

*Ефимушкина Наталья Владимировна,
Орлов Сергей Павлович*

ИМИТАЦИОННАЯ МОДЕЛЬ МУЛЬТИКОНВЕЙЕРНОЙ СИСТЕМЫ СУПЕРСКАЛЯРНОГО МИКРОПРОЦЕССОРА

В настоящей статье описывается имитационная модель для лабораторной работы по изучению структур конвейеров процессоров. В первых двух лабораторных работах с помощью модели изучались основные принципы и динамика выполнения команд в одиночном конвейере [1].

Изучение работы простейшего конвейера приводит к выводу: при эффективной работе время выполнения одной команды в нем стремится к одному такту (длительности одной микрооперации). Дальнейшее повышение производительности возможно только за счет параллельного выполнения нескольких команд, то есть использования несколько параллельно работающих конвейеров (мультиконвейерная система) [2, 3]. Такую архитектуру имеют все современные суперскалярные микропроцессоры. В отличие от векторных и матричных процессоров в суперскалярном процессоре конвейеры одновременно выполняют разнородные команды над данными различных типов. Процессор содержит конвейерные устройства для длинных и для коротких операций (для операций с памятью, команд

целочисленной арифметики и арифметики с плавающей точкой).

Особенности работы такого процессора изучаются с помощью программы **ConvMod**. На рис. 1 представлен основной интерфейс программы. Модель содержит конвейеры двух типов: короткие (5-ти ступенчатые) и длинные (10-ти ступенчатые). Кроме того, имитируется оперативная память компьютера, из которой загружается программа, кэш-память команд и кэш-па-

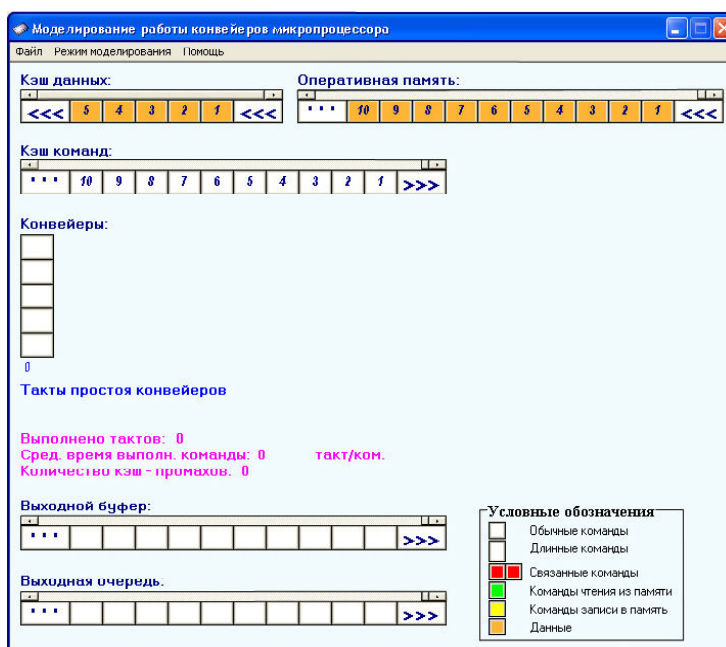
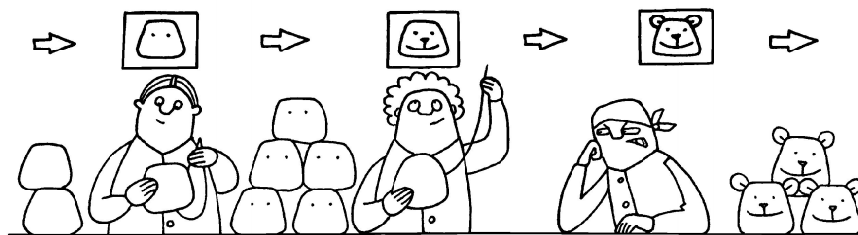


Рис. 1



мать данных, входящие в состав процессора, а также выходной буфер выполненных команд и выходная очередь завершенных команд.

Исполняемая программа в модели представляется как смесь команд, свойства которых могут быть заданы учащимся. Она может содержать однотипные (короткие) команды, а также команды, выполняемые разными (длинными и короткими) конвейерами.

Исходными данными для моделирования являются:

- 1) общее количество команд в моделируемой программе (от 100 до 500 команд),
- 2) частота появления длинных команд в общей смеси команд (чередование через 1, 2 или более короткие команды),
- 3) количество коротких и длинных конвейеров (от 1 до 10), при этом длинные конвейеры могут вообще не использоваться.

Модель позволяет определить следующие основные временные характеристики:

а) общее количество тактов, выполненных всеми конвейерами при обработке заданного числа команд,

б) среднее время выполнения одной команды в тактах.

Предусмотрено три режима:

- а) непрерывный,
- б) потактовый,
- в) по таймеру.

Выбор режимов осуществляется с помощью пункта меню **Режим моделирования**. Непрерывный режим практически не позволяет пользователю следить за ходом моделирования. Он нужен, чтобы быстро получить результаты, например, среднее время выполнения команды или количество тактов простоя конвейеров.

Потактовый режим позволяет в любой момент просматривать с помощью полос прокрутки содержимое кэша команд, кэша данных, оперативной памяти и выходной очереди, а также выходного буфера.

Режим «по таймеру» позволяет варьировать скорость процесса моделирования. При этом имитация осуществляется непрерывно. Интервал между тактами можно задать с помощью меню **Файл → Настройка → Другое**. По умолчанию он равен 1 секунде.

Перед началом моделирования необходимо задать исходные данные (выполнить настройку модели). Эта процедура выполняется с помощью меню **Файл → Настройка**. Выбрав одну из закладок, можно задать соответствующие параметры модели. Так с помощью закладки **Команды** устанавливается количество команд в моделируемой последовательности. Здесь же имеется кнопка, позволяющая включить в последовательность длинные команды и задать их чередование. С помощью закладки **Конвейеры** устанавливается тип конвейеров (длинные или короткие) и их количество.

Имитация выполнения последовательности команд в модели начинается после настройки и выбора пункта меню **Файл → Пуск**. При этом происходит перемещение номеров команд из кэш-памяти команд через конвейерные обрабатывающие устройства в выходной буфер и очередь. Моделирование заканчивается, когда все команды оказываются в выходной очереди. Результаты представляются в виде общего количества выполненных тактов и среднего числа тактов на команду.

Программа, реализация которой имитируется моделью, как отмечалось выше, может содержать операции обычной длины или

представлять собой смесь коротких и длинных команд, которые можно обработать на соответствующих конвейерах. Длинные команды при отображении во входной и выходной очередях, а также в выходном буфере имеют больший размер, чем обычные. При прохождении конвейеров все операции представляются прямоугольниками одинакового размера.

Программа **ConvMod** позволяет сделать вывод о том, что в идеальном случае n конвейеров одного типа (коротких или длинных) за такт могут выполняться n команд. Пример экранной формы моделирования мультиконвейерной системы приведен на рис. 2.

Эффективная работа конвейера может быть нарушена из-за конфликтов одного из трех типов [4]:

- а) структурных конфликтов, которые возникают из-за занятости отдельных ресурсов (например, кэш-памяти или оперативной памяти);
- б) конфликтов по данным, вызванных тем, что очередная команда использует результат предыдущей, а она еще не завершена в другом конвейере;
- с) конфликтов по управлению, которые возникают при наличии операций условных переходов, изменяющих адрес следующей команды.

Для устранения конфликтов первого типа увеличивают размер памяти или организуют многопортовую память. Конфликты по данным ликвидируются за счет изменения порядка выполнения команд. При этом между командами, связанными по данным, вставляются другие независимые команды, стоящие в конце программы. Это позволяет загрузить конвейер вычислениями, пока данные нужной команды не будут готовы.

Конфликты, как уже отмечалось, приводят к приостановке

работы конвейера и снижают производительность процессора. Программа **ConvMod** позволяет изучить все описанные типы конфликтов и оценить их влияние на время выполнения команд.

Имеется возможность воспроизвести конфликт по данным между двумя соседними командами программы, которые именуются «связкой». Эти команды в модели выделяются красным цветом (см. рис. 3). В связку могут входить как обычные, так и длинные команды. Если в некоторый конвейер поступает одна из команд связки, то вторая ожидает во входной очереди, пока первая команда не пройдет по конвейеру и не попадет в выходной буфер. В это время для заполнения конвейеров из входной очереди выбираются следующие команды.

По содержанию выходного буфера видно, что команды завершаются не в том порядке, в котором были в исходной программе. Этот эффект называется неупорядоченным выполнением команд. Выходная очередь восстанавливает первоначальный порядок следования команд в программе.

В разработанной модели также воспроизводится структурный конфликт, когда операнды очередной команды отсутствуют в

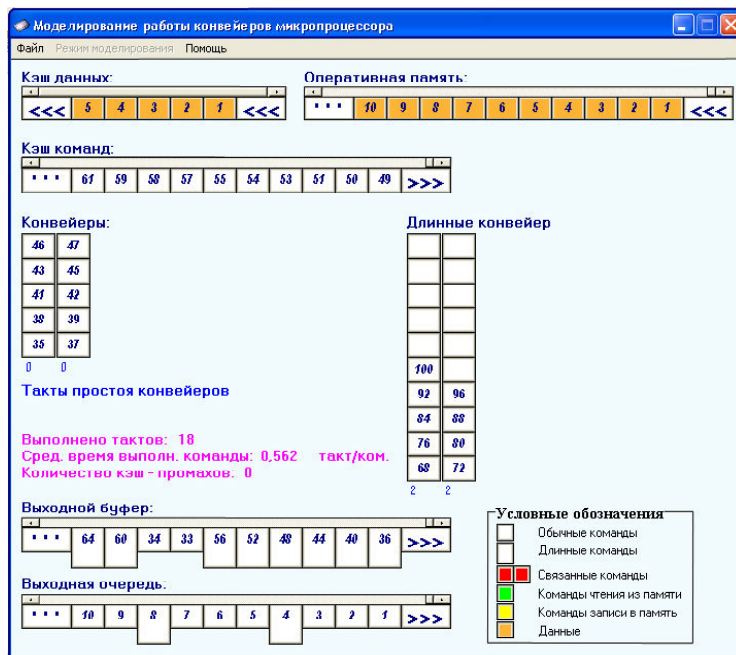


Рис. 2

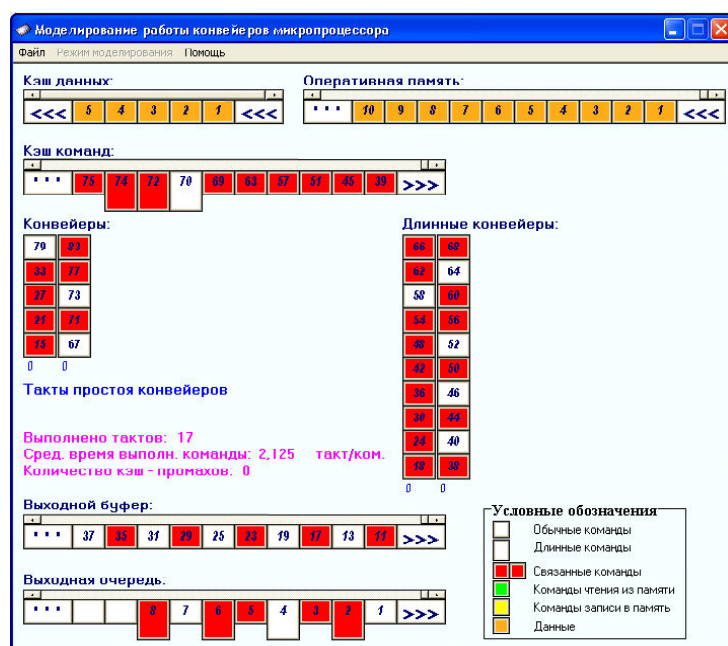


Рис. 3

кэш-памяти. В этом случае выполняются команды записи/чтения из основной памяти. Они добавляются в моделируемую программу с помощью меню **Файл** → **Настройка** → **Включить команды работы с памятью**. Любая программа содержит не только операции обращения к оперативной памяти,

но и другие команды. Поэтому обычно задается частота таких операций и процентное соотношение режимов записи и чтения. По умолчанию считается, что операции чтения/записи встречаются после каждой обычной команды, а их соотношение равно 1 к 9 (10% записи и 90% чтения). Время выполнения записи больше, чем время чтения. Операции записи и чтения в оперативную память обрабатываются в длинных конвейерах, которые должны быть установлены с помощью закладки **Конвейеры**.

Процесс моделирования выполнения некоторой программы, содержащей команды работы с оперативной памятью, представлен на рис. 4.

Заключение

Модели демонстрируют уменьшение времени выполнения программ за счет конвейеризации и использования нескольких параллельно работающих конвейерных устройств. Они позволяют также изучить, какие конфликты возникают в процессорах и как они устраняются.

Имитационные модели разработаны с использованием универсальных сред (Delphi и C++). Их важной особенностью является визуализация процессов, протекающих в ЭВМ и устройствах. Она обеспечивает максимальную наглядность и оптимальный режим обучения.

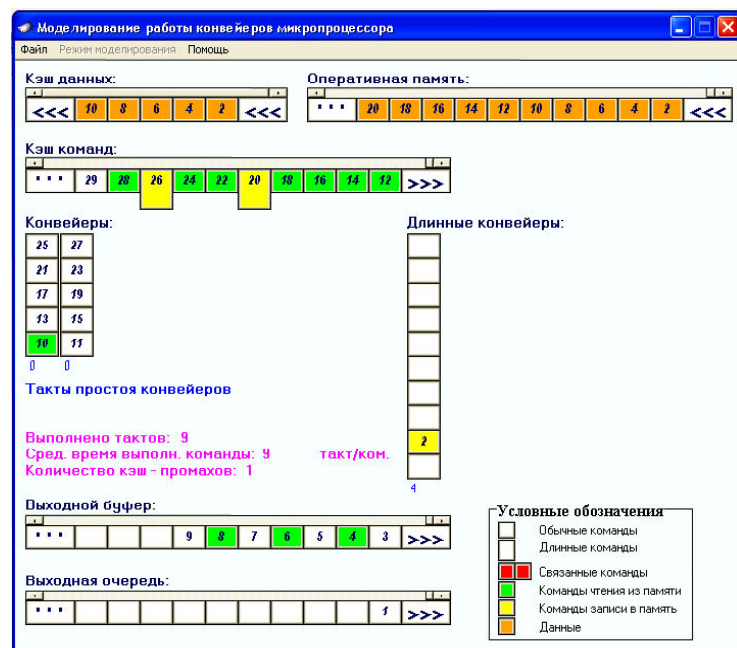


Рис. 4

Литература

1. *Ефимушкина Н.В., Орлов С.П.* Имитационная модель конвейера в микропроцессоре // Компьютерные инструменты в школе, 2014. № 4. С. 8–11.
2. *Таненбаум Э.* Архитектура компьютера / Изд. 5-е. СПб.: Питер, 2007.
3. *Хамахер К., Вранешич З., Заки С.* Организация ЭВМ / Изд. 5-е. СПб.: Питер, 2003.
4. *Ефимушкина Н.В., Орлов С.П.* Организация компьютерных систем: учебное пособие. Самара: Самар. гос. техн. ун-т, 2011.

*Орлов Сергей Павлович,
доктор технических наук, профессор,
заведующий кафедрой
Вычислительная техника СамГТУ,
Ефимушкина Наталья Владимировна,
кандидат технических наук, доцент,
доцент кафедры Вычислительная
техника СамГТУ.*



Наши авторы, 2013.
Our authors, 2013.

