



3. Лукьяненко Т. Имидж региона как фактор инвестиционной привлекательности // Рынок ценных бумаг. № 11 (146). 1999. С. 48-50
4. Кравец А. Г., Кучеренко А. А. Исследование регионального инвестиционного климата на основе факторного анализа // Прикаспийский журнал: управление и высокие технологии. — 2019. — №2. — Стр. 60-71.
5. Statsoft, программный пакет Statistica [Электронный ресурс] /. — Электрон. журн. — Режим доступа: <http://statsoft.ru/>, свободный
6. Симчера, В.М. Методы многомерного анализа статистических данных /В.М. Симчера. — Москва: Финансы и статистика, 2008. — 400с.

М.Э. Мамышев

## ИМИТАЦИОННАЯ МОДЕЛЬ СУПЕРСКАЛЯРНОГО ПРОЦЕССОРА

(Самарский государственный технический университет)

Описывается подход к построению имитационной модели типового суперскалярного процессора и сама модель. В ней воспроизводятся основные блоки процессора и отображается процесс выполнения программы, состоящей из заданной пользователем смеси команд. Результатом моделирования является общее время выполнения программы и среднее количество тактов на команду.

*Ключевые слова:* имитация, суперскалярный процессор, конфликты.

Суперскалярным называется центральный процессор (ЦП), который одновременно выполняет более чем одну скалярную команду. Это достигается за счет включения в состав ЦП нескольких самостоятельных функциональных блоков, каждый из которых отвечает за свой класс операций и может присутствовать в процессоре в нескольких экземплярах. Все современные процессоры линейки Intel Core являются суперскалярными. Можно считать типовой микроархитектуру Intel Core i7 Sandy Bridge, который состоит из четырех основных блоков: подсистемы памяти и устройств предварительной обработки, контроля исполнения с изменением последовательности, а также собственно исполнения.

Блок исполнения включает в себя планировщик ёмкостью 54 записи и шесть функциональных устройств:

- 1) АЛУ 1 и блок умножения с плавающей точкой.
- 2) АЛУ 2 и блок сложения/вычитания с плавающей точкой.
- 3) АЛУ 3 и блок обработки переходов и сравнений с плавающей точкой.
- 4) Устройство сохранения.
- 5) Устройство загрузки 1.
- 6) Устройство загрузки 2.

Наиболее перспективным для исследования суперскалярных процессоров представляется метод имитационного моделирования. При разработке имитационной модели решался целый ряд проблем:

- 1) определение цели исследования;



- 2) выбор элементов объекта, которые должны отображаться в модели;
- 3) определение уровня детализации структуры и функций объекта;
- 4) оценка адекватности модели.

Важнейшими характеристиками вычислительных систем и их устройств являются временные, поэтому при разработке модели первоочередной является оценка именно таких характеристик типового суперскалярного процессора. Указанная цель определила особенности решения остальных проблем.

Так при решении второй проблемы было признано целесообразным воспроизводить в модели те элементы процессора, которые оказывают существенное влияние на временные характеристики. В частности, в ней отображены четыре основных блока общей микроархитектуры Intel Core i7. Что касается схемы тракта данных, то из нее в модели представлены следующие элементы:

- 1) кэш команд первого уровня;
- 2) блок выборки (2 очереди по 20 команд);
- 3) блок дешифрации (3 простых дешифратора и 1 сложный);
- 4) очередь декодированных команд (2 очереди по 28 микроопераций);
- 5) кэш микроопераций;
- 6) буфер переупорядочивания (Reorder Buffer);
- 7) планировщик (Scheduler);
- 8) функциональные блоки (3 АЛУ, 2 устройства чтения из памяти и 1 устройство записи);
- 9) буфер восстановления последовательности.

Общеизвестно, что на работу вычислительных систем и устройств существенное влияние оказывают случайные факторы:

- а) времена запуска программ в общем случае – случайные величины;
- б) исходные данные этих программ являются случайной выборкой из всего множества возможных значений.

Поэтому признано целесообразным параметры программ, исполняемых процессором, представлять, как случайные величины.

Что касается уровня детализации параметров модели, то структура микропроцессора представлена с точностью до перечисленных выше элементов, а исполняемая им программа – с точностью до отдельных команд. Смесь этих команд формируется случайным образом, сохраняя процентное соотношение команд разных типов. Типы команд определяются особенностями структуры модели и включают в себя:

- 1) одноктоковые команды;
- 2) 2-4 – тактовые команды;
- 3) 5-9 – тактовые команды;
- 4) 10-20 – тактовые команды;
- 5) операции чтения из памяти;
- 6) операции записи в память;
- 7) команды переходов.



Оценка адекватности модели выполнена путем сравнения результатов моделирования с характеристиками реального процессора, измеренными с помощью соответствующего программного монитора. Результаты показали приемлемую погрешность моделирования.

Имитационная модель разработана на языке C# с пользовательским интерфейсом на основе Windows Forms (.NET Framework). Главное окно программы представлено на рисунке 1. Имеется возможность задавать общее число команд моделируемой программы и процентное соотношение команд разных типов. Для этого используется окно настроек, внешний вид которого представлен на рисунке 2.

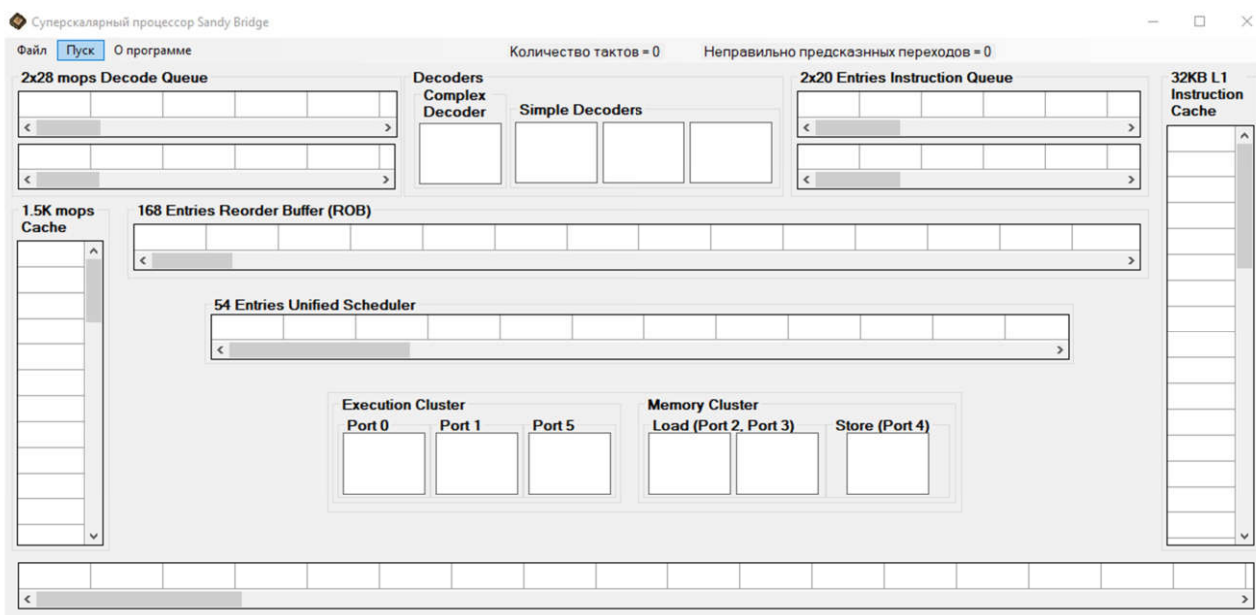


Рис 1. Главное окно программы имитационного моделирования типового суперскалярного процессора

Моделирование может выполняться в одном из двух режимов: непрерывно или по тактам. Последний иллюстрирует рисунок 3. Сначала генерируется последовательность команд (3-a), затем 6 команд поступает в очередь инструкций (3-b). Далее эти команды загружаются в декодер (3-c). После декодирования микрокоманды попадают в очередь декодированных команд (3-d) и в кэш микроопераций (3-e). При следующем декодировании таких команд микрооперации будут загружаться из этого кэша, минуя декодер.

В следующем такте микрокоманды загружаются в буфер переупорядочивания (3-f). Здесь происходит разбиение команд на независимые (от данных) и зависимые. Первые подаются в планировщик (3-g), а зависимые ожидают выполнения предшествующей им микрооперации, результат которой является их операндом.

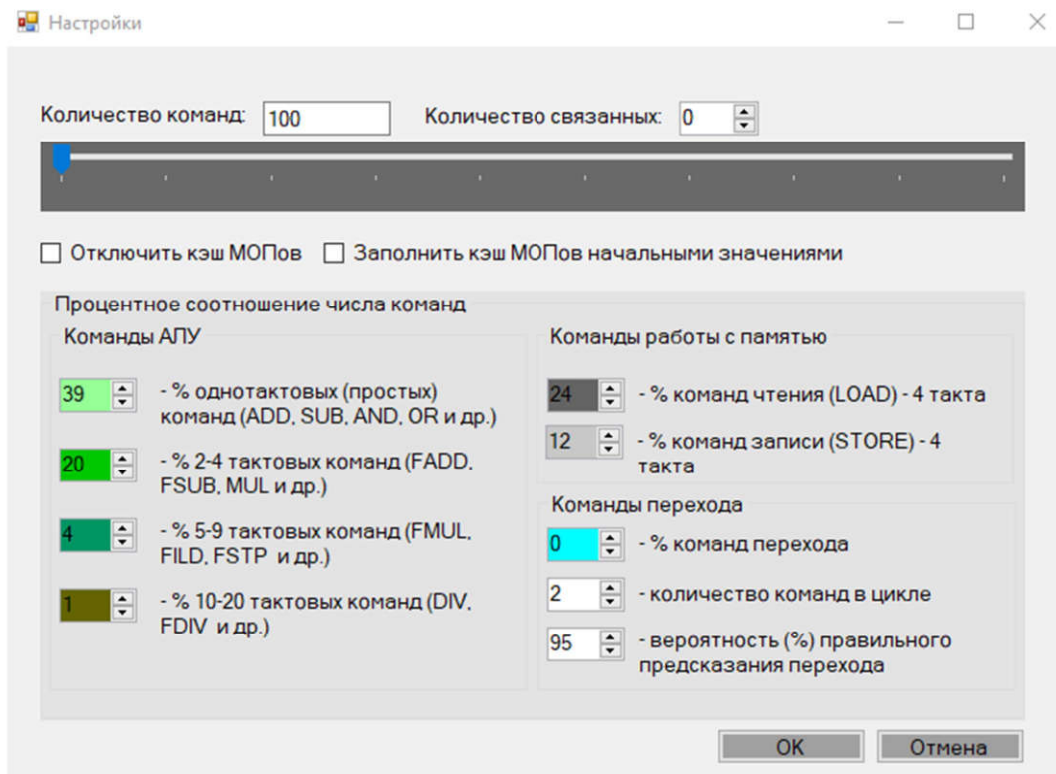


Рис 2. Окно настроек модели



Рис 3. Работа программы моделирования в режиме по тактам



Выполнение команд происходит в исполнительном блоке (3-h). После него микрооперации попадают в буфер восстановления последовательности (3-i) и ожидают выполнения остальных команд. Команда удаляется из буфера, если все предшествующие ей уже выполнены. При этом считается, что она завершена.

Результатом моделирования является общее время выполнения программы и среднее количество тактов на команду.

### Литература

1. Таненбаум, Э. Архитектура компьютера: пер. с англ. / Э. Таненбаум.- Изд. 5-е.- СПб., 2010. - 848 с.
2. Организация вычислительных машин и систем/ С.П.Орлов, Н.В. Ефимушкина. – Самара: Самар. гос. техн. ун-т, 2016. – 304 с.
3. Orlov S.P. and Efimushkina N.V., “Simulation models for parallel computing structures”, 2016 XIX IEEE International Conference on Soft Computing and Measurements (SCM), IEEE Conference Publications. V.1. P. 231-234. Publisher: IEEE Xplore, 2016.

И.Х.Н. Нааман, С.А. Потемкин, Н.И. Мельникова

## СКОРОСТНОЙ АЛГОРИТМ ФОРМИРОВАНИЯ ИЗОБРАЖЕНИЙ С ВЫСОКИМ РАЗРЕШЕНИЕМ

(Саратовский государственный технический университет  
имени Гагарина Ю.А.)

Увеличение разрешения изображений является фундаментальной задачей обработки видеoinформации. Разработчики предъявляют крайне высокие требования к аппаратному обеспечению, но во многих случаях можно воспользоваться альтернативным методом. Значительно экономнее получить несколько изображений невысокого разрешения, и затем восстановить (реконструировать) из них изображение высокого разрешения [1-4].

Под восстановлением будем понимать построение из нескольких изображений низкого разрешения (ИНР) одного изображения с бóльшим разрешением (с меньшими пикселями). Если разрешения ИНР и изображения высокого Разрешения (ИВР) совпадают, то речь идет о повышении четкости изображения, уменьшении размытия (deblurring). Два изображения тогда будут одинакового разрешения, когда одно изображенис может быть получено интерполированием другого или в том случае, когда характерные элементы в обоих изображениях идентичны. В таком понимании «разрешение» изображение не может быть просто соотнесено с размерами пикселей или максимальной частотой в изображении (например, добавление высокочастотного шума не повышает разрешения картинки).

Алгоритмы восстановления основаны на нахождении соответствия между