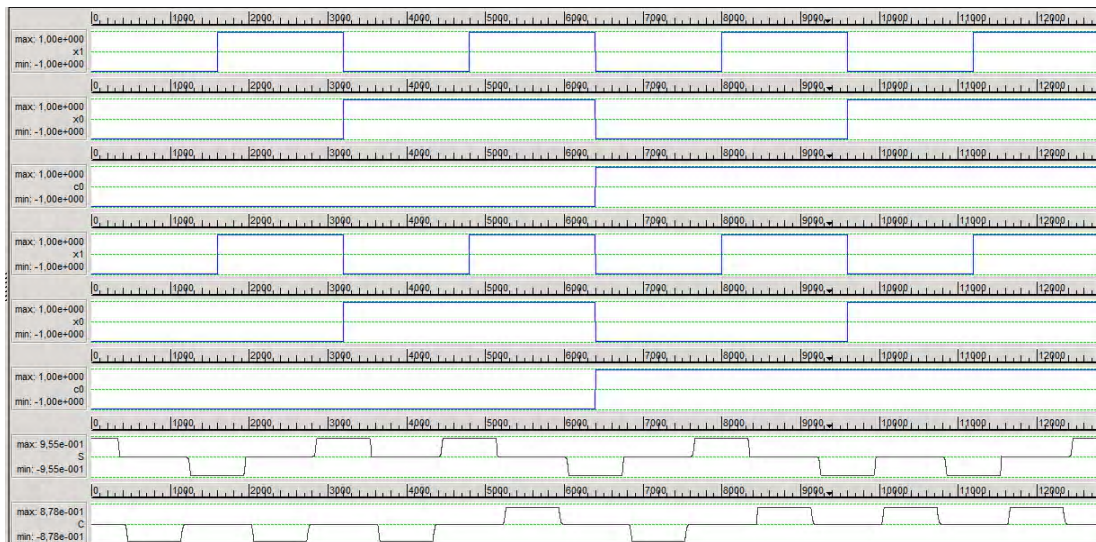


Рис. 10. Моделювання часових характеристик повного однорозрядного суматора на п'ятивходовому МЕ

Функції додавання S та переносу C визначаються правилами додавання в мажоритарно-му базисі [2]:

$$S = \text{maj}(x_1, x_0, C_0, \bar{C}) = \bar{C}(x_1 \vee x_0 \vee C_0) \vee x_1 x_0 C_0,$$

$$C = \text{maj}(x_1, x_0, C_0) = x_1 x_0 \vee x_1 C_0 \vee x_0 C_0.$$

Найпрорстіша наносхема повного суматора базується на 41 КА, а її загальний розмір становить (288x162) нм.

**Висновки.** Виконане комп'ютерне проектування одно- та дворозрядних суматорів. Використання багаторівневого перетинання провідників дозволяє уникнути проблем компланарного перетину. Моделювання виконане в такий спосіб, що виходи перебувають в останній четвертій тактовій зоні, а це відповідає фазі спокою після фази, в якій проводилися останні обчислення. Необхідно три повних тактових цикли для завершення операції додавання. Досягнута мета проектування надійного розшарування наносхеми дворозрядного суматора і підвищення його експлуатаційної надійності, але наявність недоліків і дефектів, пов'язаних з молекулярною технологією виготовлення квантових коміркових автоматів, потребує подальшої роботи в цьому напрямку.

**Список літератури:** 1. *Tougaw, P.D., Lent C.S.* Logic devices implemented using quantum cellular automata / *Tougaw, P.D., Lent C.S.* // *J. Appl. Phys., American Institute of Physics.* 1994. 2. *Пакулов Н.Н.* Мажоритарний принцип построения надежных узлов и устройств ЦВМ / *Пакулов Н.Н.* М.: Сов. радио, 1974. 3. *Bhanja, S., Ottavi, M., Lombardi, F., Pontarelli. S.* OCA circuit for robust coplanar crossing / *Bhanja, S., Ottavi, M., Lombardi, F., Pontarelli. S.* // *Journal of Electronic Testing.* 2007. P. 193-210. 4. *Melnyk O.S., Tsapok L.O.* Computer simulation of nanoelectronics arithmetic – logic devices // *Електроніка та системи управління.* - 2012,- N1 (31), P.5-10. 5. *Walus, K.* QCADesiner: A CAD Tool for an Emerging Nano-Technology / *Walus, K.* // *Micronet Annual Workshop.* 2003.

*Надійшла до редколегії 13.03.2014*

**Мельник Олександр Степанович**, канд. техн. наук, доцент кафедри електроніки Харківського національного авіаційного університету. Наукові інтереси: наноелектроніка, системи автоматизованого проектування, моделювання одноелектронних схем. E-mail: melnyk.ols@gmail.com, тел.: (050) 5501090, (093) 9945251.

**Козаревич Вікторія Олександрівна**, асистент кафедри електроніки Харківського національного авіаційного університету. Наукові інтереси: цифрова наноелектроніка, математичне моделювання одноелектронних схем. E-mail: st-viktoria@yandex.ru

**Романюк Всеволод Юрійович**, студент кафедри електроніки Харківського національного авіаційного університету. Наукові інтереси: моделювання арифметико-логічних наносхем. E-mail: sevczik@rambler.ru.

## ВИКОРИСТАННЯ АДАПТИВНИХ ОНТОЛОГІЙ В ЗАДАЧІ КВАЗІРЕФЕРУВАННЯ ТЕКСТОВИХ ДОКУМЕНТІВ

Розглядається підхід до розроблення інтелектуальних систем з використанням онтологій у складі їх баз знань. Здійснюється класифікація таких систем з погляду їх функціонування на основі онтологій. Вводиться поняття адаптивної онтології. Модель адаптивної онтології визначається як розвиток класичної моделі додаванням ваг важливості понять та відношень, які зберігаються в онтології. Показується, як такі онтології можна використати в задачі квазіреферування текстових документів.

### 1. Вступ

Сучасні дослідження побудови інтелектуальних систем (ІнС) ведуться у двох напрямках: 1) ІнС класифікації (виведення за прецедентами, англ. Case-Based Reasoning); 2) ІнС планування діяльності (пошук стану мети у просторі станів) [1-4].

Вибір ІнС залежить від типу задачі. Метод виведення за прецедентами ефективний, коли основним джерелом знань про задачу є досвід, а не теорія; рішення не є унікальними для конкретної ситуації, а можуть використовуватись в інших випадках; мета розв'язування задачі - отримати не гарантований правильний розв'язок, а найкращий серед можливих. Виведення, основане на прецедентах, є методом побудови ІнС, які приймають рішення щодо проблеми або ситуації за результатами пошуку аналогій, що зберігаються в базі класів. З математичного погляду поточна ситуація  $S$  належить до класу  $Class_k$  серед множини  $N$  класів  $Class = \{Class_1, Class_2, \dots, Class_N\}$ , якщо відстань від  $S$  до цього класу є найменшою, тобто

$$Class_k = \arg \min_i d(Class_i, S), \quad i = \overline{1, N}. \quad (1)$$

ІнС планування діяльності має досягти стану мети. Насамперед потрібно розробити план досягнення цього стану всіма можливими альтернативними способами. Процес планування ґрунтується на принципі декомпозиції. Задача планування  $ZP$  містить три складові: множину станів  $St$ , множину дій  $A$ , множину станів мети  $Goal$ , тобто

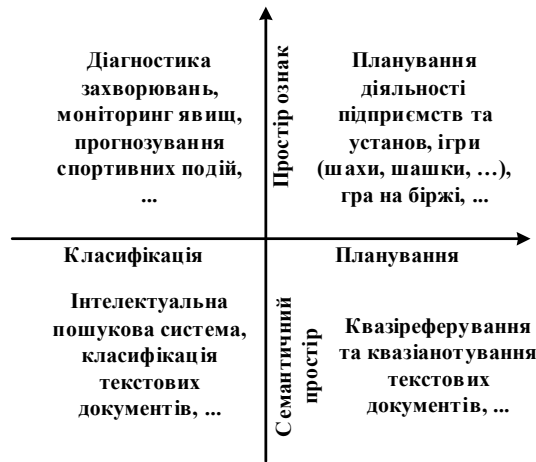
$$ZP = \langle St, A, Goal \rangle. \quad (2)$$

Для ефективного планування діяльності ІнС повинна вміти оцінювати стани та дії. Як бачимо, для обох типів ІнС необхідна метрика. У першому випадку – для оцінювання близькості класу, у другому – для визначення релевантності станів та дій. Від способу побудови цієї метрики безпосередньо залежить ефективність функціонування ІнС [1].

Проаналізувавши типи задач, для яких використовують онтології, робимо висновок, що всі задачі можна поділити на два підтипи. Перший – тип задач, для яких суттєво, які значення приймають властивості понять. Сюди належать задачі діагностики захворювань, розпізнавання образів, класифікації явищ на підставі зібраних даних тощо. Такі задачі назвемо ознаковими. Для іншого типу задач не є істотним значення понять, швидше – їх семантика або частотність вживання понять у тексті і т.д. До таких задач можна зарахувати кластеризацію інформаційних ресурсів, класифікацію текстів згідно з УДК, інтелектуальні пошукові системи, квазіреферування та квазіанотування текстових документів. Задачі такого типу назвемо семантичними. У результаті отримаємо поділ ІнС за двома вимірами (напрямом розвитку та простором функціонування), як зображено на рис 1. У кожній чверті перераховано задачі, які потрапляють у відповідний тип.

Для ефективного функціонування ІнС необхідно побудувати метрику, на основі якої можна визначити релевантність станів чи класів. Побудова такої метрики прямо залежить від типу задач: семантичні вони чи ознакові. Отже, загалом виділяють чотири різні типи задач, які розв'язують за допомогою ІнС. Зріз за напрямом досліджень потребує двох різних функціональних моделей (пошук класу та планування діяльності), зріз за типом задачі

– використання різного роду метрик для їх розв’язування та оцінювання якості отриманих розв’язків. Розглянемо всі ці типи задач, насамперед ввівши поняття адаптивної онтології (АО) (рисунок).



Типи задач, для розв’язування яких використовують ІСППР

Ефективність адаптації онтології БЗ до особливостей ПО визначають елементи її структури та механізми її адаптації через самонавчання під час експлуатації. Одним з підходів до реалізації таких механізмів є автоматизоване зважування понять БЗ та семантичних зв’язків між ними під час самонавчання. Цю роль виконують ваги важливості понять та зв’язків. Вага важливості поняття (зв’язку) – це числова міра, котра характеризує значущість певного поняття (зв’язку) у конкретній ПО і динамічно змінюється за певними правилами під час експлуатації системи. Запропоновано розширити модель онтології (1), ввівши в її формальний опис ваги важливості понять та відношень [1, 5-7]. Таку онтологію визначено як:

$$\hat{O} = \langle \hat{C}, \hat{R}, F \rangle, \quad (3)$$

де  $\hat{C} = \langle C, W \rangle$ ,  $\hat{R} = \langle R, L \rangle$ ;  $W$  – вага важливості понять  $C$ ;  $L$  – вага важливості відношень  $R$ .

Визначену у такий спосіб онтологію названо адаптивною, тобто такою, що адаптується до ПО за допомогою задання ваг важливості понять та зв’язків між ними. Така онтологія однозначно подається у вигляді зваженого концептуального графа. Тому метрику побудовано на таких графах.

Переваги моделі (3) полягають у можливості: 1) будувати метрики на основі онтології; 2) адаптувати базу знань ІнС до потреб користувача; 3) задавати важливість знань з точки зору експерта ПО. АО на відміну від звичайної онтології відображає не лише експліцитні (явні) знання, а й імпліцитні (неявні, приховані). Методи інтелектуального аналізу даних (дерева рішень, байєсівські мережі,  $k$ -найближчих сусідів) є окремим випадком АО залежно від правил задання ваг важливості понять та відношень. З точки зору побудови БЗ ІнС отримуємо такий підхід – експерту або користувачу системи надається готова БЗ, ядром якої є онтологія, а їх задача зводиться лише у налаштуванні цієї БЗ під себе шляхом задання ваг важливості її елементів.

Детальніше розглянемо використання адаптивних онтологій для задач, які знаходяться у 4-й чверті (планування-семантичний простір).

Отже, метою даного дослідження є розроблення ефективного методу автоматизованого квазіреферування природомовного документа. Для досягнення мети пропонується використати адаптивну онтологію предметної області, до якої належить текст, що квазіреферується.

## 2. Зважування міри TF-IDF

Для задач планування у семантичному просторі про стан мети Goal наперед щось важко сказати. Наприклад, для задачі реферування текстових документів станом мети є кінцевий реферат, однак ми лише можемо собі уявляти, як він приблизно має виглядати. Оцінювання

стану в такій задачі збігається з оцінюванням важливості концепту (слово, лексема, речення), залежно від задачі [8].

Отже, для оцінки станів необхідно використати інший метод. Запропоновано такий метод – зважування міри TF-IDF адаптивною онтологією ПО.

TF-IDF (від англійського TF – term frequency, IDF – inverse document frequency) – статистична міра, що використовується для оцінювання важливості слова в контексті документа. Важливість деякого слова пропорційна кількості його вживання у документі і обернено-пропорційна частоті вживання слова у інших документах колекції. Ця міра часто використовується у задачах аналізу текстів та інформаційного пошуку, наприклад, як один з критеріїв релевантності документа пошуковому запиту, під час розрахунку міри близькості документа, під час кластеризації.

TF (term frequency – частота слова) – відношення числа входження деякого слова до загальної кількості слів документа. Отже, оцінюється важливість слова  $a_i$  в межах окремо-

го документа:  $TF = \frac{n_i}{\sum_k n_k}$ , де  $n_i$  – кількість вживання слова у документі, а у знаменнику –

загальна кількість слів у цьому документі. IDF (inverse document frequency – зворотна частота документа) – інверсія частоти, з якою деяке слово зустрічається у документах

колекції. Врахування IDF зменшує важливість широкоживаних слів:  $IDF = \log \frac{|T|}{|T_j \supset a_i|}$ , де

$|T|$  – кількість текстових документів у колекції;  $|T_j \supset a_i|$  – кількість текстових документів, в яких зустрічається слово  $a_i$  (коли  $n_i \neq 0$ ).

Отже, міра TF-IDF є добутком двох множників: TF і IDF:  $TF-IDF = TF \cdot IDF$ .

Більшу важливість у TF-IDF отримують слова з високою частотою у межах конкретного документа і з низькою частотою вживання в інших документах.

Для оцінювання станів зважуватимемо міру TF-IDF важливістю понять, відображених в адаптивній онтології, тобто  $v(St) = (TF-IDF) \cdot W$ .

Таке оцінювання містить істотні переваги порівняно з іншими оцінюваннями, оскільки у ній одночасно враховується як частотний аналіз зустрічання термінів у тексті (TF-IDF), так і специфіка ПО, до якої належить тематика цього тексту.

### 3. Процес квазіреферування

Перероблення інформації, яку подано у вигляді текстів природною мовою, має багато аспектів. Сюди належать такі види інформаційних процесів, як розуміння текстів, їх переклад, стиснення семантичної інформації. Особливе значення має останній тип перероблення; сюди входять класифікація і індексування документів, їх анотування та реферування.

Завдання автоматизації реферування текстової інформації сьогодні залишається дуже актуальним, незважаючи на величезну кількість робіт, що зроблені за останні роки в цьому напрямі. Це зумовлено, насамперед, необхідністю в умовах постійного зростання інформації ознайомлювати спеціалістів та інших зацікавлених людей з необхідними їм документами, поданими стисло, але із збереженням їх змісту. Крім того, анотування й реферування є невід’ємною частиною сучасного видавничого процесу. Будь-яке видання, чи це монографія, підручник, аналітичний огляд тощо, завжди випереджуються вторинним документом (рефератом або анотацією). Реферування використовується не тільки для економії часу під час ознайомлення з великою кількістю джерел, але й з метою пришвидшення повнотекстового пошуку по множині документів, оскільки обсяг реферату у декілька разів менший, ніж обсяг вхідного документа чи їх множини.

Реферування - це одержання найважливішої інформації з одного або декількох джерел для складання їхньої скороченої версії для потреб певних користувачів або задач [9].

Реферат - це семантично адекватний виклад основного змісту первинного документа, що відрізняється ощадливим знаковим оформленням, сталістю лінгвістичних і структурних характеристик і призначений для виконання різноманітних інформаційно-комунікативних функцій у системі наукової комунікації.

Текст складається з послідовності речень  $A_1, A_2, \dots, A_k$  та утворює кортеж  $T = (A_1, A_2, \dots, A_k)$ , а речення  $A_i, i = \overline{1, k}$  – з послідовності слів  $a_{ij}, i = \overline{1, k}, j = \overline{1, n}$ , яке, своєю чергою, зображається кортежем  $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$ . Зміст (семантику) тексту  $T$  позначимо  $S(T)$ .

Реферат (summary) текстового документа  $T$  позначимо  $\hat{T}$  і визначимо як текст, який містить кортеж  $\hat{T} = (\hat{A}_1, \hat{A}_2, \dots, \hat{A}_1)$ ,  $1 \leq k$  і зберігає семантику початкового текстового документа  $T$ . Формально це запишемо як  $\text{Summary}: T \rightarrow \hat{T}$ ,  $S(T) = S(\hat{T})$ .

Побудова реферату залежить від правильного оцінювання понять (ключових слів), словосполучень ПО та вибору на основі їх ключових речень.

Отже, оцінкою речень, що входять у текстовий документ, запропоновано вибрати добуток двох важливостей TF-IDF та важливості термінів  $W$  в онтології, що відповідає темі, якій належить запропонований до розгляду документ, тобто  $\varphi = (\text{TF-IDF}) \cdot W$ .

Така оцінка містить істотні переваги порівняно з іншими оцінками, оскільки у ній одночасно враховується як частотний аналіз вживання термінів у тексті (TF-IDF), так і специфіка ПО, до якої належить тематика цього тексту.

Для відбору речень для квазіреферату за основу взято відомий алгоритм просторового ранжування. Його модифіковано з врахуванням важливостей термінів тематики, які зберігаються в онтології ПО. Цей алгоритм ранжування зв'язних структур є універсальним алгоритмом ранжування об'єктів з врахуванням їх внутрішньої зв'язкової структури. Об'єкти зображені векторами у просторі Евкліда. У цьому разі вважається, що "близькість" двох об'єктів, зображених векторами, можна обчислити, як Евклідову міру або скалярний добуток векторів. Метою алгоритму є впорядкувати об'єкти з врахуванням внутрішніх зв'язків об'єктів між собою. Формально зв'язну структуру об'єктів зображають як деякий зважений граф, вершинами якого є самі об'єкти, а важливостями дуг задаються відстані Евкліда між об'єктами. У разі ранжування речень з метою відбору найзначущих з них для побудови квазіреферату алгоритм виглядатиме так:

1. Задається текст (набір речень)  $T = (A_1, A_2, \dots, A_k)$ , тематика  $T = (A_1, A_2, \dots, A_k)$ , до якої належить цей текст  $T \in Th_1$ . Згідно з тематикою з онтології ПО вибираються відповідні важливості понять та зв'язків  $W_{11}, W_{12}, \dots, W_{ln}, L_{11}, L_{12}, \dots, L_{lm}$ .

2. Вводиться  $\varphi: T \rightarrow R$  – відображення, яке ставить у відповідність кожній точці  $A_i, i = 1, 2, \dots, k$  значення рангу  $\varphi_i$ . Ми можемо розглядати  $\varphi_i$  як вектор  $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_k)^T$ .

3. Кожне речення (об'єкт) подають у векторному просторі так:  $x_i = (\varphi_{i1}, \varphi_{i2}, \dots, \varphi_{in})^T$ , де  $\varphi_{ij} = (\text{TF-IDF})_{ij} \cdot W_{ij}$  – міра відносної важливості терма  $a_{ij}$ .

4. Набір речень являє собою зважений граф з матрицею важливостей  $X = (x_{ij})$ . Для кожної пари  $x_i$  та  $x_j$  речень обчислюється важливість їх "лексичної близькості" за допомо-

гою стандартної Евклідової міри:  $x_{ij} = \frac{(x_i, x_j)}{|x_i| \cdot |x_j|}$ .

Зауважимо, що діагональні елементи матриці  $x_{ii} = 0$ , щоб отриманий граф не містив циклів. Потрібно зазначити, що отримана матриця важливостей є симетричною відносно своєї головної діагоналі.

Матриця важливостей піддається симетричній нормалізації  $S = D^{-\frac{1}{2}} X D^{\frac{1}{2}}$ , де  $D = (d_{ij})$  – діагональна матриця, де її діагональні елементи  $d_{ii}$  дорівнюють сумі елементів  $i$ -го рядка

матриці  $\chi$ . Нормалізація матриці необхідна для того, щоб ітеративний алгоритм збігався. Значення  $\varphi$  обчислюється як результат ітеративного процесу:  $\bar{\varphi}(t+1) = \alpha \cdot S \cdot \bar{\varphi}(t) + (1-\alpha) \cdot \bar{y}$ , де  $\bar{y}$  – одиничний вектор.

Згідно з теоремою, наведеною в [10], такий ітеративний процес збігається з  $\varphi^*$ . Отже,  $\varphi_i^*$  – отриманий ранг речення  $A_i$ . Алгоритм полягає в поступовому розповсюдженні об'єктами свого рангу на суміжні об'єкти-вершини. Отже, ранг  $\varphi^*$  кожного речення  $A_i$  обчислюється не лише з врахуванням “близькості” його до еталонного об'єкта (важливостей тематики  $T_h$  в онтології  $O$ ), але й із врахуванням зв'язної структури тексту, тобто ранг “поширюється” по графу з врахуванням важливостей зв'язків структур.

Розроблено систему квазіреферування на основі розробленого методу. Квазіреферування – це перший з двох етапів реферування, який полягає у відборі значущих речень із всього текстового документа. Другий етап полягає у зв'язуванні цих речень (у цій роботі даний етап не розглядається, ця задача належить до компетенції прикладної лінгвістики).

Система шукає у вхідному тексті головне речення і формує квазіреферат з указанням смислових класів. Система використовує морфологічний і гіперсинтаксичний засоби “розуміння” тексту. Перевірка гіпотези здійснювалася на масиві 20 довільно відібраних статей за тематикою інформаційних технологій. Були введені такі якісні характеристики квазірефератів: а) повнота передавання основного змісту документа; б) точність – відсутність у квазірефераті речень, надлишкових для передавання основного змісту документа; в) зв'язність (у звичайному розумінні цього слова). Були також введені такі кількісні оцінки кожної з перелічених характеристик квазірефератів: 1 – дуже погано; 2 – погано; 3 – задовільно; 4 – добре; 5 – відмінно. Квазіреферати оцінював автор, тобто людина, яка знає мову, але не обізнана зі змістом тексту, що реферується. Оцінки виставляли винятково з погляду майбутнього користувача системи, в припущенні, що квазіреферат в ідеалі повинен мати статус самостійного документа, тобто давати користувачеві чітке уявлення про тему вхідного документа, інформувати про його основний зміст, але не містити при цьому надлишкової інформації, відрізняючись тим самим від повного документа. Документи, що опрацьовувалися, були поділені на два класи: (а) які піддаються інтелектуальному реферуванню і (б) які не піддаються інтелектуальному реферуванню (наприклад, таблиця порівнянь швидкостей процесорів).

Обсяг одержаних квазірефератів – від трьох до шести речень; у двох випадках обсяг становив 7 речень: це були документи, котрі не підлягають інтелектуальному реферуванню. Отже, експеримент дав змогу зробити такі висновки. Одержані квазіреферати містять мало надлишкової інформації, а її наявність спричинена переважно помилками, не пов'язаними з якістю нашої моделі. Речення, що входять у квазіреферат, містять, як правило, основну інформацію вхідного тексту, тобто відповідають визначенню головного речення. Кількість головних речень, як правило, становить не більше 25 % всіх речень цього тексту: коефіцієнт стиснення, менший, ніж 4, одержаний тільки для дуже коротких текстів. Припущення про те, що з головних речень може бути складений новий текст, який має власну гіперсинтаксичну структуру, частково спростовують результати експерименту: 3 реферати з 20 одержали низьку оцінку за параметром “зв'язність”, тобто ці реферати мають вигляд скоріше штучних об'єднань речень, які належать до однієї теми, ніж до тексту. З іншого боку, основною причиною цього були зовнішні для нашої моделі чинники, тому треба вважати одержаний результат попереднім і таким, що потребує додаткової перевірки.

### **Висновки**

Розроблено метод квазіреферування текстових документів на основі зважування міри TF-IDF вагами важливості елементів адаптивної онтології ПО тематики, до якої належить текстовий документ. Для цього запропоновано зважувати терміни ПО та зв'язки між ними в межах онтології. Відповідне програмне забезпечення, яке реалізує розроблений метод, написано на мові програмування C#. Побудований на основі такого підходу квазіреферат показав задовільну якість. Перспективою подальших досліджень є квазіанотування та повноцінне реферування на основі розробленого квазіреферату шляхом зв'язування речень в єдину семантику за допомогою використання методів прикладної лінгвістики.

**Список літератури:** 1. *Литвин В.В.* Бази знань інтелектуальних систем підтримки прийняття рішень / В.В.Литвин. Львів: Видавництво Львівської політехніки, 2011. 240 с. 2. *Інтелектуальні системи, базовані на онтологіях* // Д.Г. Досин, В.В. Литвин, Ю.В. Никольський, В.В. Пасічник. Львів: "Цивілізація", 2009. 414 с. 3. *Lytvyn V.* Design of intelligent decision support systems using ontological approach / V.Lytvyn // An international quarterly journal on economics in technology, new technologies and modelling processes. Lublin. 2013. Vol. II, No 1. P. 31-38. 4. *Литвин В.В.* Підхід до побудови інтелектуальних систем підтримки прийняття рішень на основі онтологій // Проблеми програмування : наук. журн. / Національна академія наук України; Інститут програмних систем. Київ, 2013. №4. С. 43-52. 5. *Литвин В.В.* Метод моделювання процесу підтримки прийняття рішень у конкурентному середовищі / В.В.Литвин, О.В.Оборська, Р.В. Вовнянка // Математичні машини й системи : наук. журн. Київ, 2014. №1. С. 50-57. 6. *Lytvyn V.* Definition of the semantic metrics on the basis of thesaurus of subject area / V.Lytvyn, O.Semotuyk, O.Moroz // An international quarterly journal on economics in technology, new technologies and modelling processes. Lublin. 2013. Vol. II, No 4. P. 47-51. 7. *Досин Д.Г.* Архітектура інтелектуальної системи інформаційного пошуку в мережі Інтернет/ Д.Г. Досин, В.М. Ковалевич //Штучний інтелект. 2012. №3. С. 241-252. 8. *Даревич Р. Р.* Оцінка подібності текстових документів на основі визначення інформаційної ваги елементів бази знань / Р. Р. Даревич, Д. Г. Досин, В. В. Литвин, З. Т. Назарчук // Штучний інтелект: наук.-техн. журн. / Національна академія наук України; Інститут проблем штучного інтелекту. Донецьк, 2006. № 3. С. 500–509. 9. *Белоногов Г.Г.* Компьютерная лингвистика и перспективные информационные технологии / Г.Г. Белоногов, Ю.П. Калинин, А.А. Хорошилов. М.: Русский мир, 2004. 246 с. 10. *Zhou J.* Ranking on data manifolds / J. Zhou, A. Weston O. Gretton, B. Scholkopf // In Proceedings of NIPS. 2003. P. 234–237.

*Надійшла до редколегії 13.02.2014*

**Черна Тарас Ігорович**, аспірант кафедри інформаційних систем Національного університету „Львівська політехніка”. Наукові інтереси: побудова інтелектуальних систем. Адреса: Україна, 79000, Львів, вул. С. Бандери, 12, тел. (032) 258-25-38.

---

---

УДК 544.636

*Ф.В. ФОМОВСКИЙ*

## **МОДЕЛИРОВАНИЕ ПРОЦЕССА РАЗРЯДА ПЕРЕЗАРЯЖАЕМЫХ ИСТОЧНИКОВ ЭНЕРГИИ С ПОСТОЯННОЙ МОЩНОСТЬЮ**

---

Рассматривается общий подход к описанию работы как источников питания, обладающих высокой удельной энергоемкостью – аккумуляторов, так и источников с высокой удельной мощностью – конденсаторов и суперконденсаторов. В рамках рассмотренной модели получены аналитические зависимости между удельной энергией и удельной мощностью для основных классов перезаряжаемых источников энергии. Установлена связь между параметрами, описывающими накопитель электроэнергии, и предельной удельной мощностью этого накопителя.

**Введение.** Накопление, хранение и рациональное использование накопленной электрической энергии – одна из самых актуальных задач современной энергетики. Традиционно для хранения и последующего использования электрической энергии используются аккумуляторы, конденсаторы и суперконденсаторы. Однако новые требования, которые выдвигаются к перезаряжаемым источникам электрической энергии, невозможно удовлетворить только путем использования любого из указанных источников питания в отдельности. Повышенные практические требования, которые предъявляются к современным источникам электропитания (масштабные характеристики, диапазон рабочих температур, срок эксплуатации), указывают на необходимость создания гибридных источников питания, которые включали бы в себя различные перезаряжаемые источники питания. Только одновременное, в одном блоке питания, использование аккумуляторов, конденсаторов и суперконденсаторов позволяет решать современные задачи обеспечения электроэнергией гибридный автотранспорт, ветровые генераторы, солнечные электростанции.

Проектирование современных высокоэффективных гибридных источников питания первоочередной задачей ставит моделирование процессов согласования различных перезаряжаемых источников питания. Первым шагом в этом направлении можно считать описание перезаряжаемых источников питания в рамках зависимости, предложенной Рейгоном [1,2].

Д. Рейгон предложил характеризовать источники энергии зависимостью удельной энергии, которую источник способен отдать на внешнюю нагрузку с постоянной удельной мощностью. Для наполнения содержанием данного предложения была проделана огромная экспериментальная работа. В качестве примера можно отметить, что для аккумуляторов удельная энергия колеблется в пределах от  $10^4$  Дж/кг до  $10^6$  Дж/кг при удельной мощности порядка 100 W/кг, а у конденсаторов удельная энергия не превышает 100 Дж/кг при удельной мощности от  $10^4$  W/кг до  $10^6$  W/кг. Однако несмотря на существенные успехи, достигнутые при решении указанных задач, остается открытым вопрос о создании модели, которая бы описывала процесс передачи энергии в приближении, предложенном Д. Рейгоном.

*Цель работы.* В рамках единого модельного представления описать процесс передачи энергии для основных перезаряжаемых источников энергии.

Для решения поставленной задачи рассмотрим процесс разряда источника электроэнергии на внешнюю нагрузку. Разряд протекает на переменную внешнюю нагрузку, которая меняется таким образом, что энергия на нагрузке выделяется с постоянной мощностью. Схема эксперимента приведена на рис. 1.

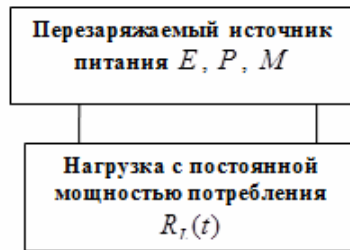


Рис. 1. Схема разряда источника питания на внешнюю нагрузку с постоянной мощностью

Как видно из рис. 1, источник питания и внешняя нагрузка подключены последовательно. Источник питания характеризуется полной запасенной энергией ( $E$ ), мощностью, с которой энергия передается на внешнюю нагрузку ( $P$ ), и массой ( $M$ ). Внешняя нагрузка представляет собой переменное сопротивление ( $R_L$ ). Сопротивление внешней нагрузки меняется таким образом, что энергия выделяется на нем с постоянной мощностью. Уравнение, которое описывает изменение тока в системе, как функцию времени, имеет вид:

$$U(t) = I(t)(R_L(t) + R_{in}), \quad (1)$$

где  $U(t)$  – напряжение источника питания;  $I(t)$  – ток, протекающий в системе;  $R_L(t)$  – сопротивление нагрузки;  $R_{in}$  – внутреннее сопротивление источника питания.

При разряде перезаряжаемого источника питания падение потенциала на внешней нагрузке равняется  $I(t)R_L(t)$ . Учитывая особенности постановки эксперимента, следует отметить, что разряд протекает через переменное сопротивление, величина которого меняется таким образом, что мощность, которая на нем выделяется, остается постоянной.

Ее можно записать в виде  $\frac{P}{I(t)}$ , где  $P$  – мощность, с которой происходит разряд. Окончательно уравнение (1) принимает вид:

$$U(t) = I(t)R_{in} + \frac{P}{I(t)}. \quad (2)$$

В рамках данного подхода при вычислении зависимости удельной плотности энергии как функции от удельной мощности вся исходная информация о перезаряжаемых источниках питания, содержащаяся в зависимости  $U(t)$  и  $R_{in}$ , описывается при помощи этих параметров. Изменение  $U(t)$  описывает разряд того или иного идеального перезаряжаемого источника питания.

*Аккумуляторы* выделяются среди других перезаряжаемых источников энергии своей высокой удельной энергоемкостью. В гибридных источниках энергии аккумуляторы являются основой энергонакапливающего элемента. Поэтому моделирование и рассмотрение перезаряжаемых источников начнем именно с аккумуляторов.

Рассмотрим разряд с постоянной мощностью идеального аккумулятора. Под идеальным подразумеваем такой аккумулятор, который не боится короткого замыкания. У идеального аккумулятора отсутствуют токи саморазряда и  $U(t) = U_0\theta(t - t_0)$ , где  $U_0$  – напряжение на внешних клеммах аккумулятора;  $t_0$  – время полного разряда аккумулятора.

Энергоемкость аккумулятора ( $E$ ) равняется:  $E = U_0Q$ , где  $Q$  – максимальный заряд, который можно запасти в аккумуляторе. Следовательно, время разряда равно  $t_0 = \frac{Q}{I(t)}$ .

Далее из уравнения (2) находим значение тока. Физический смысл имеет значение тока  $I(t)$ . Принимая во внимание, что  $E = Pt_0$ , получаем окончательное выражение, связывающее энергию, накопленную в аккумуляторе, и мощность, с которой эта энергия может быть передана на внешнюю нагрузку. Разделив левую и правую часть уравнения на массу аккумулятора, получим выражение, связывающее удельную энергию, выделяющуюся на внешней нагрузке, с удельной мощностью:

$$e_b = \frac{2QR_{in}M_b p_b}{U_0(1 - \sqrt{1 - \gamma})}, \quad \gamma = \frac{4R_{in}M_b p_b}{U_0^2}, \quad (3)$$

где  $M_b$  – масса аккумулятора;  $e_b = \frac{E}{M_b}$  – удельная энергия;  $p = \frac{P}{M_b}$  – удельная мощность.

Из уравнения (3) следует, что максимальная удельная мощность, с которой аккумулятор может передавать энергию на внешнюю нагрузку, равняется  $p_b = \frac{U_0^2}{4R_{in}M_b}$ .

Такая мощность соответствует случаю, когда сопротивление внешней нагрузки равняется внутреннему сопротивлению аккумулятора.

Следует отметить, что данная модель не рассматривает фарадеевские процессы саморазряда вредных паразитных примесей, процессы концентрационной поляризации и другие паразитные процессы, приводящие к саморазряду аккумулятора. Влияние перечисленных паразитных процессов на энергетическое состояние аккумулятора обсуждается в работе [3].

*Конденсаторы* как перезаряжаемые источники энергии используются только в импульсной технике, в случае, когда необходимо сформировать и передать мощный, но короткий импульс. Конденсатор может передавать всю накопленную энергию, другими словами, он может работать в режиме короткого замыкания и это не сказывается на его работоспособности.

При рассмотрении удельных энергетических характеристик конденсаторов воспользуемся, как и ранее, уравнением (2). Изменение разности потенциалов на внешних контактах

конденсатора описывается уравнением  $U(t) = \frac{Q_0 - q(t)}{C}$ , где  $Q_0$  – заряд суперконденсатора

в начальный момент времени;  $q(t)$  – величина заряда, на которую уменьшается заряд конденсатора в процессе разряда за время  $t$ ;  $C$  – емкость конденсатора. После подстановки в уравнение (2) выражения, которое связывает  $U(t)$  с зарядом на конденсаторе,

и учитывая, что  $I(t) = \frac{dq(t)}{dt}$ , получаем следующее уравнение:

$$\frac{(Q_0 - q(t))}{C} \frac{dq(t)}{dt} - P - \left[ \frac{dq(t)}{dt} \right]^2 R_{in} = 0. \quad (4)$$

Из уравнения (4) находим:

$$\frac{dq(t)}{dt} = \frac{1}{2} \left[ \frac{Q_0 - q(t)}{C} - \sqrt{\left( \frac{Q_0 - q(t)}{C} \right)^2 - 4R_{in}P} \right]. \quad (5)$$

Из уравнения (5) находим время разряда конденсатора с постоянной мощностью. Умножив время разряда на значение мощности, с которой разряжается конденсатор, находим энергию, которая выделяется на внешней нагрузке в процессе разряда:

$$e_c = Cp_c [x + \sqrt{x^2 - x - 1}] + Cp_c \ln(\sqrt{x} - \sqrt{x - 1}),$$

$$p_c = \frac{P}{M_c}, \quad (6)$$

$$\chi = \frac{U_0^2}{4R_{in}M_cp_c},$$

где  $M_c$  – масса конденсатора;  $e_c = \frac{E}{M_c}$  – удельная энергия, переданная конденсатором

на внешнюю нагрузку;  $p_c = \frac{P}{M_c}$  – удельная мощность разряда конденсатора.

Максимальная удельная мощность, с которой при разряде на внешнюю нагрузку конденсатор может передать максимальное количество энергии, равна  $p_c = \frac{U_0^2}{4R_{in}M_c}$ . Это

соответствует случаю, когда сопротивление внешней нагрузки равняется внутреннему сопротивлению конденсатора.

Если учесть, что конденсаторы не предназначены для длительного хранения энергии, рассмотрение процессов, приводящих к саморазряду, не представляет практического интереса. Поэтому считается, что вся накопленная в конденсаторах энергия передается на внешнюю нагрузку и их КПД приближается к 98%.

*Суперконденсаторы* – относительно новый тип перезаряжаемых источников энергии, который занимает промежуточное место между аккумуляторами и обычными конденсаторами. Суперконденсаторы широко применяются в гибридном автотранспорте, компьютерной технике, космической и бытовой технике и т.д. [4, 5]. Суперконденсаторы, как и обычные конденсаторы, не боятся короткого замыкания, количество циклов заряд/разряд у них практически не ограничено и по оценкам достигает  $10^6$ . Они способны работать в широком температурном диапазоне от  $-50^\circ\text{C}$  до  $+70^\circ\text{C}$  и не нуждаются в профилактическом обслуживании в процессе эксплуатации.

Еще одной особенностью суперконденсаторов, в отличие от обычных конденсаторов, является то, что емкость суперконденсаторов зависит от разности потенциалов на их внешних контактах [6,7]. В случае суперконденсаторов зависимость этих перезаряжаемых источников энергии от потенциала аппроксимируется линейной функцией. Традиционно выражение для емкости суперконденсатора как функции  $U$  представляется в виде:

$$C(U) = C_0 + aU, \quad (7)$$

где  $C_0$  – емкость при нулевой разности потенциалов на клеммах суперконденсатора;  $a$  – коэффициент пропорциональности, который определяется как тангенс угла наклона графика  $C = f(U)$ .

После несложных преобразований уравнение (2) можно представить в виде:

$$\left[\frac{dU}{dt}\right]^2 (C_0 + 2aU)^2 R_{in} - (C_0 + 2aU)U \frac{dU}{dt} + P = 0. \quad (8)$$

В отличие от обычных электролитических конденсаторов при разряде с суперконденсаторов снимается 75% накопленной энергии. Исходя из условий тестирования, разряд суперконденсатора осуществляется до половины его начального рабочего напряжения.

С учетом сказанного выше и из уравнения (8) находим, что для суперконденсаторов удельная энергия связана с удельной мощностью уравнением:

$$e_{sc} = \frac{C_0 U_0^2}{2M_{sc}} \left[ a(1) + \frac{3}{4} - \frac{1}{4} a(4) \right] + \frac{C_0 U_0^2 p_{sc}}{2M_{sc}} \ln \left[ \frac{2(1-a(1))}{1-a(4)} \right] + \frac{aU_0^3}{3M_{sc}} \left[ \frac{7}{8} + [a(1)]^2 - \frac{1}{8} [a(4)]^2 \right],$$

$$a(1) = \sqrt{1-p_{sc}}, \quad a(4) = \sqrt{1-4p_{sc}}, \quad (9)$$

где  $p_{sc}$  – удельная мощность суперконденсатора;  $e_{sc}$  – удельная энергия суперконденсатора;  $M_x$  – масса суперконденсатора.

В работе не рассматриваются процессы, связанные с саморазрядом суперконденсаторов. Подробно эти вопросы изложены в работах [3, 8]. Учитывая то, что суперконденсаторы не предназначены для длительного хранения энергии, пренебрежение процессами саморазряда при построении теоретической модели не должно вносить существенную погрешность при описании работы суперконденсатора.

Полученные в рамках предложенной модели уравнения (3), (6) и (9) описывают зависимость удельной энергии от удельной мощности для основных перезаряжаемых источников энергии аккумуляторов, конденсаторов и суперконденсаторов. В целях проверки справедливости сделанных предположений были рассчитаны рейгоновские зависимости для различных перезаряжаемых источников питания. Результаты приведены на рис. 2.

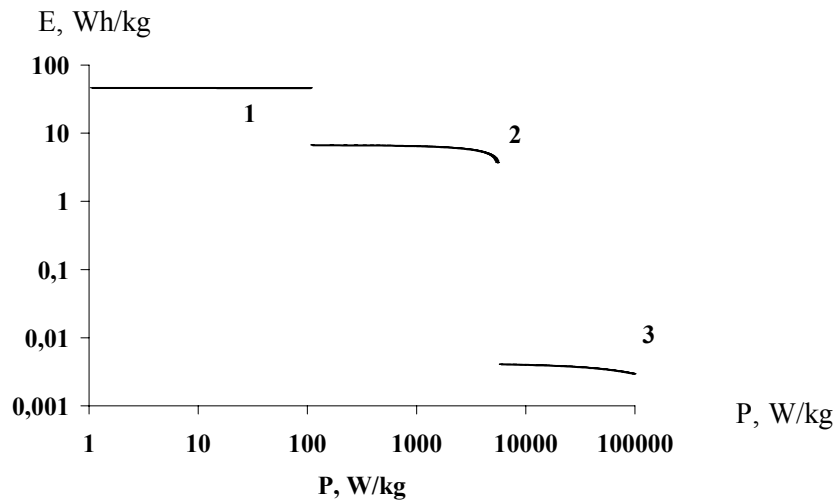


Рис. 2. Зависимость удельной энергии от удельной мощности для никель-металлгидридного аккумулятора (1), углерод/углеродного суперконденсатора (2) и электролитического конденсатора (3)

Результаты на рис. 2 приведены в традиционных для рейгоновских зависимостей логарифмических координатах, а значения удельной энергии даны в Вт-часах на килограмм. При расчетах зависимости, приведенной на рис. 2 для никель-металлгидридного аккумулятора, значения необходимых параметров были взяты из работы [9]. При расчетах рейгоновской зависимости для электролитического конденсатора использовали параметры конденсатора тайваньской фирмы NITANO емкостью 150 мкф и с рабочим напряжением 20В. Параметры макета суперконденсатора составляли емкость  $C_0 = 390$  Ф, коэффициент  $\alpha = 11$  Ф/В, рабочее напряжение – 2,7 В, внутреннее сопротивление – 1,4 мОм.

**Выводы.** 1. Предложенные результаты теоретических расчетов, показывают, что предположения, сделанные при построении теоретической модели, позволяют в рамках общего подхода описать энергетические характеристики основных перезаряжаемых источников питания.

2. При описании процессов перезарядки в перезаряжаемых источниках энергии индуктивностью можно пренебрегать, так как её влияние мало даже для электролитических конденсаторов.

3. Определено, что более важным представляется моделирование процессов теплообмена в источниках питания при их работе в критических режимах эксплуатации. Именно процессы теплообмена могут внести коррективы в определение предельных значений удельной мощности как для аккумуляторов, так и для суперконденсаторов.

В заключение автор выражает глубокую благодарность канд. хим. наук, ст. н.с. В. Ю. Изотову и д-ру физ.-мат.наук, проф. Н. И. Ключо за полезные обсуждения результатов работы и ценные замечания.

**Список литературы:** 1. *Ragone D.V.* Review of Battery Systems for Electrically Powered Vehicles // Mid-Year Meeting of the Society of Automotive Engineers, Detroit, MI, May 20–24. 1968. P 117-149. 2. *Webster J.G.* (Ed.), Wiley Encyclopedia of Electrical and Electronics Engineering vol. III Wiley, New York. 1999. 3. *Conway B.* Electrochemical supercapacitors: scientific fundamentals and technological applications. Kluwer Academic / Plenum New York. 1999. 698 p. 4. *Nishino A.* Capacitors: operating principles, current market and technical trends // Journal of Power Sources. 1996. V. 60. P. 137–147. 5. *Burke A.* R&D consideration for the performance and application of electrochemical capacitors // Electrochimica Acta. 2007. V. 53. P. 1083–1091. 6. *Kurzweil P., Frenzel B., Gallay R.* Capacitance characterization methods and ageing behavior of supercapacitors // Proc. The 15-th International Seminar On Double Layer Capacitors. Deerfield Beach, FL., U.S.A., Dec. 5–7, 2005. P. 1–12. 7. *Изотов В.Ю.* Залежність інтегральної ємності конденсаторів подвійного електричного шару від потенціалу // Наукові вісті НТУУ „КПІ”. 2011. № 3 (71). С. 119–122. 8. *Kowal J., Avaroglu E., Chamekh F, Senfelds A., Thien T., Wijaya D., Sauer D.U.* Detailed analysis of the self-discharge of supercapacitors // Journal of Power Sources. 2011. V. 196. P. 573-579. 9. *Andrew Burke.* Review. Ultracapacitor technologies and application in hybrid and electric vehicles // Int. J. Energy Res. 2010. V. 34. P.133–151.

Поступила в редколлегию 15.02.2014

**Фомовский Феликс Владимирович**, канд. техн. наук, доцент кафедры электронных аппаратов КрНУ им. М. Остроградского. Научные интересы: оборудование и технологии производства электронной техники. Адрес: Украина, 39600, Кременчуг, ул. Першотравнева, 20, тел.: (05366) 32001. E-mail: fill.fo@mail.ru.

## **ИСКЛЮЧЕНИЕ ЭФФЕКТА ОРЕОЛА ПРИ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ СВЕТОФОРНЫХ СИГНАЛОВ**

---

Рассматривается проблема исключения эффекта ореола, вызывающего искажение размера и формы сегментов при распознавании светофорных сигналов. Исследуется общая схема процесса распознавания и формируется тестовый набор кадров. Описывается подход к устранению эффекта ореола с помощью удаления области рассеяния и заполнения засвеченного сегмента. Приведенный подход повышает устойчивость процесса распознавания.

### **1. Введение**

Предотвращение дорожно-транспортных происшествий является одной из важнейших проблем обеспечения транспортной безопасности. За последние годы все большее внимание уделяется интеллектуализации транспортных средств (ИТС), как инструменту повышения контроля дорожной обстановки. Интеллектуализация обеспечивает транспортное средство устройствами автоматизации с использованием мехатроники, искусственного интеллекта или мультиагентных систем для поддержки водителя. Интеллектуализация обеспечивает различную степень автоматизации и имеет широкую область применения, начиная с упрощения процесса парковки и заканчивая полностью автоматической ездой. Существенное достоинство ИТС проявляется при использовании транспортного средства людьми с нарушениями функций восприятия. Например, водителям, не различающим цвета, очень сложно ориентироваться в городской обстановке с огромным количеством разнообразных знаков и светофоров. ИТС в таком случае может обеспечивать дублирование сигналов текстом или голосом. В настоящее время ИТС чаще всего применяется в рекомендательных и контролируемых целях, но иногда такие средства могут напрямую вмешиваться в действия водителя, например, снижать скорость движения. Для того чтобы данные средства функционировали корректно, в соответствии с правилами дорожного движения, используют ряд различных датчиков, таких как радар, сонар, лидар и др.

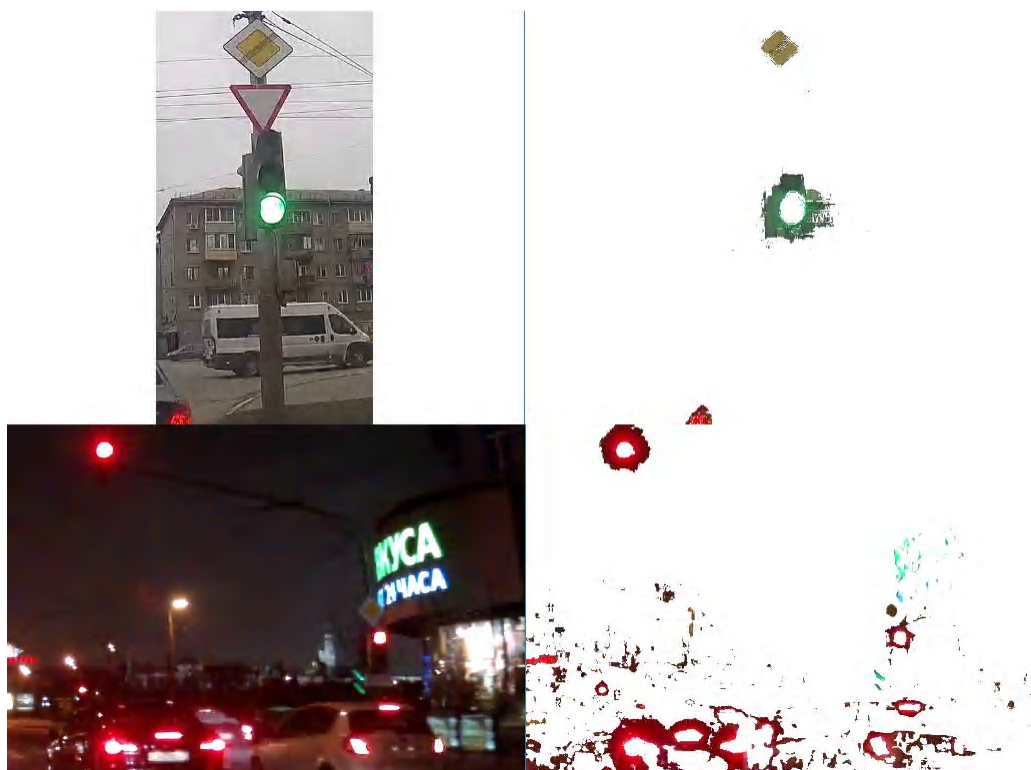
Важным элементом регулирования дорожного движения являются светофоры. Информация о состоянии светофора может быть получена только с помощью модуля фотофиксации. Разработка такого модуля сопряжена со множеством технических трудностей, которые могут значительно повлиять на его надежность. В частности, наиболее значительными проблемами являются перспективные искажения формы светофорных ламп, частичное их перекрытие и плохие погодные условия. Качество распознавания сигналов светофоров во многом определяет надежность транспортных средств с системами интеллектуализации, что особенно характерно для городской обстановки.

Общепринятой практикой при обнаружении светофора является процесс цветовой сегментации [1-8]. Сегмент представляет собой группу пикселей с однородными цветами. Данный способ предусматривает, что светофоры имеют фиксированный диапазон цветов. Светофорная лампа характеризуется яркими и насыщенными цветами для облегчения восприятия водителями, однако в системе цифрового восприятия цветовая детекция очень чувствительна к изменениям освещения, например, из-за погодных условий. Кроме того, различные лампы могут приводить к различным цветовым вариациям, что обусловлено особенностями функционального исполнения. Таким образом, сама по себе цветовая детекция не является стабильным способом обнаружения и распознавания сигнала светофора.

Кроме цвета, дополнительным параметром для выявления или идентификации светофора служит форма сегмента [6-8]. Некоторые методы применяют обобщенное преобразование Хафа или генетический алгоритм для извлечения эллипсов на изображении, как предпо-

лагаемых сегментов светофорной лампы. Подходы, которые используют округлость формы, могут не работать, если светофоры частично перекрыты другими объектами.

При выделении сегментов изображения могут возникнуть нарушения форм светящихся объектов из-за присущего им эффекта ореола. В результате, рассеяние яркости приводит к ошибочному увеличению размеров сегментов. Кроме того, в некоторых условиях освещенности ореол объекта предоставляет больше полезной информации, чем сам объект. Такая ситуация возникает по причине частичной засветки матрицы камеры, имеющей узкий диапазон экспозиции. На рисунке приведен пример ошибочной сегментации.



Ошибки цветовой сегментации

Процесс распознавания под влиянием данного фактора может значительно потерять устойчивость при различной освещенности.

*Целью исследования* является усовершенствование существующего метода распознавания сигнала светофора для исключения эффекта ореола при распознавании светофорных сигналов, что позволит повысить устойчивость процесса распознавания в различных условиях освещенности и, как следствие, улучшить надежность перемещения транспортных средств.

## 2. Общая схема процесса распознавания

Перед рассмотрением подходов к исключению эффекта ореола нам необходимо исследовать общую схему [3] распознавания сигнала светофора. Система распознавания оперирует с дискретными кадрами видеопотока и включает в себя два крупных модуля – обнаружения светофора и цветовой классификации. Процесс распознавания включает в себя следующие этапы:

- 1) конвертация пикселей изображения из цветового пространства RGB в HSI;
- 2) преобразование в бинарное изображение. Пиксель бинарного изображения будет установлен в 1, только если значения тона, насыщенности и интенсивности соответствующего пикселя исходного изображения будут входить в заданные интервалы. Интервалы определяются исходя из распределения цветов тестового набора кадров;
- 3) морфологические операции сужения (erosion) и расширения (dilation). В качестве структурирующего элемента используется матрица размером  $3 \times 3$ ;

4) сегментация и фильтрация по размеру сегмента. Сегменты образуются из групп соединенных пикселей со значением 1. Для нахождения сегмента применяется рекурсивный обход пикселей-соседей. Сегменты, у которых размер превосходит заданное пороговое значение, удаляются. На данном этапе при распознавании сигнала светофора в условиях плохого освещения, а также при неудовлетворительных характеристиках камеры возникает проблема искажения изображения – возникновения ореола, что требует выполнения дополнительных операций по обработке изображения;

5) фильтрация сегментов формы. Сегменты, у которых форма не соответствует эллипсу, удаляются;

6) цветовая классификация. Для определения цвета используются зависимости RGB компонент, выраженные в форме нестрогих неравенств. Если ни одно неравенство не выполняется, то система распознавания сигнализирует об отсутствии светофоров в кадре.

В работе [3] показано, что процесс распознавания в реальном времени, применительно к набору кадров с точно настроенной экспозицией камеры, дает не менее 97% точности и 91% полноты. Однако незначительное изменение параметров камеры или быстрая смена условий освещения вызывает появление ощутимого числа ложных срабатываний. При этом значение точности и полноты равно 50 и 90% соответственно.

### 3. Подход к исключению эффекта ореола

Для исключения эффекта ореола предлагается расширить этап 4 процесса распознавания следующей процедурой:

- 1) определение общей освещенности сегмента;
- 2) поиск и заполнение засвеченных сегментов:
  - а – выделение засвеченных пикселей;
  - б – сегментация [3];
  - в – нахождение цвета соответствия подсегментов;
  - г – заполнение цветом засвеченных подсегментов;
- 3) сегментация по оператору Лапласа;
- 4) удаление подсегмента области рассеяния.

Нахождение общей освещенности сегмента выполняется следующим образом:

$$L_S = \sum_{(x,y) \in S} I_{(x,y)} / C_S, \quad (1)$$

где  $L_S$  – общая освещенность сегмента;  $S$  – множество пикселей сегмента;  $(x,y)$  – координаты пикселя;  $I_{(x,y)}$  – интенсивность пикселя;  $C_S$  – размер сегмента.

Далее, если значение освещенности удовлетворяет условию:

$$L_S < Tr_{LS}, \quad (2)$$

здесь  $Tr_{LS}$  – заданное пороговое значение, то выполняется операция сегментации по оператору Лапласа. Иначе, выполняется операция поиска и заполнения засвеченных сегментов.

Выделение засвеченных пикселей:

$$S_d = \{(x, y) \mid (x, y) \in S, R_{(x,y)} + G_{(x,y)} + B_{(x,y)} \geq Tr_{SD}\}, \quad (3)$$

где  $S_d$  – множество засвеченных пикселей;  $(x,y)$  – координаты пикселя сегмента;  $R_{(x,y)}$  – красная компонента цвета пикселя;  $G_{(x,y)}$  – зеленая компонента;  $B_{(x,y)}$  – синяя компонента;  $Tr_{SD}$  – пороговое значение.

На данном этапе выполняется процедура сегментации, описанная в работе [3], в результате которой определяются подсегменты  $S_{sd}$ , центры подсегментов  $(X_{sd}, Y_{sd})$  и центр сегмента  $(X_S, Y_S)$ .

Нахождение цвета производится следующим образом:

$$\overline{V}_{sd} = (X_{sd} - X_S; Y_{sd} - Y_S), \quad (4)$$

$$\overline{V_{(x,y)}} = (x - X_S; y - Y_S), \quad (5)$$

$$\alpha_{(x,y)} = \cos^{-1} \left( \frac{\overline{V_{sd}} * |\overline{V_{sd}}| * |\overline{V_{(x,y)}}|}{|\overline{V_{sd}}| * |\overline{V_{(x,y)}}|} \right), \quad (6)$$

$$S_{csd} = \{(x, y) \mid (x, y) \in S_{sd}, \alpha_{(x,y)} < Tr_{CSD}\}, \quad (7)$$

$$R_{sd} = \sum_{(x,y) \in S} R_{(x,y)} / C_{sd}, \quad (8)$$

$$G_{sd} = \sum_{(x,y) \in S} G_{(x,y)} / C_{sd}, \quad (9)$$

где  $V_{sd}$  – вектор между центром подсегмента и центром сегмента;  $\overline{V_{(x,y)}}$  – вектор между пикселем цвета и центром сегмента;  $\alpha$  – угол между этими векторами;  $S_{csd}$  – множество пикселей цвета;  $Tr_{CSD}$  – пороговое значение, определяющее область вокруг подсегмента, откуда будет извлекаться цвет;  $(R, G, B)_{sd}$  – компоненты цвета подсегмента.

После нахождения цвета все белые пиксели в подсегменте заполняются этим цветом и последний передается остальным этапам метода распознавания как обычный сегмент.

Для сегментации области рассеяния используется оператор Лапласа следующего вида:

$$D_{x,y}(I_{(x,y)}) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (10)$$

Полученные значения интенсивности пикселей затем приводятся к бинарному изображению по следующему условию:

$$P_{(x,y)} = \begin{cases} 0 & L_{x,y} < Tr_p \\ 1 & L_{x,y} \geq Tr_p \end{cases}, \quad (11)$$

где  $P$  – пиксель бинарного изображения;  $Tr_p$  – пороговое значение.

Данное бинарное изображение сегмента содержит границы подсегментов, которые затем отделяются друг от друга и передаются как обычные сегменты остальным этапам метода.

В конце оставшийся подсегмент области рассеяния удаляется:

$$S = S \setminus \neg S_d. \quad (12)$$

Для оценки процесса распознавания использовалась методика, описанная в работе [3].

Модифицированный процесс позволяет в среднем повысить точность распознавания сигнала светофора на 15-20%.

## Выводы

Усовершенствован метод распознавания сигнала светофора путем исключения эффекта ореола, который искажает форму и размер сегментов изображения. Для устранения ореола объекта производится общая оценка освещенности кадра, которая позволяет определить характер искажения – изменение области рассеяния яркости или засветка области светофорной лампы. Для устранения области рассеяния применяется процедура фильтрации по оператору Лапласа. Для заполнения засвеченной области сегмента используется усредненное значение цвета из области рассеяния. В результате усиливается устойчивость процесса цветовой сегментации, что позволяет более точно распознавать светофоры в условиях различной освещенности.

**Список литературы:** 1. *Gavrila D. M., Franke U., Gurzig S., Wuhler C.* Real-time Vision for Intelligent Vehicles // IEEE Instrumentation and Measurement Magazine. 2001. V. 4, № 2. P. 22–27. 2. *Cheng H D, Jiang XH, Sun Y.* Color image segmentation: advances and prospects // Pattern Recognition. 2001. V. 34, № 12. P. 2259–2281. 3. *Lu H., Wang C. M., Chen S. Y.* Traffic light recognition // Journal of the Chinese Institute of Engineers. 2008. V. 31, № 6. P. 1069–1075. 4. *Chiang C. C., Ho M. C., Liao H. S., Pratama A., Syu W. C.* Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts // International Journal of Innovative Computing, Information and Control. 2011. T. 7, № 12. С. 6919–6934. 5. *Yung N H C, Lai A H S.* An Effective Video Analysis Method for Detecting Red Light Runners // IEEE Transactions on Vehicular Technology. 2001. V. 50, № 4. P. 1074–1084. 6. *Yun-Chung Chung, Jung-Ming Wang, Sei-Wang Chen.* A Vision-based Traffic Light System at Intersections // Journal of Taiwan Normal University. 2002. V. 47, № 1. P. 67–86. 7. *Lindner F, Kressel U, Kaelberer S.* Robust Recognition of Traffic Signals: материалы конф. «IEEE Intelligent Vehicles Symposium». 2004. P. 434. 8. *Yehu Shen, Ozguner U, Redmill K.* A Robust Video based Traffic Light Detection Algorithm for Intelligent Vehicles: материалы конф. «IEEE Intelligent Vehicles Symposium». 2009. P. 521–526.

*Поступила в редколлегию 24.03.2014*

**Чальый Сергей Федорович**, д-р техн. наук, профессор кафедры ИУС ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-14-51.

**Демченко Антон Алексеевич**, студент магистратуры специальности Информационные управляющие системы, факультет КИ, ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, e-mail: ademchk@gmail.com.

---

---

УДК 629.391

*В.В. БАРАННИК, О.Ю. ОТМАН ШАДИ, А.В. ХАХАНОВА*

## **МЕТОД РЕКОНСТРУКЦИИ ТРАНСФОРМАНТ В ДИАГОНАЛЬНО-НЕРАВНОМЕРНОМ БАЗИСЕ ОСНОВАНИЙ НА ОСНОВЕ ДЕКОДИРОВАНИЯ НЕРАВНОМЕРНЫХ КОДОГРАММ**

---

Формулируются требования относительно реконструкции базовых кадров в условиях обеспечения заданного уровня коррекционных преобразований для контролируемости искажений при реконструкции кадров Р- и В-типа. Проводится разработка метода реконструкции базовых кадров на основе диагонально-неравномерного позиционного декодирования трансформант дискретного косинусного преобразования. Основными технологическими этапами являются: декомпозиция неравномерных кодовых конструкций компрессионного представления диагоналей трансформанты; диагонально-неравномерное позиционное декодирование для переменного количества элементов диагоналей трансформанты. Излагается создание технологии реконструкции кодограмм, содержащих кодовые значения диагональных неравномерных чисел. Технология базируется на таких этапах: декомпозиция служебной и информационной частей кодовой конструкции сжатого представления фрагмента базового кадра с последующей их разметкой на кодовые конструкции отдельных диагоналей; декомпозиция информационной части кодовой конструкции путем разметки фиксированного числа кодограмм компактного представления диагоналей трансформанты.

### **1. Введение**

Повышение качества и расширение видеоинформационных сервисов приводит к росту интенсивности видеопотока [1; 2], следствием чего является увеличение нагрузки на телекоммуникационные системы. Это диктует необходимость уменьшения интенсивности потока кадров в процессе их компрессии [3 - 5]. Здесь ключевая составляющая интенсивности потока определяется интенсивностью кодового представления базового кадра. Это обусловлено тем, что базовый кадр является опорным по формированию кадров Р- и В-типа [4 - 6]. Поэтому качество базового кадра в значительной степени влияет на качество визуального восприятия всего видеопотока. В связи с этим *актуальная задача научно-прикладных исследований* заключается в обеспечении реконструкции базового кадра с заданным уровнем достоверности.

Для обработки базового кадра используется принципиально новый метод снижения интенсивности его кодового описания. Метод базируется на диагонально-неравномерном позиционном кодировании трансформант в неравновесном базисе оснований. В то же время для такого метода отсутствует соответствующий метод восстановления. Отсюда **цель исследования** состоит в разработке метода реконструкции базового кадра видеопотока на основе неравномерно-диагонального декодирования.

## 2. Разработка метода реконструкции базового кадра видеопотока

В процессе разжатия базовых кадров необходимо учитывать требования относительно их формирования с заданным уровнем коррекционных преобразований. Это обеспечивает контролируемость искажений при реконструкции кадров Р- и В-типа. Для этого метод восстановления базовых кадров должен обладать следующими возможностями учитывать технологические особенности процесса снижения интенсивности битового потока по базовому кадру, а именно [5]:

1) трансформирование фрагментов изображений на основе двумерного дискретного косинусного преобразования;

2) определение кодовых значений для диагоналей трансформант в неравномерном неравновесном позиционном базисе;

3) построение кодограмм неравномерной длины, содержащих значение кода диагонально-неравномерных позиционных чисел.

В связи с этим предлагается организовывать процесс восстановления базовых кадров на основе:

а) беспогрешностного восстановления массивов служебных данных: базиса  $D'$  оснований ДНП чисел и матриц  $Z$  знаков;

б) декомпозиции неравномерных кодовых конструкций компрессионного представления диагоналей трансформанты;

в) выполнения взаимобратных преобразований на основе известной служебной информации, включающих в себя:

- статистическое декодирование низкочастотной ДС-компоненты;

- диагонально-неравномерное позиционное декодирование для переменного количества элементов диагоналей трансформанты;

- обратное двумерное дискретное косинусное преобразование с коррекцией компонент трансформант размерностью  $n \times n$ .

Рассмотрим особенности этапов.

Восстановление служебных данных. В соответствии с особенностями процесса обработки базовых кадров служебными данными являются:

1) базис  $D'$  оснований, который задает неравновесное диагонально-неравномерное позиционное пространство трансформанты. Информация о значениях оснований  $d'_\xi$  диагоналей трансформант используется на таких этапах процесса восстановления:

- разметка кодового потока базового кадра на кодовые конструкции трансформант с последующей их разметкой на кодовые конструкции отдельных диагоналей (рисунок);

- формирование неравномерных диагоналей трансформант на основе диагонально-неравномерного позиционного декодирования;

2) матрицы  $Z$  знаков компонент трансформанты ДКП. Данная информация необходима для определения знака компоненты трансформанты, т.е. получения такого представления трансформант  $Y'$ , для которого выполняется обратное ДКП. Матрицы знаков входят в состав компактного представления в виде кодограмм, сформированных для одномерных структурных чисел.

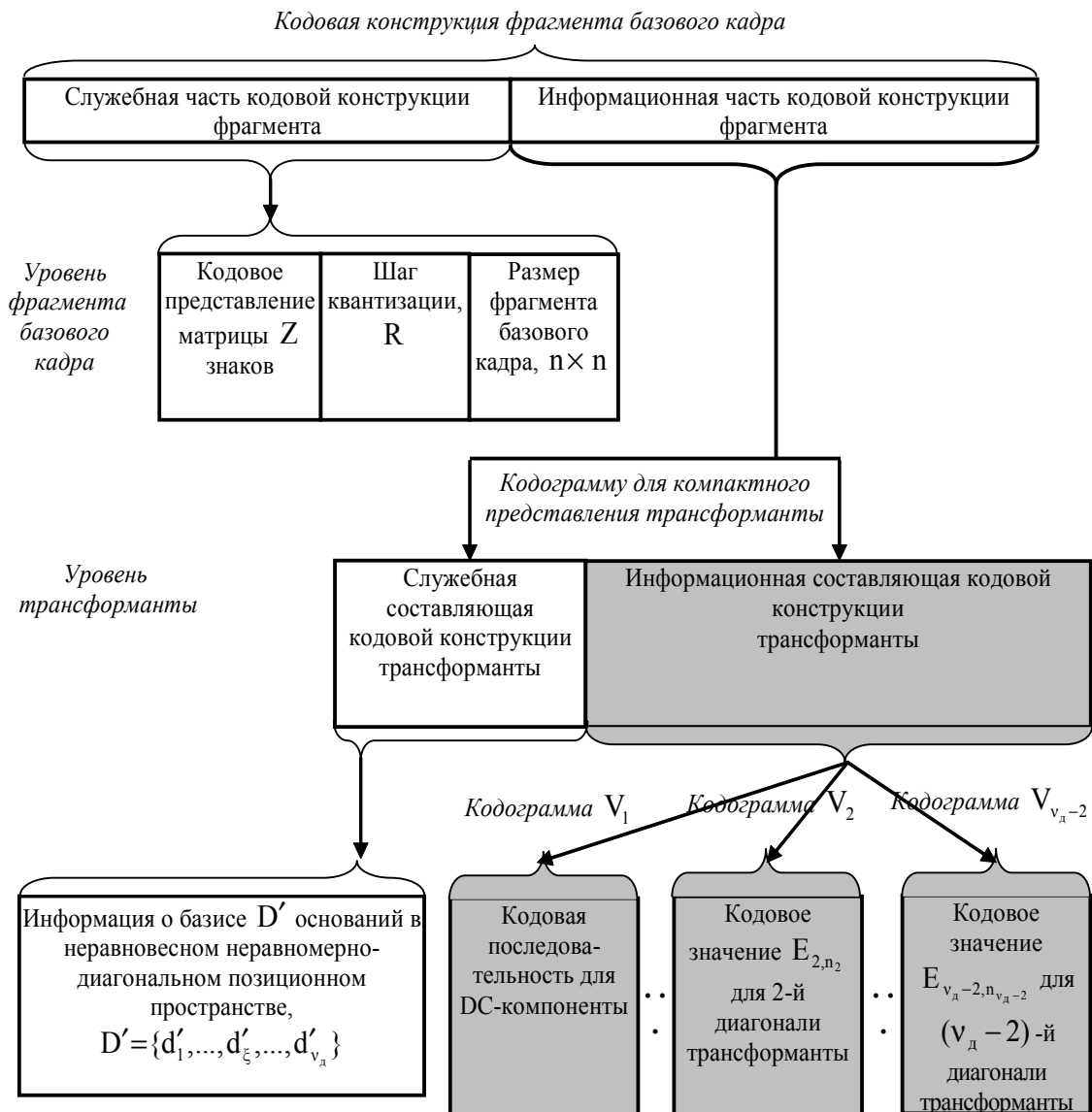
Реконструкция кодограмм, содержащих кодовые значения ДНП чисел (см. рисунок). Для этого требуется выполнить такие этапы:

1) вначале провести декомпозицию служебной и информационной частей кодовой конструкции сжатого представления текущего фрагмента базового кадра;

2) после этого провести декомпозицию информационной части кодовой конструкции путем разметки фиксированного числа кодограмм компактного представления диагоналей трансформанты.

Реализация первого этапа заключается в определении длины служебной части кодовой конструкции фрагмента базового кадра. Здесь учитываются такие особенности построения кодовых слов служебной части (см. рисунок):

- количество разрядов на представление информации о линейном размере  $n$  фрагмента базового кадра;
- кодовое представление для матрицы знаков формируется как для двоичной последовательности фиксированной длины. Это позволяет установить длину кодового слова матрицы знаков;
- количество разрядов на представление значения шага квантизации  $R$  является постоянной величиной.



Структурно-функциональная схема процесса восстановления фрагмента базового кадра

Выделение количества разрядов на представление служебной части позволяет установить начальную позицию информационной части кодовой конструкции фрагмента базового кадра.

Информационная часть кодовой конструкции фрагмента базового кадра представляет собой кодограмму для компактного представления трансформанты. Такая кодограмма содержит служебную и информационную части. Поэтому процесс декомпозиции информационной части подразумевает разметку соответствующей части кодовой конструкции на служебную и информационные составляющие (см. рисунок).

Служебная часть кодограммы формируется на базе кодовых полей, содержащих информацию о базисе  $D'$  оснований в неравновесном неравномерно-диагональном позиционном пространстве (см. рисунок). Количество оснований постоянно и зависит от размера трансформанты следующим образом:  $v_d - 2 = 2n - 3$ . Это позволяет формировать на их представление кодовые слова равномерной длины.

Информационная часть кодовой конструкции трансформанты включает в себя следующие составляющие:

1) кодовое слово  $V_1$  на сжатое представление DC-компоненты. Его длина определяется на базе использования статических таблиц, которые известны как на передающей, так и на приемной сторонах;

2) неравномерные по длине кодограммы  $V_\xi$ , несущие информацию о кодовых значениях  $\overline{E}'_{\xi, n_\xi}$  диагонально-неравномерных позиционных чисел. Количество  $v_k$  таких кодограмм фиксировано и определяется как  $v_k = v_d - 2 = 2n - 3$ .

Рассмотрим процесс выделения кодограмм ДНП чисел. Данный процесс реализуется на основе информации о базисе оснований с использованием следующих этапов:

а) вычисление накопленного произведения  $d_\xi^{n_\xi}$  оснований для текущей  $\xi$ -й диагонали трансформанты. Количество  $n_\xi$  элементов в диагонали определяется на основе соотношения

$$n_\xi = \begin{cases} \xi, & \rightarrow \xi \leq n; \\ n_\xi = 2n - \xi, & \rightarrow \xi \geq n+1; \end{cases} \quad (1)$$

б) нахождение количества  $\overline{Q}'_{\xi, n_\xi}$  разрядов на представление кодового значения  $\overline{E}'_{\xi, n_\xi}$  для  $\xi$ -го ДНП числа, т.е. для  $\xi$ -й диагонали трансформанты. Для этого используется соотношение:

$$\overline{Q}'_{\xi, n_\xi} = [\ell \log_2 (d_\xi^{n_\xi} - 1)] + 1 \text{ (бит)}.$$

Данное выражение позволяет определить количество  $\overline{Q}'_{\xi, n_\xi}$  разрядов на представление кодограмм для всех диагоналей трансформанты, где  $\xi = \overline{1, v_d - 2} = \overline{1, 2n - 3}$ .

После этого организуется переход на третий этап процесса реконструкции базового кадра, связанного с декодированием кодовых значений  $\overline{E}'_{\xi, n_\xi}$ . Для этого вначале по известному порядковому номеру  $\xi$  диагонали трансформанты определяется количество  $n_\xi$  содержащихся в ней элементов. Здесь используется соотношение (1). Далее на втором этапе организуется восстановление значений элементов диагоналей, начиная с ее начальных элементов. Данный этап организуется с использованием выделенного в кодограмме  $V_\xi$  кодового значения  $\overline{E}'_{\xi, n_\xi}$  на основе соотношения для получения элементов диагонально-неравномерных позиционных чисел, а именно:

1) для случая, когда для номера диагонали выполняется условие  $\xi \leq n$ , то  $n_\xi = \xi$  и:

$$y_{1+\tau, \xi-\tau} = \left[ \frac{\overline{E}'_{\xi, \xi}}{d_\xi'^{\xi-\tau-1}} \right] - \left[ \frac{\overline{E}'_{\xi, \xi}}{d_\xi' \cdot d_\xi'^{\xi-\tau-1}} \right] d_\xi', \quad \tau = \overline{0, \xi-1}, \quad \xi \leq n;$$

2) для случая, когда для номера диагонали выполняется неравенство  $\xi \geq n+1$ , то  $n_\xi = 2n - \xi$ . Тогда значение кода  $\overline{E}'_{\xi, 2n-\xi}$  определяется как:

$$y_{\xi-n+1+\tau, n-\tau} = \left[ \frac{\overline{E}'_{\xi, 2n-\xi}}{d_\xi'^{2n-\xi-\tau}} \right] - \left[ \frac{\overline{E}'_{\xi, 2n-\xi}}{d_\xi' \cdot d_\xi'^{2n-\xi-\tau}} \right] d_\xi', \quad \tau = \overline{0, 2n-\xi}, \quad \xi \geq n+1.$$

В обобщенной записи данные соотношения будут иметь следующий вид:

$$y_{1+\tau+\eta(\xi-n), \xi-\tau-\eta(\xi-n)} = \begin{cases} \left[ \frac{\overline{E}'_{\xi, \xi}}{d_\xi'^{\xi-\tau-1}} \right] - \left[ \frac{\overline{E}'_{\xi, \xi}}{d_\xi' \cdot d_\xi'^{\xi-\tau-1}} \right] d_\xi', & \rightarrow \eta=0 \ \& \ \xi \leq n; \\ \left[ \frac{\overline{E}'_{\xi, 2n-\xi}}{d_\xi'^{2n-\xi-\tau}} \right] - \left[ \frac{\overline{E}'_{\xi, 2n-\xi}}{d_\xi' \cdot d_\xi'^{2n-\xi-\tau}} \right] d_\xi', & \rightarrow \eta=1 \ \& \ \xi \geq n+1, \end{cases}$$

Если требуется использовать упрощенную схему алгоритмического процесса декодирования кодов ДНП чисел, то используется следующая формула:

$$y_{1+\tau+\alpha \text{sign}(1+\text{sign}(\alpha-1)), \xi-\tau-\alpha \text{sign}(1+\text{sign}(\alpha-1))} = \left[ \frac{\overline{E}'_{\xi, n_\xi}}{d_\xi'^{2n-\xi-\tau}} \right] - \left[ \frac{\overline{E}'_{\xi, n_\xi}}{d_\xi' \cdot d_\xi'^{2n-\xi-\tau}} \right] d_\xi';$$

$$\text{sign}(\alpha-1) = \begin{cases} 1, & \rightarrow \alpha > 1; \\ 0, & \rightarrow \alpha = 1; \\ -1, & \rightarrow \alpha \leq 0, \end{cases}$$

где  $n$  – линейный размер трансформанты;  $n_\xi$  – длина для  $\xi$ -й диагонали трансформанты;  $d_\xi'^{n_\xi-\tau-1}$  – весовой коэффициент для  $\tau$ -го элемента  $\xi$ -го ДНП числа.

Данный этап заканчивается после декодирования кодового значения  $\overline{E}'_{v_d-2, n_{v_d-2}}$ .

Последующие этапы процесса восстановления фрагмента базового кадра состоят в обратном трансформировании и воспроизведении фрагментов для исходной цветовой модели представления.

### 3. Выводы

1. Сформулированы требования относительно реконструкции базовых кадров в условиях обеспечения заданного уровня коррекционных преобразований для контролируемости искажений при реконструкции кадров Р- и В-типов.

2. Разработан метод реконструкции базовых кадров на основе диагонально-неравномерного позиционного декодирования трансформант дискретного косинусного преобразования. Отличительными технологическими этапами являются:

1) декомпозиция неравномерных кодовых конструкций компрессионного представления диагоналей трансформанты;

2) диагонально-неравномерное позиционное декодирование для переменного количества элементов диагоналей трансформанты; обратное двумерное дискретное косинусное преобразование с коррекцией компонент трансформант.

3. Создана технология реконструкции кодограмм, содержащих кодовые значения ДНП чисел. Для этого выполняются такие этапы:

1) декомпозиция служебной и информационной частей кодовой конструкции сжатого представления текущего фрагмента базового кадра с последующей их разметкой на кодовые конструкции отдельных диагоналей;

2) декомпозиция информационной части кодовой конструкции путем разметки фиксированного числа кодограмм компактного представления диагоналей трансформанты.

Научная новизна результатов исследований.

1. Впервые создана технология реконструкции кодограмм, содержащих кодовые значения ДНП чисел на основе декомпозиции служебной и информационной частей кодовой конструкции сжатого представления текущего фрагмента базового кадра. В отличие от других технологий проводится разметка кодового потока базового кадра на кодовые конструкции трансформант с последующей их разметкой на кодовые конструкции отдельных диагоналей с использованием информации о значениях оснований и фиксированного количества кодограмм. Это позволяет без потерь информации установить кодограммы по назначению содержащейся в них информации.

2. Впервые построена обобщенная технология прямого и обратного кодовых преобразований диагоналей без учета априорной информации о ее длине и порядковом номере в трансформанте. В отличие от существующих подходов учитывается симметрия структурных свойств трансформанты относительно наибольшей диагонали, и формирование для нее диагонально-неравномерных позиционных чисел. Это позволяет упростить алгоритмические реализации процессов кодирования и декодирования.

**Список литературы:** 1. *Олифер В.Г.* Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. / В.Г. Олифер, Н.А. Олифер. СПб.: Питер, 2006. 958 с. 2. *Gonzales R. C.* Digital image processing / R.C. Gonzales, R.E. Woods. - Prentice Inc. Upper Saddle River, New Jersey 2002. 779 p. 3. *Lee S.Y.* Temporally coherent video matting / S.Y. Lee, J.C. Yoon, I.K. Lee // Graphical Models 72. 2010. P. 25 – 33. 4. *Красильников Н.Н.* Цифровая обработка изображений. М.: Вузовская книга, 2011. 320 с. 5. *Баранник В.В.* Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков. Х.: ХУПС, 2010. 234 с. 6. *Barannik V.* Method Of Encoding Transformant Uolsha Is In Systems Air Monitoring Of Earth / V. Barannik, A. Yakovenko, A. Krasnorutkiy // Lviv-Slavsko, Ukraine, Lviv Polytechnic National University, International Conference TCSET'2009, Modern problems of radio engineering, telecommunications and computer science, February 19 – 23, 2009. P. 381-383.

*Поступила в редколлегию 16.03.2014*

**Баранник Владимир Викторович**, д-р техн. наук, профессор, начальник кафедры боевого применения и эксплуатации АСУ Харьковского университета Воздушных Сил. Научные интересы: обработка и передача информации. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79.

**Отман Шади О.Ю.**, аспирант кафедры Сети Связи ХНУРЭ. Научные интересы: обработка и передача информации. Адрес: Украина, 61023, Харьков, пр. Ленина, 14.

**Хаханова Анна Владимировна**, канд. техн. наук, доц. кафедры АПВТ ХНУРЭ. Научные интересы: сжатие и восстановление двоичных данных. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: anna\_hahan@mail.ru

## **ОБОСНОВАНИЕ ИСПОЛЬЗОВАНИЯ КОМПЛЕКСА ДИАГНОСТИЧЕСКИХ МЕТОДИК ДЛЯ ОЦЕНКИ ПСИХОФИЗИОЛОГИЧЕСКОГО СОСТОЯНИЯ ЧЕЛОВЕКА**

---

Приводится математическое обоснование выбора нескольких методов для психофизиологической диагностики организма человека. Проводится анализ и отбор диагностических методик для исследования психофизиологического состояния человека. Дается краткая характеристика выбранных методик и описываются количественные критерии, позволяющие характеризовать с различных сторон текущее функциональное состояние центральной нервной системы человека: функциональный уровень системы, устойчивость реакции и уровень функциональных возможностей.

### **1. Введение**

Поведение человека в сложных экстремальных ситуациях определяется его психологическим состоянием и готовностью к принятию решения и адекватным действиям. Для снижения возможности проявления ошибочных действий человека необходимо организовать обучение, тренировки, развивающие быстроту мышления, подсказывающие, как использовать прежний опыт для успешного принятия решения, для перевода действий работника на уровень стереотипов, а также формирующие способность к прогнозированию. Кроме этого, нужно проводить профессиональный отбор, а также соответствие психофизиологических возможностей человека условиям труда. Профессиональный психологический отбор работников любой сферы деятельности ставит задачу выявить людей, у которых процесс обучения дает максимальный эффект при минимальном времени обучения. Профессиональная пригодность определяется положительной мотивацией к данной специальности; высоким порогом ощущения опасности; быстротой реакции на экстремальные ситуации; хорошим глазомером; устойчивостью, концентрацией и распределением внимания; нормальным состоянием двигательного аппарата; высокой пропускной способностью анализаторов[1].

Поэтому процессы профессионального становления и управления профессиональной деятельностью персонала требуют специальных знаний и умений в области выявления психофизиологических механизмов оптимального обеспечения трудовых функций. Это делает возможным своевременно определить степень профессиональной пригодности человека к конкретному виду деятельности посредством всестороннего изучения его личности: изучения уровня соответствия физических качеств и психофизиологических особенностей индивида к профессиональным требованиям. В настоящее время для изучения и контроля психофизиологических особенностей индивида все больше и больше привлекают технические средства, в частности ЭВМ. Но все разработанные технические средства, как правило, предполагают выдачу заключения о психофизиологическом состоянии (ПФС) человека на основании тестирования по одной методике[2].

*Целью* работы является теоретическое обоснование применения комплексной методики диагностических исследований психофизиологических показателей, повышающей достоверность результирующих заключений о состоянии человека.

### **2. Обоснование использования комплекса диагностических методик для оценки психофизиологического состояния человека**

Дадим формализацию понятия ПФС организма человека. Рассмотрим различные психофизиологические состояния организма, которые имеют свои нозологические единицы («образы»). Каждое из состояний может быть описано многими способами, принадлежащими к разным областям человеческих знаний [3]. Конечное множество состояний обозначим  $A = \{A_1, A_2, A_3, \dots, A_n\}$ .

Предположим, что для определения некоторого психофизиологического состояния  $A_i$  ( $i=1,2,3,\dots,n$ ) используются  $m$  методов диагностики. Пусть  $D = \{D_1, D_2, D_3, \dots, D_m\}$  - объединение этих методов в множество. Существует  $k$  диапазонов величин, используемых для измерений каждого из состояния  $A_i$  ( $i = 1, 2, 3, \dots, n$ ). Все имеющиеся диапазоны объединим в множество  $V = \{B_1, B_2, B_3, \dots, B_k\}$ .

Следовательно, каждое из ПФС организма  $A_i$  ( $i = 1, 2, 3, \dots, n$ ) можно охарактеризовать с помощью различных методов диагностики  $D_j$  ( $j = 1, 2, 3, \dots, m$ ), каждый из которых использует определенные диапазоны параметров  $B_l$  ( $l = 1, 2, 3, \dots, k$ ).

Так как множество методов диагностики  $D = \{D_1, D_2, D_3, \dots, D_m\}$  образует множество несовместных событий, то исходя из теории вероятностей, его можно представить в следующем виде:

$$\sum_{i=1}^k p(D_i) = 1, D_i \cdot D_j = 0, i \neq j; i, j = 1, 2, 3, \dots, m. \quad (1)$$

Психофизиологическое состояние  $A_i$ , получаемое с помощью конкретного метода диагностики  $D_j$  с использованием определенных диапазонов параметров  $B_l$  ( $l = 1, 2, 3, \dots, k$ ), обозначим:

$$A_i = A_i \cdot D_j \cdot B_1 + A_i \cdot D_j \cdot B_2 + \dots + A_i \cdot D_j \cdot B_k = \sum_{l=1}^k A_i \cdot D_j \cdot B_l. \quad (2)$$

Пусть  $a_j$  – количественная составляющая оценки психофизиологического состояния  $A_i$ , полученная при диагностике методом  $D_j$ , а  $d_j = 1$ , если используется при диагностике метод  $D_j$ , и  $d_j = 0$ , если при диагностике не используется метод  $D_j$ . Тогда психофизиологическое состояние  $a_i^*$ , получаемое с помощью всех применяемых методов диагностики  $D_j$  ( $j=1,2,3,\dots,m$ ) с использованием определенных параметров  $B_l$  ( $l = 1, 2, 3, \dots, k$ ), обозначим:

$$a_i^* = \left( \sum_{j=1}^m \sum_{l=1}^k a_i d_j B_l \right) / \sum_{l=1}^k B_l. \quad (3)$$

Для определенного метода диагностики  $D_j$  множество параметров  $B_l$  ( $l = 1, 2, 3, \dots, k$ ) также является полной системой несовместимых событий:

$$\sum_{i=1}^k p(B_i) = 1, B_i \cdot B_j = 0, i \neq j; i, j = 1, 2, 3, \dots, k. \quad (4)$$

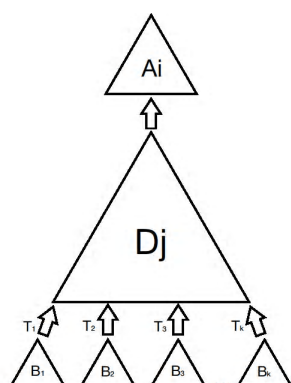
Определим вероятность диагностики психофизиологического состояния  $A_i$  исходя из того, что выбор методов диагностики равновероятен и выбор параметров при каждом из них также равновероятен. Получаем условную вероятность, т.е. вероятность обнаружения образа состояния при конкретной диагностике, используя равновероятность выбора метода диагностики, теоремы сложения и умножения вероятностей и формулу полной вероятности:

$$\begin{aligned} \rho(A_i) &= \rho(A_i \cdot D_j \cdot B_1 + A_i \cdot D_j \cdot B_2 + \dots + A_i \cdot D_j \cdot B_k) = \rho(A_i \cdot D_j) \cdot \rho\left(\sum_{l=1}^k B_l\right) = \\ &= \rho(A_i \cdot D_j) = \rho(D_j) \cdot \rho(A_i/D_j) = \rho(A_i/D_j) \end{aligned} \quad (5)$$

Вероятность диагностики психофизиологического состояния человека с помощью используемых методик и диапазонов параметров (среднее арифметическое условных вероятностей) будет:

$$\begin{aligned} \rho(A_i) &= \rho(A_i \cdot D_1 \cdot B_1 + A_i \cdot D_1 \cdot B_2 + \dots + A_i \cdot D_1 \cdot B_k + A_i \cdot D_2 \cdot B_1 + A_i \cdot D_2 \cdot B_2 + \dots + \\ &+ A_i \cdot D_2 \cdot B_k + \dots + A_i \cdot D_m \cdot B_1 + A_i \cdot D_m \cdot B_2 + \dots + A_i \cdot D_m \cdot B_k) = \\ &= \rho(A_i \cdot D_1 + A_i \cdot D_2 + \dots + A_i \cdot D_m) = \sum_{j=1}^m \rho(D_j) \cdot \rho(A_i/D_j) = \frac{\rho(A_i/D_j)}{m} \end{aligned} \quad (6)$$

Описание психофизиологического состояния человека может не совпадать с состоянием, получаемым с помощью одного метода диагностики, что приводит к ошибкам в диагностике состояния. Используя несколько методов диагностики психофизиологического состояния человека, можно избежать этих ошибок.



Общая схема проведения диагностики психофизиологического состояния организма человека

Процесс принятия решения (диагностики психофизиологического состояния человека) может быть представлен логической схемой (рисунок), на которой определено время выполнения измерений и состояния человека.

Следовательно, для точной оценки ПФС человека необходимо использовать несколько методов диагностики. Совпадение результатов работы нескольких диагностических средств, функционирующих в одном программно-аппаратном комплексе, увеличивает достоверность получаемых результатов по оценке состояния.

### 3. Выбор методик

Выбор методик, которые были использованы для обследования, обусловлен диагностическими возможностями методов, их безопасностью и комфортностью операторов, а также малым временем, требуемым на проведение измерений. Для оценки функционального состояния центральной нервной системы (ЦНС) были подобраны следующие методики.

*Методика “Простая зрительно-моторная реакция”.* Простая зрительно-моторная реакция (ПЗМР) – это элементарный вид произвольной реакции человека на зрительный стимул. Она состоит из двух последовательных компонентов: сенсорного (латентного) периода и моторного периода.

Простая зрительно-моторная реакция лежит в основе других целенаправленных приспособительных реакций человека, поэтому на основании показателя скорости ПЗМР можно сделать вывод о временных параметрах более сложных составляющих поведения человека. Кроме того, скорость простой зрительно-моторной реакции позволяет оценить интегральные характеристики ЦНС человека, так как при ее реализации задействованы как основные анализаторные системы человека (зрительная и кинестетическая), так и определенные отделы головного мозга и нисходящие нервные пути[4].

Оценка результатов по методике “Простая зрительно-моторная реакция” при наличии нормального распределения производится на основании среднего значения времени реакции и стандартного отклонения. Среднее значение отражает среднюю скорость ПЗМР, характерную для данного индивида: чем меньше среднее значение времени реакции, тем выше скорость реагирования. Стандартное отклонение является показателем стабильности сенсомоторного реагирования: чем меньше стандартное отклонение, тем более стабильной является скорость сенсомоторной реакции.

Для получения наиболее полной информации о свойствах и состоянии центральной нервной системы на основании результатов по данной методике можно использовать дополнительные показатели, в частности критерии Т.Д. Лоскутовой и коэффициент точности Уиппла.

Для получения такой информации используются три количественных критерия, позволяющие характеризовать с различных сторон текущее функциональное состояние центральной нервной системы: ее функциональный уровень, устойчивость реакции и уровень функциональных возможностей[5].

Первый критерий – функциональный уровень системы (ФУС). Его величина определяется, главным образом, абсолютным значением времени реакции и отражает текущее функциональное состояние ЦНС, степень развития утомления под влиянием факторов окружающей среды.

Второй критерий – устойчивость реакции (УР). Величина этого показателя тем больше, чем меньше вариабельность значений времени простой двигательной реакции. Поскольку разнообразие значений времени реакций является проявлениями непрерывных флуктуаций состояния ЦНС, показатель УР рассматривается как критерий устойчивости состояния

ЦНС. Следовательно, чем выше показатель УР, тем устойчивее, стабильнее текущее функциональное состояние ЦНС.

Третий критерий – уровень функциональных возможностей (УФВ). Он наиболее полно характеризует состояние ЦНС и позволяет судить о ее способности сформировать и достаточно долго удерживать соответствующее функциональное состояние.

Оценка функционального уровня системы вычисляется по формулам:

$$\PhiУС = \ln 1 / T_{\text{мод}} \cdot \Delta T_{0,5} [c^2]. \quad (7)$$

Оценка устойчивости реакции:

$$УР = \ln P_{\text{макс}} / \Delta T_{0,5} [c^{-1}]. \quad (8)$$

Оценка уровня функциональных возможностей:

$$УФВ = \ln P_{\text{макс}} / \Delta T_{0,5} \cdot T_{0,5} [c^{-2}], \quad (9)$$

где  $T_{\text{мод}}$  – время реакции, соответствующее пику гистограммы (наиболее вероятному значению);  $P_{\text{макс}}$  – вероятность (частота) появления значения, равного моде (амплитуда моды);  $\Delta T_{0,5}$  – диапазон времени реакций на уровне  $0,5P_{\text{макс}}$ .

На основании имеющегося экспериментального материала для аппаратной реализации метода получены [6] следующие граничные значения критериев функционального состояния ЦНС для здоровых испытуемых (таблица):

функциональный уровень системы = 4,2-5,5;

устойчивость реакций = 1,0-2,8;

уровень функциональных возможностей = 2,7-4,8.

Состояние системы	ФУС	УР	УФВ
Высокий уровень	4,9-5,5	2,1-2,8	3,9-4,8
Средний уровень	4,5-4,8	1,5-2,0	3,1-3,8
Низкий уровень	4,2-4,4	1,0-1,4	2,7-3,0

Это исследование очень важно для интегральной оценки здоровья человека, особенно спортсменов и лиц экстремальных видов деятельности, подвергающихся воздействию чрезмерных и физических, и психоэмоциональных нагрузок. Для включения результатов теста зрительно-моторной реакции в итоговый протокол комплексного обследования вычисленные значения критериев пересчитываются в процентах.

Коэффициент точности Уиппла выявляет соотношение ошибок и правильных реакций и вычисляется по формуле:

$$КТ = \frac{N - R}{N + P}, \quad (10)$$

где  $N$  – число измерений (предъявленных сигналов);  $R$  – количество правильных нажатий;  $P$  – количество ошибок.

Чем меньше данный показатель, тем выше степень точности выполнения заданий. Показатель точности свидетельствует о степени свойства устойчивости внимания, обусловленного в свою очередь силой и уравновешенностью нервных процессов.

*Методика “Реакция на движущийся объект”.* Реакция на движущийся объект представляет собой разновидность сложной сенсомоторной реакции, т.е. такой реакции, которая помимо сенсорного и моторного периодов включает период относительно сложной обработки сенсорного сигнала центральной нервной системой.

*Методика “Реакция различения”.* Реакция различения является разновидностью сложной сенсомоторной реакции. В отличие от простой реакции, реакция различения осуществляется на один определенный стимул из нескольких разнообразных стимулов. Поэтому процесс обработки сенсорной информации центральной нервной системой происходит не только по принципу наличия либо отсутствия сигнала, но и по принципу различения сигналов, отбора сигналов определенного цвета из общего их числа и формирования реакции на заданный вид сигнала. В связи с более сложным процессом обработки сенсорной информации центральной нервной системой скорость реакции различения меньше, чем скорость

простой реакции, т.е. время, затраченное на осуществление реакции различения, больше, чем на осуществление простой реакции [6].

#### 4. Заключение

Обоснован формализованный подход к оценке психофизиологического состояния человека с использованием вероятностных методов, в частности векторного анализа, что позволяет снизить уровень ошибок разрабатываемой системы диагностики за счет увеличения количества методов диагностики. В качестве основных критериев, влияющих на выбор методики тестирования, предложены следующие: неинвазивность, портативность и возможность автономной работы, время и место обследования, возможность проведения обследования тренером, инструктором или врачом, а также представление результатов в доступном виде для специалиста различной квалификации.

Используя комплекс методик, можно решать широкий спектр научных и практических задач в области общего и профессионального образования, военной службы, физкультуры и спорта, социального обслуживания.

**Список литературы:** 1. *Копанев В.И.* Коррекция функционального состояния организма летчиков-инструкторов авиационных училищ в период интенсивных полетов / В.И. Копанев, В.А. Егоров // Военно-медицинский журнал. 1988. № 10. С. 54–56. 2. *Иорданская Ф.А.* Мониторинг здоровья и функциональная подготовленность высококвалифицированных спортсменов в процессе учебно-тренировочной работы и соревновательной деятельности / Ф.А. Иорданская, М.С. Юдинцева. М.: Советский спорт, 2006. 184 с. 3. *Пиеничный Б.Н., Данилин Ю.М.* Численные методы в экстремальных задачах. М.: Наука, 1975. 319 с. 4. *Короленко Ц.П.* Психофизиология человека в экстремальных условиях / Ц.П. Короленко. Л.: Медицина, 1978. 272 с. 5. *Одеришев Б.С.* Методики измерения функционального состояния человека / Б.С. Одеришев // Экспериментальная и прикладная психология. Психические состояния / Под ред. А.А.Крылова. Л., 1981. С. 30–38. 6. *Tuckow A.P.* Nocturnal growth hormone secretory dynamics are altered after resistance exercise: deconvolution analysis of 12-hour immunofunctional and immunoreactive isoforms / A.P. Tuckow // American Journal of Physiology. Regulatory, Integrative and Comparative Physiology. 2006. Vol. 291, N 6. P. 1749–1755.

*Поступила в редколлегию 22.03.2014*

**Боброва Наталья Леонидовна**, старший преподаватель кафедры МПСС Белорусского государственного университета информатики и радиоэлектроники. Научные интересы: нейронные сети, нечеткая логика, объектно-ориентированные языки программирования, проектирование ПО. Адрес: Беларусь, Минск, ул. Долгобродская, 3-88, тел.: +37529-109-23-93, e-mail: natasha.bobrowa@gmail.com

**ПОДСИСТЕМА КОНТРОЛЯ ЗНАНИЙ ПО ТЕМЕ  
«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ DELPHI. РАБОТА  
С ЗАПИСЯМИ»**

---

Описываются результаты разработки программного и информационного обеспечения подсистемы контроля знаний по теме «Программирование на языке Delphi. Работа с записями». Приводятся результаты тестирования подсистемы контроля знаний, которые показывают, что разработанный программный продукт работает в соответствии с техническим заданием. Показываются примеры работы программы, демонстрирующие корректность ее функционирования. Эта подсистема может быть использована в учебном процессе для оперативного контроля знаний студентов как автономно, так и в составе более полной системы. Проводится экспериментальная проверка эффективности использования этой разработки. Проверка подтвердила правильность гипотезы исследования.

**1. Введение.** В последние годы методы автоматизированного контроля знаний студентов получают все большее распространение. Такой метод контроля знаний не может претендовать на объективную и всеобъемлющую оценку знаний студентов. Он никогда не заменит такие формы контроля, при которых выявляется творческое восприятие знаний, способность студентов самостоятельно мыслить и формулировать ответы на поставленные вопросы, умение применять полученные знания на практике (семинары, обсуждения, практические занятия, лабораторные работы, зачеты, экзамены).

Тестирование может применяться в учебном процессе для оперативного контроля студентов, «грубой» оценки усвоения полученных знаний и даже как косвенный способ закрепления пройденного материала. Таким образом, метод контроля знаний с помощью тестирования является весьма актуальным. Нет сомнений, что и в дальнейшем он будет получать все большее распространение (совместно с другими формами контроля).

Создание систем тестирования естественным образом распадается на две основные задачи: 1) разработка соответствующего программного обеспечения; 2) разработка набора вопросов и вариантов ответов к ним (информационного обеспечения). От преподавателя зависит, будет ли тест содержать достаточное количество вопросов, насколько «правдоподобно» сформулированы варианты ответов.

Работа посвящена решению двух задач – разработке программного и информационного обеспечения подсистемы контроля знаний по теме «Программирование на языке Delphi. Работа с записями».

**2. Основное содержание.** Как уже отмечалось, подсистема контроля знаний состоит из двух составляющих – программного и информационного обеспечения. К каждой из этих составляющих могут быть предъявлены определенные требования [1]. Прежде всего, программное обеспечение подсистемы контроля знаний должно исключить возможность, когда студент может просто «механически» запомнить, какие номера вариантов ответов являются правильными. Чтобы исключить такую возможность, подсистема контроля знаний при каждом новом запуске теста должна предлагать варианты ответов к вопросам в случайном порядке. Еще лучше будет, если и вопросы будут каждый раз появляться в случайном порядке. В этом случае учащемуся придется запоминать и содержание вопроса, и содержание правильного ответа к нему, что уже само по себе способствует содержательному усвоению материала.

Программное обеспечение должно давать студенту возможность возвращаться к уже пройденным вопросам.

Следующим требованием к программному обеспечению подсистемы контроля знаний является проверка того, что студент полностью ответил на все вопросы теста. Иначе возможно получение положительной оценки, в то время как отсутствие ответов на какие-либо вопросы свидетельствует о наличии у учащегося пробелов в знаниях по заданной теме.

Весьма важным требованием к программному обеспечению является возможность преподавателю по окончании тестирования просмотреть, какие именно варианты ответов выбирал студент. Во-первых, это позволяет выявить типичные ошибки, которые связаны со сложностью материала или с недостатками в преподавании. Во-вторых, результаты тестирования полезно обсудить со студентами, чтобы помочь им разобраться в своих ошибках и лучше усвоить материал.

Основным требованием к информационному обеспечению является наличие достаточного количества вопросов по заданной теме. Это необходимо для того, чтобы снизить вероятность случайного получения положительной оценки и повысить точность и объективность результатов тестирования. Можно утверждать, что минимальное количество вопросов в тесте должно быть в два раза больше, чем максимальный балл, по которому оцениваются знания (например, при пятибалльной системе в тесте должно быть как минимум десять вопросов, при 12-балльной системе – 24 вопроса).

По этой же причине может быть выдвинуто требование к количеству вариантов ответов. Оптимальное количество вариантов ответов на каждый вопрос составляет от 4 до 6. Меньшее количество вариантов повышает вероятность случайного угадывания правильного ответа, большее – затрудняет студенту правильное осмысление содержания вариантов ответов. Обязательным требованием к ошибочным вариантам ответов является их «правдоподобность». Ответы должны быть сформулированы таким образом, чтобы студент не мог чисто логическими умозаключениями отбросить явно неподходящие варианты. Они должны быть «похожими» на правильный ответ.

Вполне понятным требованием является полнота теста, т.е. вопросы должны по возможности охватывать весь материал по заданной теме.

Таким образом, в результате проведенного анализа можно сформулировать следующие общие требования к подсистеме контроля знаний, которые необходимо учитывать при ее разработке [2] :

– требования к программному обеспечению:

1) программа должна обеспечивать появление вариантов ответа на каждый вопрос в случайном порядке. Вопросы и варианты ответов к ним должны появляться в случайном порядке;

2) давать возможность студенту возвращаться к уже пройденным вопросам и при необходимости изменять выбор варианта ответа;

3) обеспечивать требование ответа на все вопросы теста;

4) программа должна давать возможность преподавателю после окончания тестирования просмотреть, какие варианты ответов выбирал студент;

– требования к информационному обеспечению:

1) тест должен содержать достаточное количество вопросов (как минимум в два раза больше, чем максимальный балл);

2) вопросы теста должны наиболее полно охватывать всю тему;

3) для каждого вопроса должно быть не менее четырех вариантов ответа;

4) варианты ответов должны выглядеть «правдоподобными».

Проведенный анализ показал, что возможны, как минимум, два подхода к организации данных. При первом подходе данные теста (вопросы, варианты ответов, информация о правильных ответах) хранятся в исходном коде программы. Основное преимущество здесь состоит в том, что эти данные полностью защищены от внешнего «изучения», в результате которого можно было бы получить информацию о правильных ответах. Студент работает только с одним исполняемым файлом (.exe) без каких-либо дополнительных данных. Главный недостаток такого подхода заключается в том, что для изменения имеющегося теста или разработки нового необходимо иметь исходный текст программы и уметь вносить в него необходимые изменения.

Второй подход предусматривает хранение данных теста во внешних файлах. Основное преимущество такого подхода состоит в универсальности программного обеспечения. Один раз откомпилированная программа может использоваться как основа для любого теста. Разрабатывается и изменяется только информационное обеспечение, которое может готовиться с помощью обычных текстовых редакторов. Однако в этой универсально-

сти заключается и основной недостаток. Если внешние данные оставить в исходном виде, то они могут быть легко прочитаны и легко может быть получена информация о правильных ответах. Чтобы этого избежать, данные необходимо «шифровать» (кодировать), а это вызывает необходимость дополнительного программного обеспечения.

В данной работе выбраны следующие параметры программного и информационного обеспечения:

- данные теста хранятся в исходном коде программы;
- тест рассчитан на 10 вопросов;
- к каждому вопросу предлагается 4 варианта ответа, только один из которых является правильным;
- программа рассчитана на оценку по 5-балльной системе;
- программа всегда предьявляет варианты ответов в случайном порядке (вопросы следуют в одном и том же порядке);
- программа предусматривает переходы от одного к вопросу к другому в любом направлении (от первого к последнему или наоборот) с возможностью изменений выбора варианта ответа.

Разработка программного обеспечения осуществлена в среде Delphi.

В качестве диалогового окна подсистемы контроля знаний была разработана следующая форма (рис. 1).

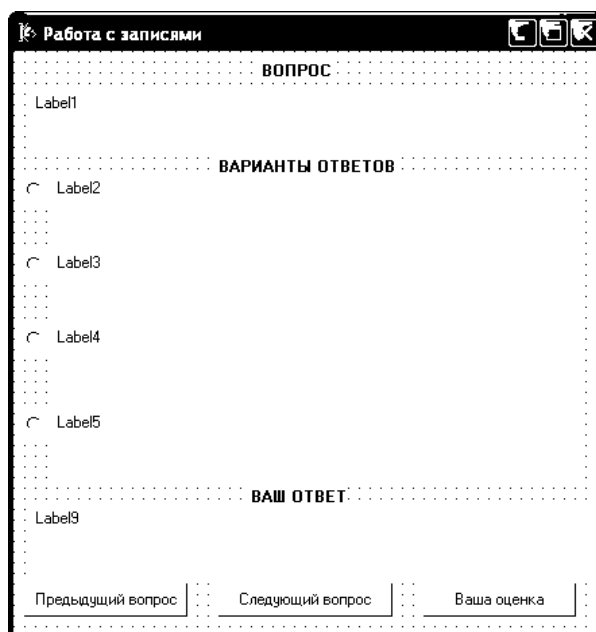


Рис. 1. Диалоговое окно подсистемы контроля знаний

Назначение элементов управления следующее. Метка Label1 используется для отображения текущего вопроса. Метки Label2 – Label5 содержат четыре варианта ответа на текущий вопрос. Метки Label6 – Label8 служат для отображения заголовков частей окна (надписи «Вопрос», «Варианты ответов» и «Ваш ответ»). В метке Label9 будет отображаться выбранный учащимся вариант ответа. Элементы управления RadioButton1 – RadioButton4 позволяют студенту указать выбранный вариант ответа. При разработке программы были выбрана следующая структура данных. Прежде всего, были определены следующие глобальные переменные:

1) строковый массив `Vopros`, содержащий вопросы теста и варианты ответов на каждый вопрос (ниже приведен фрагмент описания массива):

```
var  
  Vopros:array[1..50] of string=  
  (' Что такое «запись» в языке Delphi? ');
```

‘Это структура данных, которая состоит из отдельных именованных’ +  
’ компонент различного типа’,

‘Это синоним типа символьной строки’,

‘Это структура данных, содержащая набор (массив) символьных строк’,

‘Это специальная структура, в которую необходимо помещать данные’ +  
‘ перед их записью в файл’,

‘Что такое поля записи?’

‘Это именованные компоненты различного типа, из которых состоит запись’,

‘Это особые дополнительные данные, которые описывают структуру записи’,

Этот массив построен следующим образом. Первый элемент массива содержит первый вопрос. В следующих четырех элементах (со 2-го по 5-й) содержатся варианты ответа на этот вопрос. Далее идет второй вопрос, а в четырех следующих за ним элементах содержатся варианты ответа к нему и т. д. Таким образом, весь массив разбивается на группы по пять элементов, в первом из которых содержится вопрос, в четырех следующих за ним элементах – варианты ответа к нему;

2) строковый массив Prav, содержащий правильные ответы:

Prav: array [1..10] of string =

‘Это структура данных, которая состоит из отдельных именованных’ +  
’ компонент различного типа’,

‘Это именованные компоненты различного типа, из которых состоит запись’,

‘В разделе type’,

‘record, end’,

3) строковый массив Otvet, в который в процессе работы будут заноситься выбранные учащимся варианты ответов:

Otvet: array[1..10] of string=(‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’);

4) целочисленная переменная NV, которая в процессе работы будет содержать номер текущего вопроса:

NV: integer;

5) также был объявлен новый тип, описывающий целочисленный массив из четырех чисел:

type

AR4 = array [1..4] of integer;

Исходя из основных требований к программному обеспечению, были разработаны процедуры: Rand4 (var RA:AR4), ShowVopros.

В работе выполнено программирование событий FormCreate, Button1Click, Button2Click, Button3Click.

Выполнена также разработка информационного обеспечения. Приведен полный список вопросов теста и варианты ответов к ним. В результате проведенного исследования разработано программное и информационное обеспечение в соответствии с выдвинутыми ранее требованиями.

Разработанная Delphi-программа имеет простой и понятный интерфейс. Вся работа с программой осуществляется только при помощи мыши.

После запуска программы на экране появляется диалоговое окно программы с первым вопросом теста. Варианты ответов приведены в случайном порядке (т.е. при других запусках программы порядок ответов будет другим) (рис.2).

В заголовке окна отображается название теста. Ниже – номер текущего вопроса. Правильный (по мнению учащегося) вариант ответа указывается путем нажатия левой клавишей мыши в белом кружочке, расположенном возле выбранного ответа. Выбранный вариант отображается в нижней части окна (поле «Ваш ответ») (рис. 3).

В нижней части окна находятся две кнопки с надписями «Предыдущий вопрос» и «Следующий вопрос». С помощью этих кнопок студент может перейти к следующему вопросу или вернуться к предыдущим (например, чтобы изменить выбор варианта ответа).

Когда все вопросы пройдены, появляется следующее сообщение (рис. 4).

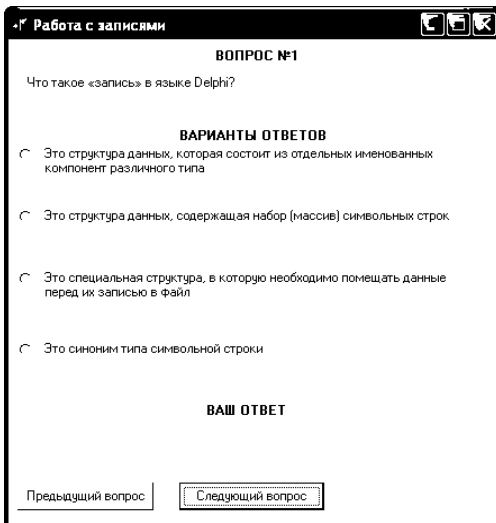


Рис. 2. Отображение первого вопроса теста после запуска программы

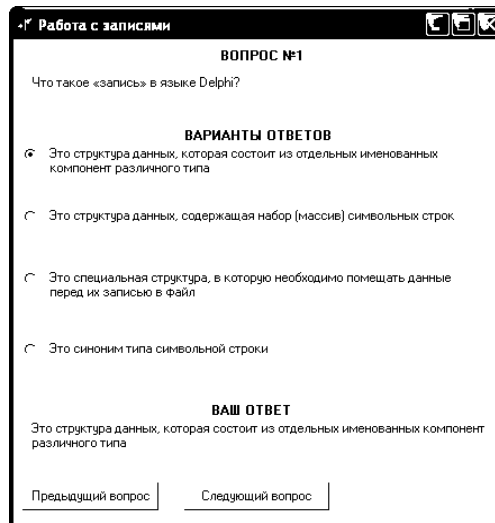


Рис. 3. Выбор варианта ответа

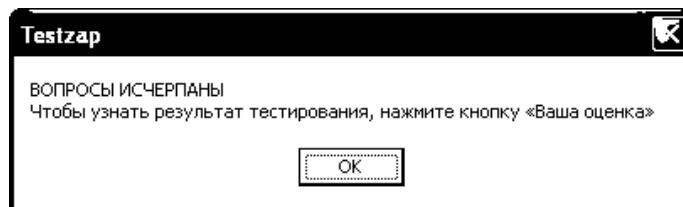


Рис. 4. Сообщение о том, что все вопросы исчерпаны

В нижней части окна появляется кнопка «Ваша оценка».

До того, как нажать на эту кнопку для получения результата тестирования (т.е. оценки), студент может еще раз просмотреть свой выбор вариантов ответов по всем вопросам и, при необходимости, изменить его. После нажатия на кнопку «Ваша оценка» уже ничего нельзя будет изменить.

Если на какой-либо вопрос не был выбран вариант ответа (вопрос был пропущен), об этом будет выдано соответствующее сообщение и вопрос будет вновь предъявлен для ответа (рис. 5).

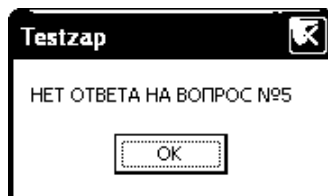


Рис. 5. Сообщение о том, что на определенный вопрос не был дан ответ

Когда на все вопросы будут даны ответы, будут выданы результаты тестирования, которые должны быть предъявлены преподавателю (до нажатия кнопки ОК) (рис. 6). После нажатия кнопки ОК происходит окончательный выход из программы (окно закрывается).

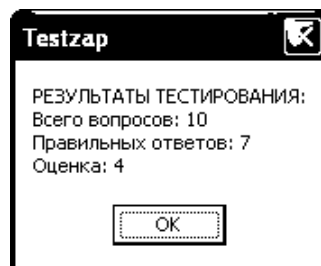


Рис. 6. Результаты тестирования

Подробнее результаты тестирования можно просмотреть в текстовом файле ResTest.txt, который создается в той же папке, из которой был произведен запуск программы. В нем с помощью любого текстового редактора (например, Блокнота) можно просмотреть все вопросы и выбранные варианты ответов, а также увидеть, какие из них были правильными, а какие – ошибочными (рис. 7).

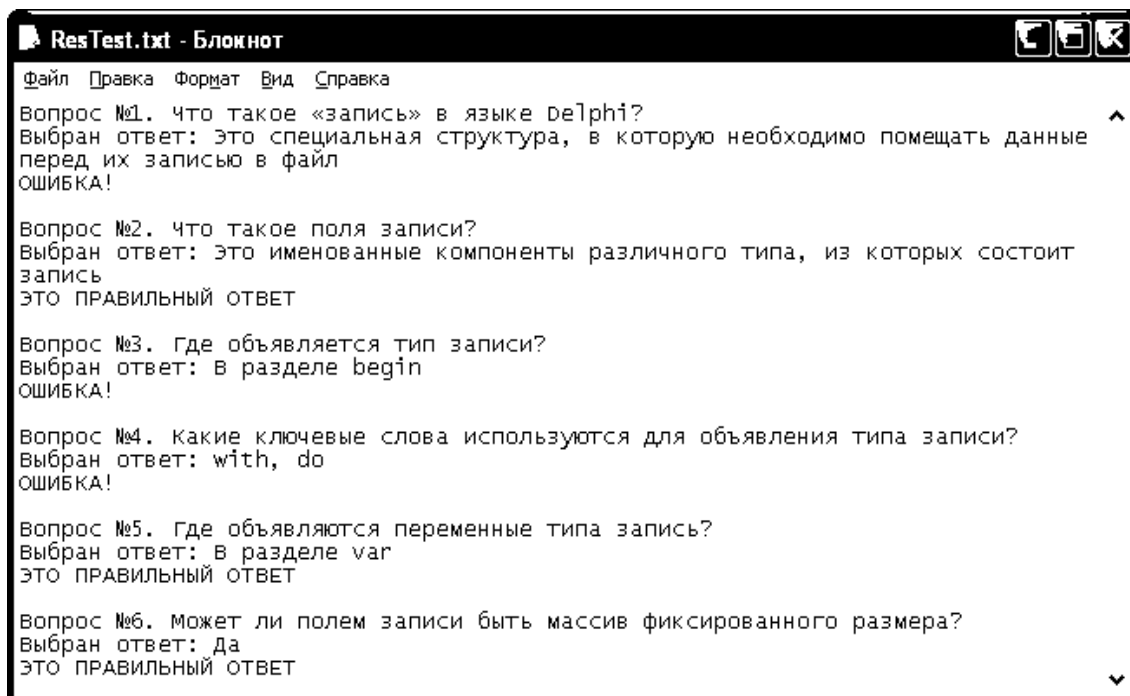


Рис. 7. Просмотр результатов тестирования в текстовом файле (фрагмент)

*Научная новизна.* Разработана и протестирована подсистема контроля знаний по теме «Программирование на языке Delphi. Работа с записями». Подсистема удовлетворяет основным требованиям, предъявляемым к программному и информационному обеспечению для таких систем.

В результате проделанной работы был проведен анализ общих требований, выдвигаемых к подсистемам контроля знаний, разработано соответствующее программное и информационное обеспечение для подсистемы контроля знаний по теме «Программирование на языке Delphi. Работа с записями». Показаны примеры работы программы, демонстрирующие корректность ее функционирования.

Разработанная подсистема может быть использована в учебном процессе для оперативного контроля знаний студентов как автономно, так и в составе более полной системы.

**Список литературы:** 1. Бондаренко М.А. Інформаційні технології та програмування у середовищі Delphi. Х.: ФЛП Александрова К.М., 2014. 536 с. 2. Бондаренко М.А. Програмування у середовищі Delphi. Х.: «Бізнес Інформ», 2010. 704 с.

Поступила в редколлегию 12.03.2014

**Бондаренко Николай Андреевич**, канд. техн. наук, профессор, Украинская инженерно-педагогическая Академия. Научные интересы: информационные технологии и проектирование технических систем. Адрес: Украина, 61000, Харьков, ул. Клочковская, 195 г, кв. 44, тел. 7 19 50 01.

## **МЕТОДЫ АНАЛИЗА И ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ (АНАЛИТИЧЕСКИЙ ОБЗОР)**

Рассматриваются основные направления технологий проектирования, валидации и обеспечения качества (quality assurance) систем на кристаллах и в пакетах кристаллов.

### **Введение**

Цель – аналитический обзор квантовых методов вычислений, технологий тестирования и диагностирования цифровых систем на кристаллах при их проектировании и верификации, ориентированный на повышение быстродействия программных и аппаратных средств анализа цифровых устройств, а также существенное увеличение выхода годной продукции и уменьшение времени ее выхода на рынок микроэлектроники.

Задачи: 1) Особенности классических и квантовых вычислений. 2) Построение моделей цифровых систем. 3) Моделирование исправного поведения цифровой систем. 4) Моделирование неисправностей цифровой системы. 5) Генерация тестов. 6) Методы диагностирования неисправностей. 7) Построение кубитных моделей цифровых систем.

Источники: классические и квантовые вычисления [1-20]; квантовые вентили [6, 15, 27]; виды неисправностей цифровых схем [28-32, 36, 40-44]; генерация тестов [36, 37, 45]; моделирование цифровых систем [28-35, 38, 39]; технологии тестирования и диагностирования систем на кристаллах [28-34, 36-39].

### **1. Классический и квантовый бит информации**

Классический и квантовый компьютеры оперируют числами. Двоичная система счисления классического компьютера основывается на использовании двух чисел: 0 и 1. Количество информации, хранящееся в одной двоичной ячейке, равно 1 биту. В двух двоичных ячейках хранится 2 бита информации, которые могут составлять следующие комбинации: 00, 01, 10, 11. В регистре, состоящем из  $n$  двоичных ячеек, хранится  $n$  бит информации, количество комбинаций возможных значений которых равно  $2^n$ .

В квантовом компьютере [1-20] информация также может быть представлена в виде двоичных чисел. Однако в ячейке, называемой квантовым битом (кубитом), может храниться не одна из двух цифр двоичного счисления (0 или 1), а одновременно обе эти цифры [1,2]. Например, в двух кубитах могут храниться одновременно 4 двоичных числа 00, 01, 10, 11. В  $n$  кубитах могут храниться одновременно  $2^n$  двоичных чисел длины  $n$ . Кубиты могут быть связаны друг с другом (запутаны): изменение одного из них приводит к изменениям других. Совокупность запутанных кубитов представляет собой заполненный квантовый регистр, который может не только находиться во всевозможных комбинациях составляющих его битов, но и реализовывать всевозможные зависимости между ними [2]. В процессе вычислений на квантовом компьютере одновременно обрабатываются все биты квантового регистра (имеет место квантовый параллелизм).

В процессе вычислений квантовые компьютеры выполняют логические операции над квантовыми состояниями путём унитарных преобразований, не нарушающих квантовые суперпозиции, в соответствии с алгоритмом [3]:

- 1) запись (подготовка, инициализация) начального состояния;
- 2) вычисление (унитарные преобразования начальных состояний);
- 3) вывод результата (измерение, проецирование конечного состояния).

Классический компьютер оперирует битами – булевыми переменными, принимающими значения 0 и 1 – и на любом этапе вычислений в каждом бите хранятся определенные значения, используемые для вычислений. На первом этапе вычислений в каждый бит записываются исходные данные – определенные значения (0 или 1).

Квантовый компьютер оперирует состояниями квантового регистра. Данные в процессе вычислений представляют собой квантовую информацию, которая по окончании вычислительного процесса преобразуется в классическую путём измерения конечного состояния квантового регистра. Выигрыш по быстродействию в квантовых алгоритмах достигается за счёт того, что при применении одной квантовой операции большое количество коэффициентов суперпозиции квантовых состояний, которые в виртуальной форме содержат классическую информацию, преобразуется одновременно (квантовый параллелизм).

## 2 Квантовые вентили

2.1. *Простые квантовые вентили.* По аналогии с обычным электронным цифровым компьютером вентиль квантовой вычислительной системы может обрабатывать один или несколько кубитов [6, 15]. Квантовая система имеет входные и выходные состояния, определяемые (задаваемые) одним или несколькими кубитами. Система преобразует их в соответствии с аксиомами квантовой механики. Состояния квантовой системы преобразуются с помощью унитарных, обратимых (инвертирующихся) операторов. Следовательно, квантовые вентили должны иметь одинаковое количество входных и выходных кубитов.

Простой квантовый вентиль  $A$ , преобразующий один входной кубит в один выходной, представлен на рис. 1.



Рис. 1. Квантовый вентиль

Здесь предполагается, что в квантовом вентиле поток информации направлен слева направо, поэтому стрелки на рис. 1 не показаны. Данное упрощение графического представления квантового вентиля стало возможным благодаря тому факту, что квантовые вентили преобразуют кубиты с помощью унитарных обратимых операторов. В этом случае графическое представление логических вентилях, преобразующих классические биты в электронном цифровом компьютере, не является удобным для квантовых вентилях.

На рис. 1  $|\psi\rangle, |\chi\rangle \in C^2$  – кубиты;  $A: C^2 \rightarrow C^2$  – унитарный линейный обратимый оператор. Удобно использовать матричное представление квантовых вентилях:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

где для кубитов  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, |\chi\rangle = \gamma|0\rangle + \delta|1\rangle$ , для которых  $A|\psi\rangle = |\chi\rangle$ , справедливо:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}. \quad (1)$$

Пример квантового вентиля NOT, задаваемого унитарной матрицей:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Нетрудно заметить, что в этом случае выражение (1) преобразуется к виду:

$$X(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle.$$

Другими словами, вентиль NOT представляет собой переключатель между состояниями  $|0\rangle$  и  $|1\rangle$ .

Другой пример квантовых вентилях, описываемых унитарными матрицами:

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

которые действуют на данный кубит в соответствии с выражениями:

$$Y(\alpha|0\rangle + \beta|1\rangle) = -i(-\beta|0\rangle + \alpha|1\rangle),$$

$$Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle.$$

X, Y и Z называют матрицами Паули.

Вентиль Адамара осуществляет самообратимую операцию формирования суперпозиции состояний и определяется унитарной матрицей [6, 15]:

$$H = \left(\frac{1}{\sqrt{2}}\right) \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Справедливо следующее выражение

$$X^2 = Y^2 = Z^2 = H^2 = I,$$

которое означает, что каждый вентиль X, Y, Z и H определяется с помощью корня квадратного единичной матрицы и соответствующего квантового вентиля I.

2.2. Многокубитные квантовые вентили. В многокубитном вентиле должно быть одинаковое количество кубитов на входе и выходе [6].

Двухкубитные вентили соответствуют операциям поворота в гильбертовом пространстве двух взаимодействующих кубитов, которые не могут быть представлены в виде прямого произведения независимых однокубитовых операций [15].

Основным двухкубитовым вентилем является обратимый контролируемый инвертор или оператор «контролируемое НЕ» (CNOT) с двумя входными и двумя выходными кубитами (рис. 2), который функционирует в соответствии с выражениями:

$$\begin{aligned} |00\rangle &\mapsto |00\rangle, & |01\rangle &\mapsto |01\rangle, \\ |10\rangle &\mapsto |11\rangle, & |11\rangle &\mapsto |10\rangle. \end{aligned}$$

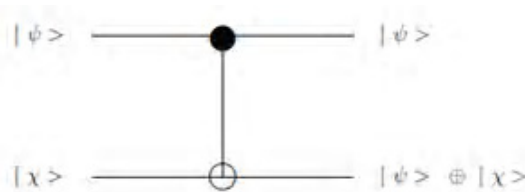


Рис. 2. Контролируемое НЕ

Когда  $|\psi\rangle = |0\rangle$ , то  $|\psi\rangle \oplus |\chi\rangle = |\chi\rangle$ ; для  $|\psi\rangle = |1\rangle$  справедливо  $|\psi\rangle \oplus |\chi\rangle = X|\chi\rangle$ .

Вентиль CNOT описывается матрицей:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \delta \\ \gamma \end{pmatrix},$$

где  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ ,  $|\chi\rangle = \gamma|0\rangle + \delta|1\rangle$ .

Кубит  $|\psi\rangle$  является контролирующим, кубит  $|\chi\rangle$  – контролируемым, над которым производится операция NOT при условии, что первый кубит находится в состоянии  $|1\rangle$ .

Двухкубитовый оператор обмена состояниями кубитов SWAP может быть реализован путем последовательного выполнения трех операций CNOT (рис. 3) [15] и описывается матрицей:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

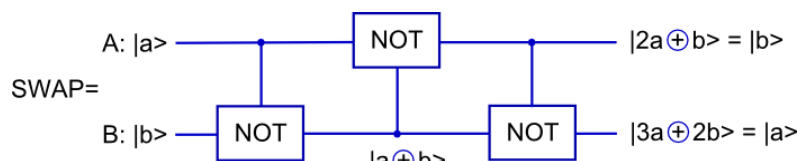


Рис. 3. Оператор SWAP

Трехкубитовый вентиль Тоффоли (CCNOT) представлен на рис. 4 и содержит два управляющих кубита А и В, а также один управляемый С.

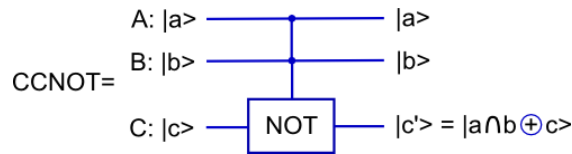


Рис. 4. Вентиль Тоффоли

Вентиль Тоффоли описывается матрицей 8x8 в базисных состояниях  $|0,0,0\rangle, |0,0,1\rangle, |0,1,0\rangle, |1,0,0\rangle, |0,1,1\rangle, |1,0,1\rangle, |1,1,0\rangle, |1,1,1\rangle$ :

$$CCNOT = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Данная операция может быть также реализована в виде пяти двухкубитовых операций. N-битный обобщенный вентиль Тоффоли описывается как  $(k_1, k_2, \dots, k_n) \rightarrow (k_1, k_2, \dots, (k_1, k_2, \dots, k_{n-1}) \oplus k_n)$  [20]. Вентиль NOT является частным случаем вентиль Тоффоли, для которого  $n=1$ , вентиль CNOT – частным случаем, когда  $n=2$ .

Расширенный  $n+1$ -битный вентиль Тоффоли (extended Toffoli gate, ETG) с двумя управляемыми линиями  $(o_n, o_{n+1})$  показан на рис. 5. ETG имеет входной вектор  $(k_1, k_2, \dots, k_n, k_{n+1})$  и выходной вектор  $(o_1, o_2, \dots, o_n, o_{n+1})$ , где  $o_j = k_j$  для  $j = \overline{1, (n-1)}$ ,  $o_n = k_1, k_2, \dots, k_{n-1} \oplus k_n$ ,  $o_{n+1} = k_1, k_2, \dots, k_{n-1} \oplus k_{n+1}$ . Первые  $n-1$  битов являются управляющими, последние два бита – управляемыми.

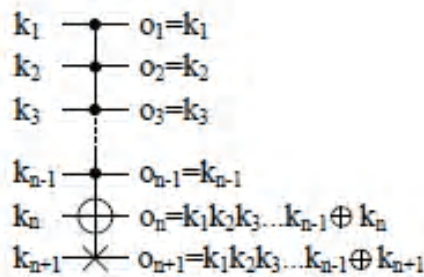


Рис. 5.  $n+1$ -битный вентиль Тоффоли

### 3. Виды дефектов

Подходы к моделированию неисправностей, диагностированию, тестированию и обеспечению отказоустойчивости цифровых схем имеют важное значение для создания надежных изделий [40]. С развитием VLSI технологии существенно увеличилось количество компонентов, размещаемых на одном кристалле, и, как результат – возросла степень интеграции и количество неисправностей.

В процессе проектирования цифрового изделия необходимо периодически проверять (путем тестирования) корректность функционирования схемы и отсутствие неисправностей, а также обеспечить корректную работу схемы в случае появления неисправностей (отказоустойчивость) [36].

Под неисправностью схемы понимают физический дефект одного или более компонентов [28, 29, 32]. Различают постоянные и нерегулярные неисправности. Постоянные (жесткие) неисправности могут быть следствием разрушения или износа компонента. Нерегулярные (мягкие) дефекты проявляются в определенные промежутки времени и могут быть кратковременными (transient) и перемежающимися (intermittent) [40-44].

Неисправности могут быть логическими и параметрическими. Логическая неисправность изменяет булеву функцию, которая реализуется схемой. Параметрическая – изменяет значение параметра схемы (ток, напряжение). К параметрическим дефектам относят неисправности задержки, связанные с различным временем прохождения сигнала через логические вентили, что приводит к гонкам (согласованиям) сигналов [46].

Моделирование большого количества физических дефектов может быть основано на использовании одной модели логической неисправности, что позволяет существенно уменьшить сложность моделирования. Модель логической неисправности не зависит от технологии имплементации проекта, а тесты, разработанные для обнаружения логической неисправности, могут применяться также и для выявления физических дефектов.

Модель логической неисправности может быть явной или неявной. Явная модель неисправности определяет пространство неисправностей, в котором каждая неисправность может быть идентифицирована, а неисправности, подлежащие анализу, могут быть явно описаны. Явная модель неисправности практически применима, если ее размерность не является слишком большой. Неявная модель описывает пространство неисправностей путем совокупной идентификации неисправностей определенного типа их характеристическими признаками. Моделирование неисправностей тесно связано с моделированием системы. Неисправности, определяемые в сочетании со структурной моделью, относятся к структурным неисправностям, проявляющимся в изменении структуры межсоединений компонентов. Функциональные неисправности определяются в сочетании с функциональной моделью. Например, функциональная неисправность может проявляться в изменении таблицы истинности компонента или искажении RTL операции.

Классы неисправностей. Существует три класса логических неисправностей: константная (stuck-at-fault), мостиковая (bridging fault) и неисправность задержки (delay fault).

Наиболее распространенная модель константной неисправности – одиночная константная неисправность. Сущность модели заключается в том, что неисправность логического вентиля приводит к «залипанию» логического 0 (константа 0, s-a-0) или логической 1 (константа 1, s-a-1) на одном из его входов или выходов [42, 44].

Модель константной неисправности также используется для представления кратных неисправностей в схеме. При этом предполагается, что более, чем на одной линии схемы имеется константная неисправность s-a-0 или s-a-1. Другими словами, совокупность константных неисправностей существует в схеме в одно и то же время. Разновидностью кратной неисправности является однонаправленная неисправность – когда все конститuentы (составляющие части) неисправности представляют собой s-a-0 или s-a-1, но не обе одновременно.

Модель константной неисправности не является эффективной при моделировании СБИС (Very Large Scale Integrated, VLSI), построенных по CMOS технологии. Неисправности в CMOS схемах не обязательно представляют собой логические дефекты, которые могут быть описаны моделью константной неисправности. Дефекты CMOS-схем отображаются также моделями устойчивых обрывов транзисторов SOP (stuck-open) и устойчивых замыканий транзисторов SON (stuck-on) [42].

Мостиковые неисправности типа «короткое замыкание» представляют собой постоянные дефекты, которые не могут быть смоделированы константной неисправностью. Короткое замыкание возникает, когда две или более сигнальных линий схемы электрически связаны друг с другом. Мостиковые неисправности вентильного уровня классифицируются следующим образом: входные – вызваны коротким замыканием входов логического элемента, неисправности типа обратной связи – вызваны замыканием входной и выходной линий, а также неисправности без обратной связи, которые не относятся к первым двум типам. В теории моделирования мостиковых неисправностей делается предположение, что вероятность замыкания более двух линий является низкой и логика межсоединений реали-

зуется в виде связей. Мостиковая неисправность в положительной логике возникает в том случае, когда ее поведение описывается проводным AND (0 является доминантным логическим значением), и в отрицательной логике - когда ее поведение описывается проводным OR (1 является доминантным логическим значением).

Неисправности задержки [41, 43]. Небольшое количество дефектов, которые могут вызвать разрывы и короткие замыкания в схеме, имеют достаточно высокую вероятность появления из-за наличия отклонений параметров производственного процесса. Дефекты могут также приводить к нарушениям временных параметров схемы без изменения логики ее работы: задержка переключения сигнала из 0 в 1, и наоборот. Существует два вида неисправностей задержки: неисправность задержки вентиля и неисправность задержки пути. Задержка вентиля используется для моделирования дефектов, при которых время прохождения сигнала через вентиль превышает предельно-допустимое. Данная модель может быть использована только для изолированных, не транспортируемых дефектов, например, несколько малых задержек. Модель задержки пути может быть использована как для изолированных, так и для транспортируемых дефектов. При этом предполагается, что неисправность проявляется в случае, если задержка распространения сигнала вдоль линии схемы превышает допустимое значение.

Кратковременные и перемежающиеся неисправности рассматриваются как временные дефекты. Основная часть неисправностей цифровых схем вызвана именно временными дефектами, которые характеризуются сложностью выявления и устранения. Кратковременные дефекты являются неповторяющимися и вызываются, как правило, флуктуациями напряжения питания или воздействием радиационного излучения. Они являются основной причиной отказа элементов памяти систем на кристаллах.

Перемежающиеся неисправности могут появляться в результате нарушения межсоединений, применения дефектных компонентов, воздействия внешних факторов (температура, влажность, вибрация) или быть следствием ошибок проектирования. Перемежающиеся неисправности возникают случайным образом и моделируются с помощью вероятностных методов (Марковские модели).

#### **4. Генерация тестов**

Непрерывное совершенствование технологий проектирования и производства цифровых изделий приводит к увеличению плотности компоновки и сложности устройств, достижение необходимого уровня надежности которых обеспечивается тестированием. Для решения задачи тестирования современных сверхсложных электронных устройств необходимы новые, более эффективные методы построения тестов [47-52]. Производство систем на кристаллах (system-on-chip, SoC) с использованием технологии глубокого субмикрона (Deep Submicron, DSM) позволяет снизить затраты на производство, но при этом стоимость тестирования остается неизменной и представляет собой значительную часть общей стоимости проекта. Уменьшить затраты на тестирование изделия можно путем повторного использования блоков интеллектуальной собственности (IP cores), а также разработки моделей и методов тестирования SoC на высоком уровне иерархии [53]. Высокоуровневые модули системы на кристалле описываются в терминах поведения функциональных компонентов, что не позволяет использовать для их тестирования готовые технические решения вентильного уровня. Классическая модель одиночной константной неисправности (stuck-at fault), представляющая внутренние логические вентили или их межсоединения, не применима для использования на системном уровне. Структурное высокоуровневое тестирование не может быть выполнено с использованием готовых тестовых решений, поскольку генерация теста выполняется после структурного синтеза. Реализация тестирования зависит от технологии изготовления SoC и изменяется в процессе жизненного цикла изделия [36].

Для обеспечения возможности многократного использования тестов в новых проектах необходимо разработать такую модель неисправности, которая является независимой от реализации системы на кристалле. Следует также найти ответы на вопросы: 1) Может ли тест, построенный на базе функциональной модели неисправности, быть эффективно использован для непокрываемых тестом физических дефектов? 2) Как эффективность теста зависит от синтезируемой структуры? Эти вопросы являются важными не только с точки зрения повторного использования тестов, но также из-за того, что программные модули

могут быть синтезированы с помощью существующих систем автоматизированного проектирования и сохранены в библиотеках IP модулей.

Для успешного проектирования и производства изделия необходимы методология тестирования и модели неисправностей, которые обеспечивают высокоуровневую валидацию проекта [36].

Дефект-ориентированное тестирование, основанное на генерации тестов на транзисторном уровне и использовании токовых моделей (IDDQ), эффективно применяется в технологии глубокого субмикрона. IDDQ метод тестирования основан на измерении тока и хорошо работает, когда средний ток схемы с неисправностью больше, чем ток исправного устройства [37]. Дефект-ориентированное тестирование начинается после реализации этапа размещения и трассировки (place and route) (рис. 6) [36].

Сущностью константной модели неисправностей является абстракция реального дефекта. Модель является основой для автоматической генерации тестовых наборов и формирования алгоритмов моделирования неисправностей. Условием достижения высокого уровня покрытия неисправностей является то, что тест должен транспортировать некоторое конкретное значение (значения) в дефектную область от управляемых точек ввода и далее до наблюдаемых выходов в целях обнаружения неисправного поведения. Данная модель наиболее эффективна для тестирования на кристалле (post-silicon testing). Генерация тестов для константных неисправностей выполняется перед размещением и трассировкой проекта (рис. 6). Тесты для константных неисправностей покрывают реальные дефекты топологии только примитивных вентилях. В этом случае говорят о генерации тестов, не зависящих от топологии кристалла.

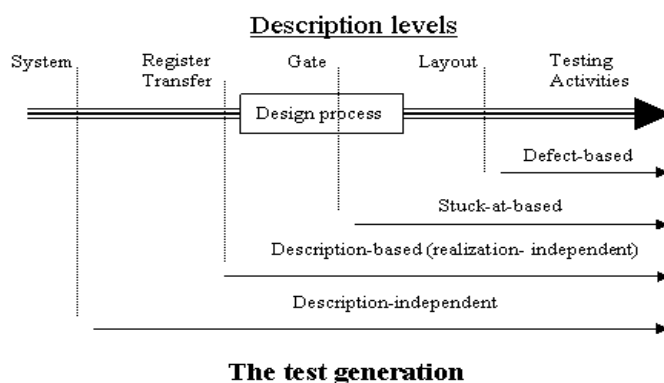


Рис. 6. Генерация тестов в процессе проектирования

Известны несколько подходов к генерации тестовых наборов на уровне регистровых передач (Register Transfer Level, RTL) [36]: использование двоичных деревьев решений, выявление искажений в RTL описании схемы, объединение статического анализа с симуляцией. Большинство из них позволяют генерировать тестовые последовательности удовлетворительного качества, совместимые со средствами ATPG вентилярного уровня. Главным преимуществом тестирования устройства на RTL уровне является то, что размерность описания схемы здесь намного меньше, чем на логическом уровне. При генерации тестов на уровне регистровых передач набор тестовых последовательностей создается для всех возможных реализаций, и можно говорить о генерации тестов, не зависящих от имплементации. Задача генерации тестов может выполняться параллельно с синтезом схемы на вентилярном уровне (см. верхнюю часть рис. 7).

Генерация тестов на системном уровне зависит от используемой модели неисправностей. В этом случае не только реализация, но и синтезируемое поведенческое описание не известны. Задача генерации тестов в этом случае является более сложной, но может решаться одновременно с формированием синтезируемого описания и синтезом схемы на вентилярном уровне (см. рис. 7) [36].

Перед отправкой проекта в производство необходимо сформировать тестовые наборы. Верификация устройства предполагает запуск тестовой программы на рабочей модели кристалла. Тестеры являются дорогостоящими компонентами и могут использоваться в течение длительного времени. Начало разработки тестов в конце процесса проектирования значительно увеличивает время выхода изделия на рынок. Если используется методология проектирования сверху вниз, то системная модель изделия на кристалле формируется в самом начале процесса проектирования и может быть использована при разработке тестовой программы. Таким образом, инженер-тестировщик может стать участником процесса проектирования на ранних стадиях и использовать виртуальный прототип устройства в виде системной модели. Это позволит существенно уменьшить время проектирования и стоимость изделия.

Методология проектирования сверху вниз ориентирована на автоматический синтез списка соединений вентильного уровня с использованием поведенческого описания или системной модели. Время выхода изделия на рынок (time-to-market) зависит от длительности процедуры логического синтеза, продолжительности тестопригодного проектирования и генерации тестов (см. рис. 7). Тестопригодное проектирование и генерация тестов на основе системной модели позволит сократить время выхода на рынок [36, 45]. Анализ тестового покрытия на вентильном уровне не является времязатратной процедурой и дает возможность уменьшить длину тестовых последовательностей, полученных на системном уровне.

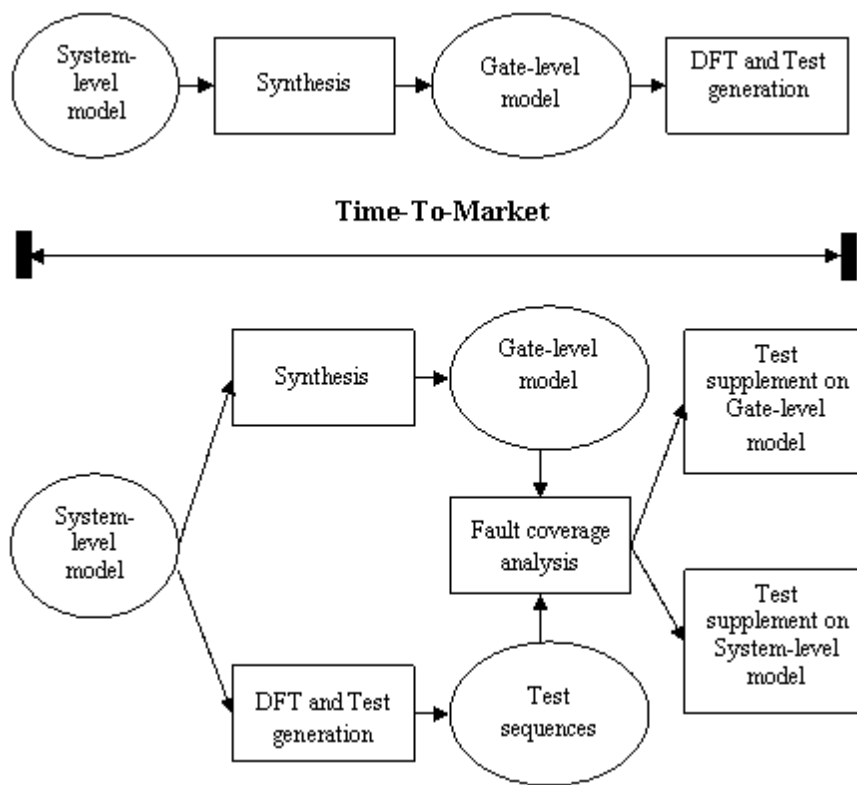


Рис. 7. Время time-to-market на различных этапах процесса проектирования

Генерация тестов на системном уровне не может гарантировать 100% покрытие неисправностей вентильного уровня для каждой возможной имплементации. Формирование тестовых последовательностей необходимо выполнять параллельно на системном и на вентильном уровнях, поскольку вероятность генерации тестов для труднообнаруживаемых неисправностей может быть различной на каждом уровне описания.

## 5. Модели функциональных неисправностей

Все более широкое использование программно-аппаратных систем в критических приложениях привело к повышению значимости верификации и тестирования программных и аппаратных модулей. В настоящее время существует ряд проблем верификации, связанных с высокой сложностью программно-аппаратных приложений и их гетерогенной структурой [35, 48-53]. Стоимость верификации системы увеличилась до такой степени, что иногда даже превышает стоимость проекта. Формальные методы верификации позволяют проверить функциональность с помощью формальных методов (проверка моделей, проверка эквивалентности, автоматическое доказательство теорем). Для управления сложностью задачи верификации предложены методы, основанные на симуляции (эмуляции) описания системы заданной входной последовательностью.

Функциональные неисправности искажают пространство состояний цифрового изделия, представленное спецификацией. Дефект проектирования представляет собой неправильную деталь проекта, сформированную разработчиком. Дефекты проектирования являются следствием либо синтаксических (семантических) ошибок в описании устройства, либо фундаментального непонимания функциональности, описанной проектной спецификацией. Количество потенциальных дефектов проектирования может быть слишком большим, чтобы с ними можно было бороться автоматически или вручную, поэтому необходимо применять способы уменьшения сложности проекта без ущерба для точности результатов. Модель проектной неисправности описывает поведение некоторого множества дефектов проектирования. Модель функциональной неисправности описывает физические и проектные дефекты аппаратных и программных модулей. Модель функциональной неисправности можно оценить точностью моделирования проектных дефектов и эффективностью (количеством обнаруживаемых неисправностей схемы).

Большинство аппаратных систем разрабатываются на основе методологии проектирования сверху вниз, которая начинается с поведенческого описания системы. Как результат, большинство моделей функциональных неисправностей являются моделями поведенческого или алгоритмического уровня. Существующие модели функциональных неисправностей могут быть классифицированы по стилю поведенческого описания, на котором они базируются. Системное поведение описывается на языках программирования (System C) или описания аппаратуры (VHDL, Verilog) и преобразуется во внутренний формат для использования в процессе симуляции.

5.1. *Текстовые (семантические) модели неисправностей.* Текстовая (семантическая) модель неисправностей применяется для исходного текстового поведенческого описания проекта [38]. Простейшей текстовой моделью является метрика покрытия инструкций (statement coverage metric), используемая при тестировании программного обеспечения, которая связывает потенциальную ошибку с каждой строкой кода и требует, чтобы каждый оператор поведенческого описания выполнялся во время тестирования [38]. Эта модель не очень эффективна, поскольку количество возможных неисправностей равно числу строк кода. Ограничение точности покрытия инструкций позволяет в сочетании с другими моделями неисправностей повысить эффективность тестирования.

Ряд моделей функциональных неисправностей базируются на обходе путей графа потоков управления (Control-Data Flow Graph, CDFG), описывающего поведение системы [38]. Ранние модели неисправностей CDFG основывались на покрытии ветвей и путей графа. Покрытие ветвей предполагает, что множество всех проверяемых путей графа CDFG охватывает два направления реализации всех бинарных условий. Покрытие ветвей широко используется при тестировании программного и аппаратного обеспечения, однако применение только данной модели не позволяет получить полную гарантию корректности кода.

Метрика покрытия путей является более эффективной по сравнению с метрикой покрытия ветвей, поскольку она отражает количество путей графа потоков управления. Предполагается, что дефект связан с некоторым путем графа потоков управления и, следовательно, для гарантированного обнаружения всех неисправностей должны быть выполнены все пути потока управления. Количество путей управления может быть бесконечным, если граф CDFG содержит цикл. Поэтому метрика обхода путей может быть ограничена длиной пути. Поскольку общее число путей потока управления растет экспоненциально с количе-

ством условных операторов, можно выбрать подмножество всех путей потока управления, необходимое и достаточное для тестирования. Одним из критериев выбора пути может быть базисный набор путей или подмножество путей, которые линейно-независимы и могут образовывать любой другой путь. При тестировании потоков данных появление каждой переменной рассматривается либо как описание переменной, либо как ее использование. При выборе пути рассматриваются такие, которые связывают определение переменной с ее применением. Критерии тестирования потоков также применяются для проверки поведенческого описания аппаратных модулей.

Большинство моделей неисправностей графа потоков управления рассматривают пути без ограничения значений переменных и сигналов. В противоположность им существуют модели неисправностей, ориентированные на переменные/сигналы, которые включают более жесткие ограничения на величину сигнала в целях обнаружения неисправностей. Методика анализа домена при тестировании программного обеспечения рассматривает не только путь потока управления, но и значения переменных и сигналов во время выполнения. Домен представляет собой подмножество пространства входных элементов программы, в котором каждый элемент активизирует выполнение программы по некоторому пути. Неисправность домена вызывает выполнение программы, следствием которого является переход в неправильную область. Данный метод может быть также применен для тестирования аппаратного обеспечения.

Многие модели неисправностей графа потоков управления содержат требования к активизации неисправностей независимо от значения наблюдаемости. Для устранения этого недостатка предложены поведенческие модели неисправностей, применимые для тестирования software и hardware модулей. Подход OCCOM [38] базируется на добавлении неисправностей, называемых тегами, к каждому определению переменной, представляющих положительное или отрицательное смещение от правильного значения сигнала. Знак ошибки известен, но величина – нет. Анализ наблюдаемости вдоль пути потока управления делается вероятностно, с использованием алгебраических свойств операций и данных моделирования. Тег будет распространяться с помощью поведенческой операции, если будут выполнены два условия: 1) совпадет знак; 2) другие входы в процессе выполнения операции не контролируются. Разработан также точный метод определения наблюдаемости, в котором константная неисправность вводится на внутренних переменных, и ее распространение обеспечивается поведением объекта. Поскольку анализ наблюдаемости является точным, вычислительная сложность при этом возрастает.

5.2. *Мутационные модели неисправностей.* Мутационное тестирование основано на искусственном внесении неисправностей в код программы и применяется для тестирования программного и аппаратного обеспечений [38]. Основная идея заключается в имитации типичных ошибок программиста и создании специальных тестов для их выявления (тестов, которые бы обнаруживали неисправности, если бы они присутствовали). Неисправности вводятся в оригинальную программу и создается много неисправных версий программы. Каждая из них содержит одну ошибку. Ошибочные программы называются мутантами оригинальной программы. Целью генерации тестов является разграничение оригинальной программы и всех ее мутантов. Оригинальная программа и все ее мутации тестируются на одном и том же наборе тестов. Если на этом наборе подтверждается правильность программы и выявляются все ошибки в программах-мутантах, то оригинальная программа объявляется правильной. Если в некоторых мутантах не были выявлены все ошибки, то тестовый набор считают не полным, и он подлежит доработке. Набор мутационных операторов языка VHDL включает изменения следующих объектов: арифметические операторы, определение и изменение абсолютного значения и константы, логические операторы, реляционные операторы, добавление унарного оператора. Каждый оператор представляет определенный класс неисправностей. Все возможные изменения в программе не могут рассматриваться из-за их непомерно большого количества. Изменения могут быть ограничены до приемлемого набора на основе двух гипотез: эффект сцепления и компетентный программист. Эффект сцепления гласит, что сложные неисправности могут сочетаться с простыми неисправностями, таким образом, тестовый набор, который обнаруживает все простые ошибки в программе, будет обнаруживать также и сложные неисправности.

Гипотеза «компетентный программист» утверждает, что компетентный программист стремится писать программы, которые практически являются правильными. Другими словами, программа, написанная компетентным программистом, может быть неправильной, но она будет отличаться от правильной версии относительно простыми ошибками. Недостатком мутационных моделей является локальный характер мутаций, что ограничивает применение моделей для описания большого набора дефектов проектирования.

5.3. *Модели неисправностей конечного автомата.* Конечные автоматы (КА) являются классическим способом описания поведения последовательностных схем, и для них определены модели неисправностей [38]. Наиболее распространенной является модель покрытия состояний, основанная на требовании покрытия всех возможных состояний и выполнения всех возможных переходов в процессе тестирования. Дефекты КА не выводят его за пространство состояний, заданное спецификацией. Проблема использования моделей неисправностей конечного автомата заключается в высокой сложности решения задачи тестирования, которая обусловлена большой размерностью пространства состояний типовой вычислительной системы. Решением указанной проблемы может быть выявление подмножества состояний конечного автомата, имеющих решающее значение для его корректного функционирования. Модели расширенного конечного автомата (Extended Finite State Machine, EFSM) и машины контроля (Extracted Control Flow Machine, ECFM) позволяют уменьшить конечный автомат путем его разделения на пространство состояний и пространство данных. Уменьшенный конечный автомат генерируется путем проецирования оригинального конечного автомата на множество состояний, имеющих наибольшее значение для процесса валидации.

## **6. Модель исправного поведения объекта**

Модель объекта диагностирования – это совокупность гетерогенных компонентов, взаимосвязанных во времени и пространстве, с заданной адекватностью описывающих некоторый процесс или явление. Модель может быть представлена в аналитической, табличной, векторной, графической или другой форме и задана в явном или неявном виде [28, 29, 32].

Явная модель объекта диагностирования состоит из описаний его исправной и всех неисправных модификаций. Неявная модель содержит описание исправного объекта, модели его физических неисправностей и правила получения по ним все неисправных модификаций объекта. Универсальной математической моделью объекта диагностирования является таблица функций неисправностей (ТФН). Каждое неисправное состояние объекта диагностирования соответствует одной неисправности (одиночной или кратной) из заданного класса неисправностей. Недостатком ТФН являются ее большие размеры. Модель дискретной системы может быть представлена в виде таблицы истинности, логической сети, альтернативного графа, эквивалентной нормальной формы представления булевых функций, таблицы переходов-выходов многотактной схемы. Выбор модели влияет на глубину и трудоемкость процесса диагностирования.

## **7. Логическое моделирование**

Логическое моделирование является формой верификационного тестирования проекта с использованием модели проектируемой системы. На вход модели подаются входные стимулы, выполняется построение и анализ временных диаграмм для внешних входов, выходов схемы и внутренних линий [35].

Верификация проекта позволяет проверить функциональные режимы относительно спецификации. Проверка осуществляется на каждой стадии преобразования модели от системного уровня до уровня имплементации проекта в кристалл путем сравнения результатов, полученных в процессе моделирования, и эталонных результатов, предусмотренных спецификацией.

Задачи, решаемые в процессе логического моделирования: 1. Проверка правильности функционирования цифровой схемы. 2. Исследование временных параметров схемы (быстродействие, время выполнения операций, тактовая частота). Обнаружение состязаний, рисков сбоя, анализ задержек. 3. Оптимизация проектных решений. 4. Генерация временных диаграмм. 5. Генерация тестовых последовательностей. 6. Моделирование неисправностей.

Для верификации проекта необходим прототип устройства, функционирующий на заданной рабочей частоте, однако создание прототипа является дорогостоящим и трудоемким процессом. Замена прототипа программной моделью называется симуляцией. Верификация проекта с использованием симулятора имеет следующие преимущества:

- проверка ошибочных условий (например, конфликты шин);
- возможность изменения задержек в модели для проверки граничных временных параметров;
- проверка задаваемых пользователем значений параметров схемы;
- возможность начала моделирования схемы на любом этапе проектирования;
- точный контроль тайминга асинхронных событий (например, прерываний);
- возможность формирования автоматизированного тестового окружения моделируемой схемы, использование RTL модели для управления и наблюдения за поведением схемы в процессе моделирования.

Технологии использования аппаратного прототипирования с помощью PLD имеют ряд существенных преимуществ по сравнению с программным симулятором: быстрое действие и возможность коррекции проекта.

Методы логического моделирования можно классифицировать по следующим признакам: 1) в зависимости от способа учета времени распространения сигнала – синхронные (без учета задержек в элементах схемы) и асинхронные (с учетом задержек); 2) в зависимости от способа представления сигналов – двоичные и многозначные (троичные, пятизначные); 3) по способу организации работы программы – компилятивные и интерпретативные; 4) в зависимости от организации очередности моделирования – пошаговые и событийные.

Синхронное моделирование предназначено для анализа переходных процессов в цифровых устройствах вентильного и функционального уровней описания на основе моделей элементов, представленных их логическими функциями без учета задержек сигналов. В процессе моделирования вычисляют значения сигналов на выходах логических элементов схемы по заданным входным сигналам. При этом предполагается, что время существования переходного процесса намного больше номинальной задержки схемы. Синхронное моделирование наиболее эффективно используется для анализа работы комбинационных схем в установившемся режиме. Результат моделирования в этом случае наиболее точно соответствует реальному режиму работы устройства. К методам синхронного моделирования относят:

- метод Эйхельбергера, предназначенный для синхронного анализа переходных процессов в цифровых устройствах вентильного уровня описания;
- многозначное синхронное моделирование, позволяющее обнаруживать все реальные состязания в схеме. Этот метод иногда указывает на ложные состязания, что приводит к дополнительным затратам при логической верификации цифровых систем.

Решением указанной выше проблемы являются асинхронные методы анализа цифровых схем. Их разнообразие определяется значностью алфавита моделирования и степенью адекватности моделей по реальным временным параметрам.

Асинхронное моделирование применяется для анализа переходных процессов в логических схемах с учетом времени распространения сигналов в элементах и соединительных цепях схемы. Каждый компонент схемы характеризуется некоторой средней задержкой, значение которой может изменяться в зависимости от режима работы компонента, комбинации входных сигналов, температуры, отклонений в технологии изготовления.

К методам асинхронного моделирования относят:

- $\Delta$ -троичное моделирование, которое устраняет недостатки двоичного асинхронного и троичного синхронного методов;
- асинхронное троичное моделирование с нарастающей неопределенностью, которое устраняет детерминизм в модельной задержке компонента схемы, заключая ее в некоторый интервал.

## 8. Моделирование неисправностей

Сущность моделирования неисправностей состоит в определении влияния одного или нескольких дефектов на состояния линий объекта при подаче тестовых последовательностей [39, 54]. Методы моделирования неисправностей можно классифицировать следующим образом: одиночное, параллельное, дедуктивное, кубическое и совместное моделирование.

Одиночное моделирование неисправностей базируется на внесении одной одиночной константной неисправности эквипотенциальной линии в схему. При подаче тестовых последовательностей выполняется анализ проявления неисправности на внешних выходах объекта диагностирования. Метод ориентирован на обработку схем нерегистрового уровня и не требует значительных временных затрат.

Параллельное моделирование неисправностей основывается на использовании машинных команд параллельной обработки слов (регистров): логическое сложение, умножение, инверсия, исключающее ИЛИ. Метод относится к компилятивному моделированию, поскольку поведение примитивов схемы описывается с помощью алгоритмических языков или ассемблеров. В процессе моделирования одновременно выполняется анализ  $P$  неисправностей на входном наборе, где  $P$  – разрядность машинного слова, доступного для параллельной обработки. К недостаткам метода относят сложность проектирования моделей и их ориентацию на конкретную вычислительную платформу. Быстродействие метода в  $P$  раз выше одиночного моделирования неисправностей. Идея параллельной обработки бинарного вектора с помощью только логических операций может быть использована для существенного увеличения скорости моделирования.

Дедуктивное моделирование неисправностей заключается в одновременной обработке всех одиночных константных неисправностей схемы на одном входном наборе и выделении при этом подмножества проверяемых дефектов. Метод ориентирован на вентильный уровень описания модели проектируемого объекта в базисе И-ИЛИ-НЕ. Необходимость получения аналитических формул для каждого типа примитивного элемента и большие затраты памяти для хранения списков неисправностей усложняют практическую реализацию метода.

В совместном (конкурентном) моделировании, как и в дедуктивном, выявляются сразу все проверяемые неисправности для данного входного набора. Метод ориентирован на обработку различных типов моделей схем, неисправностей, задержек и сигналов. В отличие от дедуктивного метода, где дефекты моделируются неявно, конкурентный алгоритм анализирует явно исправную работу и те неисправности, которые модифицируют состояния входов или выходов схемы, что используют эффективные модели элементов, такие как табличные и функциональные.

Дедуктивно-параллельное моделирование неисправностей цифровых систем основывается на применении преимуществ дедуктивного и параллельного алгоритмов [39] и позволяет за одну итерацию обработки моделируемой схемы выявить все неисправности, проверяемые на тест-векторе. Метод позволяет существенно повысить быстродействие моделирования одиночных константных неисправностей для оценки качества синтезируемых тестов цифровых систем, имплементируемых в ПЛИС, содержащих миллионы вентиляей.

## 9. Методы диагностирования неисправностей

Задачами технического диагностирования являются: определение технического состояния объекта, поиск места и определение причин отказа. Техническое состояние объекта определяется с помощью специальной тестовой последовательности входных воздействий. Методы формирования тестовых последовательностей для диагностирования неисправностей можно условно разделить на несколько типов.

1. Разделение, использование деревьев решений – заключается в моделировании поведения исправной системы и систем с  $N$  заранее определенными неисправностями. Отклик каждой из них на входное воздействие применяется для формирования  $(N+1)$  систем. При этом должны выполняться условия:

– исправная система должна быть быстро отделена от неисправных (выявление неисправностей);

– все системы являются однозначно идентифицируемыми (различимыми) (выявление неисправностей и определение их местоположения).

Результирующее разделение или построение дерева решений определяет диагностическую тестовую последовательность, которая позволяет однозначно определить принадлежность тестируемой системы одной из  $(N+1)$  категорий.

Для поиска неисправности применяют последовательный, комбинационный и последовательно-комбинационный методы. Последовательный метод заключается в таком построении процедуры поиска неисправностей, при котором информация о состоянии отдельных тестируемых систем вводится и логически обрабатывается последовательно. Реализация метода заключается в основном в определении очередности контроля. Программа поиска при этом может быть жесткой или гибкой. Жесткая программа предусматривает наличие заранее определенной последовательности контроля. При гибкой программе содержание и порядок последующих проверок зависят от предыдущих результатов. Комбинационный метод заключается в том, что на вход тестируемой системы подается фиксированный набор тестов. Диагноз формируется только после того, как будут получены отклики на все тестовые воздействия.

2. Активизация одномерного пути. Данный метод основывается на введении известной неисправности в схему и ее транспортировании на один из первичных выходов по активизированному пути. При этом любое изменение логического значения в месте неисправности приводит к изменению значения на соответствующем первичном выходе. Описанная процедура носит название прямой фазы. Обратная фаза заключается в определении значений других первичных входов и выходов, таких, чтобы заданная неисправность проявлялась на первичном выходе. Метод прост и удобен в использовании, однако в схеме могут существовать неисправности, для проверки которых необходимо активизировать несколько путей (в случае наличия сходящихся разветвлений).

3. Использование таблицы функций неисправностей и таблицы неисправностей. Таблица функций неисправностей является специальной формой представления поведения объекта диагностирования в исправном и неисправном состояниях. Таблица неисправностей связывает набор тестов и проверяемые ими неисправности. Ограничением данного метода является размерность указанных таблиц.

4. Метод булевых производных. Булева производная определяется путем выполнения операции OR над двумя булевыми функциями, представляющими исправный и неисправный объект. Если булева производная равна 1, считается, что проявляется ошибка и определяется соответствующая тестовая последовательность. Тестовые наборы определяются путем формирования булевой производной для каждой неисправности.

5. Метод эквивалентной нормальной формы основан на представлении булевой функции в виде эквивалентной нормальной формы, описывающей конкретную реализацию схемы. Эквивалентная нормальная форма может быть вычислена методом подстановки, с той разницей, что избыточные термы не исключаются, так как они характеризуют конкретную реализацию схемы.

Двоичное моделирование или эмуляция квантовых схем на классических компьютерах не отражает всех свойств квантовых схем. Эмуляция квантовых компонентов на классических компьютерах не дает существенных результатов по увеличению быстродействия [21-25]. В то же время моделирование цифровых классических схем на основе использования отдельных свойств кубита может обогатить «бедные» алгоритмы классических машин. Например, существует еще не решенная проблема, к которой можно подступиться: как за один такт промоделировать  $2^n$  в степени  $n$  входных набора, поданных на цифровую схему, если эти наборы представлены одним двоичным вектором-кубитом?

## **10. Заключение**

Таким образом, анализ современных рыночных технологий синтеза, анализа, тестирования, диагностирования и восстановления работоспособности цифровых систем на кристаллах, описанный в первом разделе, дает основания сформулировать актуальные на рынке электронных технологий тенденции в области Design and Test, сущность которых заключается в создании кубитных структур данных и «квантовых» методов моделирования, верификации, диагностирования и восстановления работоспособности, интегрированных в инф-

раструктуру сервисного обслуживания программно-аппаратных компонентов цифровых систем на кристаллах в целях повышения качества изделий или выхода годной продукции на основе уменьшения времени ремонта за счет увеличения аппаратной стоимости вычислительных проектов.

Основная инновационная идея квантовых или кубитных вычислений заключается в переходе от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно точечное решение, к квантовым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений. Функция цели есть повышение качества цифровых изделий за счет создания инфраструктуры, включающей технологии диагностирования и ремонта дефектных компонентов цифровой системы на кристалле  $L$ , минимизация времени восстановления работоспособности  $T$  за счет программно-аппаратной избыточности  $H$  инфраструктуры SoC:

$$E = F(L, T, H) = \min\left[\frac{1}{3}(L + T + H)\right],$$

$$Y = (1 - P)^n;$$

$$L = 1 - Y^{(1-k)} = 1 - (1 - P)^{n(1-k)};$$

$$T = \frac{(1 - k) \times H^s}{H^s + H^a}; H = \frac{H^a}{H^s + H^a},$$

где  $L$  – инверсная переменная выхода годной продукции  $Y$ , зависящая от тестопригодности проекта  $k$ , вероятности  $P$  существования дефектных блоков и числа необнаруженных неисправностей  $n$ . Время тестирования и диагностирования зависит от тестопригодности архитектуры  $k$ , умноженной на число примитивов, деленное на общее количество логических примитивов проекта. Нефункциональная избыточность для ремонта устройства зависит от сложности инфраструктуры, деленной на аппаратную сложность проекта.

**Список литературы:** 1. Дуплий В.А. Квантовая информация, кубиты и квантовые алгоритмы / С.А. Дуплий, В.В. Калашников, Е.А. Маслов // Вісник Харківського університету. 2005. №657. С. 99-104. 2. Нильсен М. Квантовые вычисления и квантовая информация / М. Нильсен, И. Чанг. Пер. с англ. М.: Мир 2006. 824 с. 3. Бекман И.Н. Лекции по информатике // Электронный ресурс <http://www.twirpx.com/file/602601/>. 4. Benenti G., Casati G., Strini G. Principles of Quantum Computation and Information. Volume 1: Basic Concepts. World Scientific. 2004. 256 p. 5. Vedral V., Plenio M.B. Basics of Quantum Computation. 1998. 28 p. Электронный ресурс: [arXiv:quant-ph/9802065](http://arXiv:quant-ph/9802065) v1 25 Feb 1998 <http://www.tfp.uni-karlsruhe.de/~cuevas/Lehre/SS04/9802065.pdf>. 6. Rosinger E.E. Basics of Quantum Computation ( Part 1 ). 2004. 87 p. Электронный ресурс: [arXiv:quant-ph/0407064](http://arXiv:quant-ph/0407064) v1 8 Jul 2004 <http://chaos.swarthmore.edu/courses/TSG/2004d.pdf>. 7. Stenholm S., Suominen K.-A. Quantum approach to informatics. A John Wiley & Sons, Inc., Publication. 2005. 249 p. 8. Imai Hiroshi, Hayashi Masahito. Quantum Computation and Information. From Theory to Experiment. Springer. 2006. 234 p. 9. Nielsen M.A., Chuang I.L. Quantum Computation and Quantum Information. Cambridge University Press. 2010. 710 p. 10. Mikio Nakahara. Quantum computing: an overview // Mathematical Aspects of Quantum Computing. 2007. 53 p. <http://www.worldscibooks.com/physics/6851.html>. 11. Whitney M.G. Practical Fault Tolerance for Quantum Circuits. A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science in the Graduate Division of the University of California, Berkeley. 2009. 206 p. 12. DiVincenzo D.P. The Physical Implementation of Quantum Computation // IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA. 9 p. Электронный ресурс: [arXiv:quant-ph/0002077](http://arXiv:quant-ph/0002077) v3 13 Apr 2000. [http://www.unifiedfieldtheories.com/0002077\\_DiVincenzo\\_Phys\\_Imp.pdf](http://www.unifiedfieldtheories.com/0002077_DiVincenzo_Phys_Imp.pdf). 13. Svore K.M., Terhal B.M., DiVincenzo D.P. Local Fault-Tolerant Quantum Computation // Электронный ресурс: [arXiv:quant-ph/0410047](http://arXiv:quant-ph/0410047) v2 6 Jun 2005. <http://research.microsoft.com/pubs/143764/local2005.pdf>. 14. Бейтсон Г. Шаги в направлении экологии разума / Г. Бейтсон. М.: КомКнига. 2005. 248 с. 15. Валиев К.А. Квантовые компьютеры: надежды и реальность / К.А.Валиев, А.А.Кокин. Ижевск: ПХД. 2001. 352 с. 16. Feinstein D.D.Y. Computer-Aided-Design Methods for Emerging Quantum Computing Technologies / D.D.Y. Feinstein. BiblioLabsII. 2011. 184 p. 17. Hayes J.P. Testing for Missing-Gate Faults in Reversible Circuits / John P. Hayes, Iliia Polian, Bernd Becker // Proc. Asian Test Symposium. Taiwan. November, 2004. 18. Матюшкин И.В. Квантовые клеточные автоматы / И.В. Матюшкин // Электронный научный журнал «ИССЛЕДОВАНО В РОССИИ». 2011. С. 367-392. Электронный ресурс <http://zhurnal.ape.relarn.ru/articles/2011/029.pdf>. 19. Feinstein D.Y. Partially Redundant Logic Detection Using Symbolic Equivalence Checking in Reversible and Irreversible Logic Circuits / D.Y. Feinstein, M.A. Thornton, D.M. Miller // Design, Automation and Test in Europe, DATE '08. 2008. P. 1378 –

1381. **20.** *Nayeem N.M.* Online Fault Detection in Reversible Logic / N.M. Nayeem, J.E. Rice // Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT). 2011. P. 426-434. **27.** *Golubitsky O.* A Study of Optimal 4-bit Reversible Toffoli Circuits and Their Synthesis / O. Golubitsky, D. Maslov // IEEE Transactions on Computers. 2011. P. 1-14. **28.** *Основы технической диагностики* / Под ред. П.П.Пархоменко. М.: Энергия. 1976. 460с. **29.** *Пархоменко П.П.* Основы технической диагностики (Оптимизация алгоритмов диагностирования, аппаратурные средства) / П.П. Пархоменко, Е.С. Согомоян. Под ред. П.П. Пархоменко. М.: Энергия. 1981. 320 с. **31.** *Novak O.* Handbook of testing electronic systems / O. Novak, E. Gramatova, R. Ubar. Czech University Publishing House. 2005. 402 p. **32.** *Автоматизация диагностирования электронных устройств* / Ю.В.Малышенко и др. / Под ред. В.П.Чипулиса. М.: Энергоатомиздат, 1986. 216с. **35.** *Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А.* Проектирование и верификация цифровых систем на кристаллах. Verilog & System Verilog: Харьков. Новое слово. 2010. 528с. **36.** Электронный ресурс: <http://www.scrigroup.com/limba/engleza/92/The-Design-Flow-and-Fault-Mode51775.php> **37.** *Андрюхин А.И.* Параллельная генерация тестов для МОП-структур на переключательном уровне / А.И. Андрюхин // Наукові праці ДонНТУ. Серія "Інформатика, кібернетика та обчислювальна техніка". 2010. Вип. 11(164). С. 75-78. **38.** *Harris I.G.* Hardware-Software Covalidation: Fault Models and Test Generation / Ian G. Harris // Design and Test of Computers. Vol. 20, Num. 4. July-August 2003. 12 p. Электронный ресурс: <http://www.ics.uci.edu/~harris/pubdir/hldvt01hsw.pdf> **39.** *Семенец В.В., Хаханова И.В., Хаханов В.И.* Проектирование цифровых систем с использованием языка VHDL. Харьков: ХНУРЭ. 2003. 492 с. **40.** *Надежность технических систем* / Под ред. И.А.Ушакова. М.: 1985. 512 с. **41.** *Stanisavljevi M.* Reliability of Nanoscale Circuits and Systems / M. Stanisavljevi, M. Schmid, Y. Leblebici. Springer. 2011. 240 p. **42.** *Fan X.* Fault diagnosis of VLSI designs: cell internal faults and volume diagnosis throughput / Xiaoxin Fan // PhD (Doctor of Philosophy) thesis, University of Iowa. 2012. 134 p. **43.** *Pomeranz I.* Transition Path Delay Faults: A New Path Delay Fault Model for Small and Large Delay Defects / I. Pomeranz, S.M. Reddy // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2008. Vol. 16, No.1. P. 98-107. **44.** *Pomeranz I.* Selection of a Fault Model for Fault Diagnosis Based on Unique Responses / I. Pomeranz, S.M. Reddy // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. – 2010. – Vol.18, No.11. P. 1533-1543. **45.** *Vanitha K.* Implementation of an integrated FPGA based automatic test equipment and test generation for digital circuits / K. Vanitha, C.A. Sathiyamoorthy // Intern. Conf. on Information Communication and Embedded Systems (ICICES). 2013. P. 741-746. **46.** *Niemann H.* Fault tolerant control based on active fault diagnosis / H. Niemann // Proceedings of the 2005 American Control Conference. 2005. Vol. 3. P. 2224-2229. **47.** *Abramovici M.* BIST-Based Delay-Fault Testing in FPGAs / Miron Abramovici, Charles E. Stroud // Journal of Electronic Testing: Theory & Applications. 2003. Vol. 19, No. 5. P. 549-558. **48.** *Ulbricht M.* A new hierarchical built-in self-test with on-chip diagnosis for VLIW processors / Markus Ulbricht, Mario Scholzel, Tobias Koal, Heinrich Theodor Vierhaus // Design and Diagnostics of Electronic Circuits & Systems (DDECS). April 2011. P. 143-146. **49.** *Elm M.* BIST: Scan-based Built-In self-diagnosis / M. Elm, H.-J. Wunderlich // Design, Automation & Test in Europe Conference & Exhibition (DATE). March 2010. P. 1243-1248. **50.** *Huang Yu-Jen.* A built-in self-test scheme for the post-bond test of TSVs in 3D ICs / Yu-Jen Huang, Jin-Fu Li, Ji-Jan Chen, Ding-Ming Kwai, Yung-Fa Chou, Cheng-Wen Wu // VLSI Test Symposium (VTS). May 2011. P.20-25. **51.** *Shianling Wu.* Logic BIST Architecture Using Staggered Launch-on-Shift for Testing Designs Containing Asynchronous Clock Domains / Wu Shianling, Laung-Terng Wang, Yu Lizhen, H. Furukawa, Wen Xiaoqing, W.-B. Jone, N.A. Touba, Zhao Feifei, Liu Jinsong, Hao-Jan Chao, Li Fangfang, Jiang Zhigang // Defect and Fault Tolerance in VLSI Systems (DFT). Oct. 2010. P. 358-366. **52.** *Da Silva F.* The Core Test Wrapper Handbook. Rationale and Application of IEEE Std. 1500™ / F. Da Silva, T. McLaurin, T. Waayers. Springer. 2006. XXIX. 276 p. **53.** *Marinissen E.J.* Guest Editors' Introduction: The Status of IEEE Std 1500 / E.J. Marinissen, Yervant Zorian // IEEE Design & Test of Computers. 2009. No26(1). P.6-7. **54.** *Elm M.* Scan Chain Organization for Embedded Diagnosis / M. Elm, H.-J. Wunderlich // Design, Automation and Test in Europe, DATE '08. 2008. P. 468-473.

*Поступила в редколлегию 11.03.2014*

**Baghdadi Ammar Awni Abbas**, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование вычислительных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: hahanov@kture.kharkov.ua.

**Хаханов Владимир Иванович**, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. IEEE Senior Member. IEEE Computer Society Golden Core Member. Научные интересы: проектирование и тестирование вычислительных систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, теннис, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: hahanov@kture.kharkov.ua.

**Литвинова Евгения Ивановна**, IEEE Member, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование цифровых систем и сетей на кристаллах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-421. E-mail: kiu@kture.kharkov.ua.