

## ЭМУЛЯТОР ЧЕТЫРЕХРАЗЯДНОГО ПРОЦЕССОРА

Г.И. Костарев, И.А. Тутов  
Томский политехнический университет  
[kostarevgi@gmail.com](mailto:kostarevgi@gmail.com)

### Введение

На сегодняшний день, информационные технологии и вычислительная техника являются одними из самых стремительно развивающихся отраслей в мире. Современный человек каждый день сталкивается с различными вариациями электронно-вычислительных устройств, которые значительно облегчают процессы получения, обработки и обмена информацией. Многообразие архитектур построения данных систем определяет широкий спектр направлений профессиональной подготовки специалистов в данной области. Вместе с тем, во многих высокотехнологичных сферах, таких как микропроцессорная техника, робототехника, космическая техника и т.п., специалисту необходимо знать и понимать не только общие принципы построения таких систем, но и глубинные процессы, происходящие непосредственно на аппаратном уровне. При подготовке специалиста, на данном этапе возникает проблема высокой сложности изложения изучаемого материала в доступной и наглядной форме.

Как правило, в силу того, что современные вычислительные системы имеют крайне сложную архитектуру, подача материала в текстовом виде не позволяет обучающемуся в полной мере понять все процессы, которые одновременно протекают в подобных устройствах, поэтому обучение ведется с применением программ-эмуляторов, которые воспроизводят логическую структуру и функционал простейших моделей процессоров.

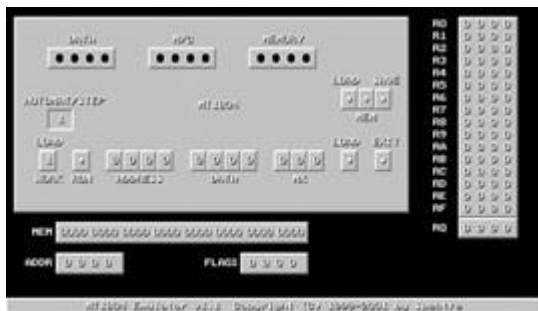


Рис. 1. Эмулятор MT1804 (MS-DOS)

Но, к сожалению, практика показывает, что разработчики подобных эмуляторов, при написании программы, стараются достигнуть максимальной схожести с функционалом эмулируемого устройства, не уделяя должного внимания созданию дружелюбного графического интерфейса пользователя (см. рис.1.), а также наглядности отображения процессов, происходящих внутри вычислительной машины. Еще одной проблемой является устаревание операционной системы (как

правило MS-DOS или Win95 – 98), для которой разработано большинство подобных программ.

### Проектирование архитектуры

Как результат решения описанных выше проблем, была поставлена задача создать современную программу-эмулятор, с простой четырехразрядной архитектурой, которая воспроизводила бы все свойства и функционал процессора, при этом являясь достаточно наглядной и легкой в освоении (при разработке архитектуры упор делался в первую очередь на доступность к пониманию процессов, происходящих внутри, в ущерб производительности).

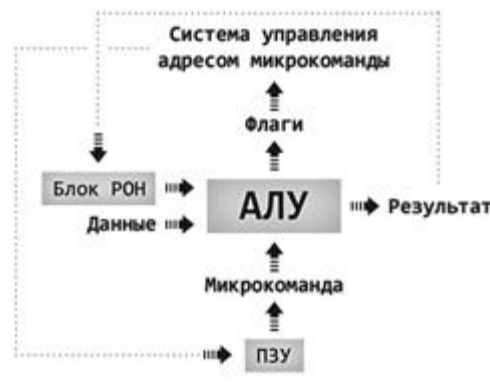


Рис. 2. Структура эмулируемого процессора

Центральное место в разработанной архитектуре занимает 4-х разрядное арифметико-логическое устройство (АЛУ), которое позволяет выполнять ряд основных логических и математических операций процессора, таких как:

- сложение чисел без учета знака (от 0 до 15)
- сложение чисел с учетом знака (от -7 до 7)
- вычитание чисел с учетом знака
- логическое сложение (OR)
- логическое умножение (AND)
- исключающее «ИЛИ» (XOR)
- инкремент числа

Процесс выполнения программы начинается с загрузки микрокоманды процессора с постоянного запоминающего устройства (ПЗУ). После этого, мультиплексоры определяют, какие данные будут отправлены на вход АЛУ (это могут быть внешние данные, либо данные ПЗУ, либо данные из блока регистров общего назначения (РОН)). Затем арифметико-логическое устройство производит с поступившими числами заданную операцию и формирует регистр флагов (в данном АЛУ предусмотрены флаги по переполнению, нулевому, ненулевому и отрицательному результату), после

чего данные поступают в регистр сдвига. Сдвиговый регистр, в зависимости от команды, производит (или не производит) сдвиг двоичного числа в соответствующем направлении (операция эквивалентна умножению или делению числа на 2) и отправляет результат в демультиплексор и буфер. Демультиплексор отвечает за внесение результата операции в блок РОН, а буфер, за вывод информации во внешнюю среду. После выполнения всех описанных выше действий, система управления адресом микрокоманды, которая состоит из генератора тактовых импульсов, счетчика, мультиплексора и дешифратора, определяет, к какому адресу памяти ПЗУ необходимо обратиться на следующем шаге выполнения программы. На данном этапе, происходит сравнение флагов арифметико-логического устройства с флагами перехода, которые задает пользователь, и в случае совпадения, счетчик (и дешифратор, который им управляется) принимает значение адреса следующей строки памяти ПЗУ. В случае команды остановки, счетчик отключается и выполнение программы прерывается.

Такая структура позволяет реализовывать простейшие базовые конструкции языков программирования, например:

- условный оператор «if – else»
- цикл «do – while»
- безусловный переход «goto»

## Разработка

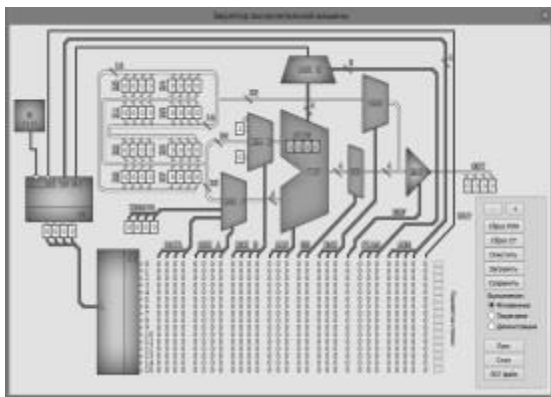


Рис. 3. Интерфейс эмулятора вычислительной машины

Создание программы происходило в интегрированной среде разработки Embarcadero RAD Studio, на языке Delphi, с использованием стандартных компонентов. После запуска эмулятора, пользователю сразу предоставляется схема внутреннего устройства процессора, с возможностью получения информации об управляющих командах (см. рис. 4.), структуре и функциональном назначении каждого из элементов. Большинство элементов, помимо текстового описания, снабжены интерактивной анимацией, наглядно демонстрирующей принцип их работы.



Рис. 4. Вспомогательные меню, содержащие команды управления элементами

Также, в эмуляторе реализована динамическая подсветка, что позволяет акцентировать внимание на действии или операции, которая выполняется в данный момент. Для удобства пользователя, в программе предусмотрена возможность загрузки и сохранения файлов, содержащих данные ПЗУ, а также возможность проследить ход выполнения программы, при помощи генерации так называемого ЛОГ-файла, в котором команды машинного кода расшифровываются и представляются в удобочитаемом виде.

## Заключение

Результатом применения данного эмулятора в учебном процессе в рамках изучения дисциплины «Микропроцессорная техника» является увеличение скорости освоения материала студентами по соответствующей тематике с двадцати до четырех академических часов. Обучающиеся отмечают удобство и простоту работы с программой, а также проявляют инициативу досрочного выполнения лабораторных работ. В последующем, студенты, прошедшие обучение на эмуляторе, быстрее осваивают курс программирования на языке «Assembler» и испытывают меньше трудностей при изучении архитектур других вычислительных систем. К настоящему моменту, созданная программа находится в процессе государственной регистрации в Федеральном институте промышленной собственности (ФИПС).

## Литература

1. Таненбаум Э. Архитектура компьютера.: Пер. с англ. – СПб.: Питер, 2012. – 846с.
2. Кнут Д. Э. Искусство программирования, том 1, выпуск 1. MMIX – RISC-компьютер нового тысячелетия.: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2007. – 160с.
3. Осипов Д. Delphi. Профессиональное программирование.: «Символ-плюс», 2006. – 1056с.
4. Тутов И. А. , Костарев Г. И. Организация учебного процесса подготовки программистов с использованием интегрированного в эмулятор учебника // Высшее образование сегодня. - 2014 - №. 9.