



### Литература

- 1 Портал Самарского государственного аэрокосмического университета [Электронный ресурс] – <http://www.ssau.ru/news/12208-Obyavlen-12-y-konkurs-molodykh-prepodavateley-i-nauchnykh-sotrudnikov-SGAU/> (дата обращения 16.03.2016 г.).
- 2 Еленев, Д.В. Личные кабинеты научно-педагогических работников университета [Текст] / Д.В. Еленев, А.О. Линник // Самара: Известия СНЦ РАН, 2015. №2(5), том 17. – С. 964-969.
- 3 Еленев, Д.В. Система личных кабинетов научно-педагогических работников / Д.В. Еленев, А.О. Линник // Международная научно-техническая конференция "Перспективные информационные технологии ПИТ-2015", 28 - 30 апр. 2015 г. [Текст] Самара: Изд-во СНЦ РАН, 2015 г. – С. 6 - 10.
- 4 Еленев, Д.В. Автоматизация системы управления национальным исследовательским университетом и мониторинга его деятельности [Текст] / Д.В. Еленев, В.С. Кузьмичев, Д.Е. Пашков // Программные продукты и системы. – 2012. – № 3. – С. 31-34.

Н.В. Ефимушкина

### ПРИМЕНЕНИЕ ИМИТАЦИОННЫХ МОДЕЛЕЙ ДЛЯ ИЗУЧЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ИХ ПОДСИСТЕМ

(Самарский государственный технический университет)

*Аннотация:* Описываются результаты использования имитационных моделей при изучении современных вычислительных систем и их подсистем бакалаврами и магистрами направлений «Информатика и вычислительная техника» и «Программная инженерия». Моделирующие программы позволяют исследовать типовые структуры этих объектов и режимы их работы. Они широко применяют принципы анимации для обеспечения наглядности и лучшего усвоения материала.

*Ключевые слова:* вычислительная система, имитация, процессор, память, конвейер.

Общеизвестно, что современные вычислительные системы характеризуются сложными структурами и режимами функционирования. Для изучения особенностей их работы используется теория вычислительных систем (ВС) [1, 3]. При этом применяются аналитические, имитационные и экспериментальные методы. Наиболее достоверные результаты позволяют получить эксперименты непосредственно над объектом в реальных или специально созданных условиях. Высокая сложность вычислительных систем ограничивает применение этих методов для обучения студентов.



Наиболее перспективными представляются методы имитационного моделирования, которые представляют функционирование системы в виде алгоритма, называемого имитационной моделью. Соответствующая программа содержит процедуры, воспроизводящие структуру системы и протекающие в ней процессы. Важнейшее свойство имитационного моделирования — универсальность. Метод позволяет исследовать системы любой сложности, с любым уровнем детализации, учитывая влияние различных факторов и воспроизводить типовые ситуации. При этом в модели можно отобразить наиболее важные элементы объекта, опуская второстепенные, что повышает эффективность обучения. Существенной особенностью рассматриваемого метода является возможность использования средств анимации, которые обеспечивают большую наглядность. Имитационные модели дают возможность экспериментатору формировать представления о свойствах и, познавая систему через ее модель, принимать обоснованные проектные решения.

На кафедре «Вычислительная техника» Самарского государственного технического университета в течение длительного времени разрабатываются и применяются в учебном процессе имитационные модели вычислительных систем и их подсистем. На их использовании построен лабораторный практикум по дисциплинам «Аппаратные средства вычислительной техники», «Распределенные высокопроизводительные вычислительные системы», «ЭВМ и периферийные устройства» и др.

Модели позволяют изучить особенности организации вычислительных процессов в центральных процессорах, памяти, многопроцессорных комплексах и сетях, а также исследовать влияние самых разнообразных факторов на производительность этих устройств и систем. Они обеспечивают оценку временных характеристик. В настоящее время применяются моделирующие программы для исследования следующих типовых элементов и систем:

- 1) Одно и нескольких конвейеров;
- 2) Суперскалярных процессоров;
- 3) Процессоров с явным параллелизмом команд (EPIC);
- 4) Подсистемы «кэш – оперативная память» однопроцессорной системы и многопроцессорной системы;
- 5) Подсистемы ввода-вывода данных с клавиатуры и монитора;
- 6) Многопроцессорного комплекса с фиксированной и переменной структурой;
- 7) Одноранговой локальной сети, работающей в условиях с помехами в канале и без них;
- 8) Локальной сети со звездообразной архитектурой;
- 9) Локальной сети с петлевой архитектурой;
- 10) Сети с маршрутизаторами.

Изучение начинается с простейших устройств и моделей. Первые работы посвящены исследованию пятиступенчатого конвейера и выявлению его основных свойств и режимов эффективной работы. Известно, что конвейерный принцип обработки широко применяется в современных процессорах. Студен-



ты, исследуя модель конвейера, убеждаются, что с ростом числа команд среднее время их выполнения стремится к длительности одной микрооперации. В следующей работе можно изучить влияние увеличения числа конвейеров на производительность системы.

Известно, что эффективная работа конвейера может быть нарушена из-за конфликтов одного из трех типов: структурных, по данным и по управлению. Все конфликты приводят к приостановке выполнения команды, в которой они возникли (pipeline stall), а также всех следующих за ней до конца конвейера. Эта ситуация называется «конвейерным пузырем» (pipeline bubble). Пузырь проходит по конвейеру, не выполняя никакой работы. Описываемые модели позволяют исследовать все описанные типы конфликтов и оценить их влияние на производительность системы.

В настоящее время существует три основных класса центральных процессоров: суперскалярные, с очень длинным командным словом и с явным параллелизмом команд. Все они содержат несколько параллельно работающих конвейеров. В первых конфликты разрешаются динамически, в процессе выполнения программы, во вторых – компилятором, до начала ее работы. Последний класс сочетает в себе достоинства первых двух. Большинство современных процессоров являются суперскалярными. Они имеют сложную структуру и режим работы. В модели типового суперскалярного процессора отображены наиболее существенные элементы оригинала, которые позволяют уяснить принципы его функционирования. Процессоры EPIC устраняют конфликты на этапе компиляции программ. В них могут возникать только коллизии по управлению. Модель процессора EPIC позволяет исследовать этот тип конфликтов.

Важным элементом вычислительных систем, наряду с центральными процессорами, является память. Она имеет многоуровневую иерархическую структуру и включает в себя регистровое, кэш, оперативное (ОП) и внешние запоминающие устройства. Основная идея такой иерархии – согласование скоростей работы операционных устройств, в первую очередь процессора, с запоминающими. При этом если информация хранится на верхнем уровне, то имеется ее копия и на нижнем. Например, кэш-память исполняет роль буфера между ОП и процессором. Ее организация существенно влияет на производительность системы в целом. Обращение к кэш-памяти может закончиться двойко: попаданием (если информация найдена) или кэш-промахом (если информации нет).

Существует три типа организации описываемой памяти:

- с прямым отображением;
- полностью ассоциативная;
- множественно-ассоциативная.

Первая является наиболее простой, но характеризуется большим числом промахов. Полностью ассоциативный кэш – самый быстродействующий и самый сложный. Наиболее распространенной является множественно-ассоциативная память, содержащая несколько параллельных блоков. В ней для поиска блоков используется прямое отображение, а внутри блока – полностью ассоциативный метод.



Студенты имеют возможность изучить с помощью модели кэш различной структуры, основные методы замещения строк в ней, а также влияние объемов памяти обоих типов и времени выполнения операций обращения к ним на общие характеристики работы системы. При этом важная роль отводится визуализации процессов, протекающих в подсистеме.

Содержимое кэш-памяти меняется под управлением процессора. При этом основная память может оставаться неизменной. С другой стороны, внешние устройства могут изменять данные в ОП в режиме прямого доступа. При этом кэш-память остается неизменной. Еще сложнее ситуация в мультипроцессорных системах, когда несколько процессоров обращаются к общей памяти. Возникает проблема когерентности кэш-памяти.

Модель типовой симметричной многопроцессорной вычислительной системы с однородным доступом к памяти и шинной организацией позволяет отобразить работу основной (оперативной), локальной и кэш-памяти. Студенты могут добавлять или убирать память любого из трех перечисленных типов, варьировать объемы памяти, число процессоров и количество команд обращения к памяти в моделируемой смеси. Таким образом, появляется возможность исследовать разнообразные структуры центральной части многопроцессорных систем и режимы их работы. Модели многопроцессорных комплексов с фиксированной и переменной структурой обеспечивают изучение влияния на характеристики системы различных методов организации вычислительного процесса.

Типовыми устройствами подсистемы ввода-вывода компьютера являются клавиатура и монитор. Особенности их совместной работы позволяет изучить соответствующая модель. Она имитирует ввод символов с клавиатуры с одновременным выводом на экран и без него.

В настоящее время широкое применение получили локальные вычислительные сети (ЛВС), которые строятся на ПЭВМ, соединенных линией связи и аппаратурой, поддерживающей соответствующий интерфейс. Информация в таких сетях передается в виде пакетов стандартной длины и формата. Наиболее распространенными сетевыми топологиями являются одноранговая, звездообразная и петлевая. В модели одноранговой сети имеется возможность исследовать типовые методы доступа: с проверкой столкновений и несущей, а также влияние помех на работу сети. При этом в наглядной форме на экране демонстрируются коллизии, возникающие в канале.

Модели звездообразной и петлевой топологий позволяют оценить влияние на основные характеристики сети параметров передающей среды, количества станций и числа передаваемых ими пакетов. При этом появляется возможность определить условия, обеспечивающие оптимальные режимы работы систем.

Наиболее сложную структуру имеют сети с маршрутизаторами. Последние образуют так называемую коммуникационную сеть, которая может иметь самую разнообразную топологию. Модель такой сети содержит 16 узлов, между которыми можно устанавливать различные маршруты. Она позволяет оце-



нить основные характеристики коммуникационной сети, имеющей различные структуры - от полносвязной до линейной. При этом появляется возможность определить режимы, при которых каждая структура является оптимальной.

Важной особенностью описываемых моделей является то, что соответствующие программы разработаны под руководством автора студентами, обучающимися на кафедре «Вычислительная техника» Самарского государственного технического университета. Они являются результатами научных исследований, выполняемых в СНО. В процессе разработки алгоритмов и программ студенты глубоко изучили соответствующие вопросы и выявили наиболее важные детали структуры и режимов работы моделируемых объектов.

### Литература

1. Таненбаум, Э. Архитектура компьютера: пер. с англ. / Э. Таненбаум.- Изд. 5-е.- СПб., 2010. - 848 с.
2. Хамахер, К. Организация ЭВМ: пер. с англ. / К. Хамахер, З.Вранешич, С. Заки; Сер.: Классика computer science.- Изд. 5-е.; - СПб: Питер, 2003г. - 845 с.
3. Орлов С.П. Организация компьютерных систем: учебное пособие/С.П. Орлов, Н.В. Ефимушкина. – Самара: Самар. гос. техн. ун-т, 2011. – 188 с.

И.М. Избяков, М.А. Кудрина

### ПРОГРАММА ДВОИЧНОГО НЕРАВНОМЕРНОГО АЛФАВИТНОГО КОДИРОВАНИЯ С РАВНОЙ ДЛИТЕЛЬНОСТЬЮ ИНФОРМАЦИОННЫХ СИГНАЛОВ

(Самарский национальный исследовательский университет  
имени академика С.П. Королёва)

Цель данной работы состояла в разработке программного обеспечения (ПО), позволяющего автоматизировать процесс создания заданий для лабораторных работ по предмету «Теория информации» для студентов, обучающихся по направлению 230100.62 – Информатика и вычислительная техника. Разработанное ПО позволяет составлять задания по теме «Двоичное неравномерное алфавитное кодирование с равной длительностью информационных сигналов», а именно кодировать и декодировать сообщения при помощи различных методов неравномерного кодирования.

При применении неравномерного кодирования символы исходного алфавита кодируются комбинациями символов двоичного алфавита – «0» и «1», длина кодов отдельных символов варьируется в зависимости от вероятности появления символов в первичном алфавите. Длительности элементарных сигналов «0» и «1» при этом равны. Оптимизация кодирования в данном случае происходит за счет того, что символам первичного алфавита, которые встречаются чаще, присваиваются более короткие коды, а символам, встречающимся более редко – более длинные.



Рассмотрим кодовую таблицу для букв русского алфавита (см. табл. 1), используем для построения кода таблицу частотности русских букв из источника [1].

Табл. 1. Коды букв русского алфавита при неравномерном кодировании

буква	код	$p_i$	$k_i$	буква	код	$p_i$	$k_i$
пробел	000	0,145	3	я	1011000	0,019	7
о	100	0,095	3	ы	1011100	0,016	7
е, ё	1000	0,074	4	з	1101000	0,015	7
а	1100	0,064	4	ь, ъ	1101100	0,015	7
и	10000	0,064	5	б	1110000	0,015	7
т	10100	0,056	5	г	1110100	0,014	7
н	11000	0,056	5	ч	1111000	0,013	7
с	11100	0,047	5	й	1111100	0,01	7
р	101000	0,041	6	х	10101000	0,009	8
в	101100	0,039	6	ж	10101100	0,007	8
л	110000	0,036	6	ю	10110000	0,006	8
к	110100	0,029	6	ш	10110100	0,006	8
м	111000	0,026	6	ц	10111000	0,004	8
д	111100	0,026	6	щ	10111100	0,003	8
п	1010000	0,024	7	э	11010000	0,003	8
у	1010100	0,021	7	ф	11011000	0,002	8

Разделителем отдельных кодов будет выступать последовательность «00», а разделитель слов (то есть пробел) будет кодироваться последовательностью «000». При таких условиях становятся очевидными особенности построения кодов [1-3]:

- 1) разделитель кодов никогда не существует отдельно, поэтому он сразу включается в конец кода любого символа первичного алфавита;
- 2) разделитель кодов никогда не существует отдельно, поэтому он сразу включается в конец кода любого символа первичного алфавита;
- 3) комбинация двух и более элементарных сигналов «0» внутри одного кода недопустима, так как в противном случае она будет восприниматься как конец символа;
- 4) любой код всегда начинается с «1»;
- 5) разделителю слов «000» всегда предшествует признак конца знака «00», образуя последовательность «00000», таким образом, комбинации «000» и «0000» в конце кода не воспринимаются как разделитель слов, а значит коды могут оканчиваться на «0» и на «00» до признака конца знака «00».

На рисунке 1 приведен пример декодирования сообщения, зашифрованного неравномерным методом.

Код будет считаться оптимальным, если каждый элементарный символ будет передавать максимальную информацию. А это, согласно свойству энтропии, имеет место только в случае равновероятности состояний, поэтому в основе оптимального кодирования лежит требование, чтобы элементарные символы в закодированном тексте встречались в среднем с одинаковой частотой.