

## О РАЗРАБОТКЕ ПРОГРАММНЫХ МОДЕЛЕЙ МИКРОКОНТРОЛЛЕРОВ

© 2012 А. П. Жмакин<sup>1</sup>, Д. И. Селиванов<sup>2</sup>

<sup>1</sup>*канд. техн. наук, доцент каф. программного обеспечения  
и администрирования информационных систем  
e-mail: anatoly.zhmakin@gmail.com*

<sup>2</sup>*студент факультета физики, математики и информатики  
e-mail: diselivanov@gmail.com*

*Курский государственный университет*

В статье выделены классификационные признаки микроконтроллеров и предложен вариант классификации однокристалльных ЭВМ; описана авторская программная модель виртуальной машины KP1878BE1, учитывающая требования, предъявляемые к обучающим моделям микроконтроллеров.

**Ключевые слова:** микроконтроллер, однокристалльная ЭВМ, классификация, модель микроконтроллера.

Микроконтроллеры (МК) в настоящее время можно рассматривать как одно из наиболее массовых изделий электронной техники.

Особенностями микроконтроллеров (по сравнению с микропроцессорами) является их «самодостаточность»: на кристалле МК, помимо процессорного ядра, обязательно присутствует память (часто несколько разновидностей), тактовая система и некоторый набор внешних устройств, позволяющих включать такой МК непосредственно в контур управления самых разнообразных систем.

Микроконтроллеры нашли свое применение во многих встроенных системах, например в бытовой технике, где нет необходимости большой точности вычисления, но нужно учитывать изменения, например амплитуды сигнала, течение времени и т.д. В настоящее время достаточно широко используются как однокристалльные ЭВМ нового поколения, способные выполнять миллионы операций в секунду, так и микроконтроллеры более ранних модификаций. Выбор типа микроконтроллера обусловлен тем кругом задач, который стоит перед разработчиками. Решение поставленной задачи выводит на передний план проблему классификации однокристалльных ЭВМ.

До настоящего времени не существует устоявшейся классификации микроконтроллеров. Это обусловлено тем, что «микроконтроллеры – это класс быстро развивающихся электронных приборов, поэтому должно пройти некоторое время, чтобы система структурировалась» [Рюмик 2010: 20]. Предложенная Рюмиком классификация, являясь достаточно полной, не акцентирует внимание на тех архитектурных особенностях, которые, на наш взгляд, являются важными. Автор выделяет следующие «классообразующие» признаки: по разрядности шины данных ЦПУ, по архитектуре вычислительной системы, по фирменным платформам, по выполняемым функциям, по семействам 8-битных МК. В то же время в предложенной классификации не выделяются такие важные особенности, как тип памяти программ, форма входной и выходной информации и другие. Очевидно, любую классификацию технических средств нельзя считать идеальной, так как существуют объекты, которые можно отнести сразу к нескольким выделенным ранее классам одного классификационного признака.

В данной работе предлагается классифицировать МК по нескольким признакам:

- *разрядность микроконтроллера* – 4, 8, 16, 32 бита;
- *функциональное назначение* – системы реального времени, системы цифровой обработки сигналов, специализированные системы, системы на кристалле;
- *архитектурные особенности*:
  - архитектура процессора – RISC или SISC,
  - организация памяти – фон Неймана (принстонская) или гарвардская,
  - возможность расширения ресурсов за счёт подключения внешней памяти – предусмотрена или отсутствует;
- *форма входной и выходной информации* – только цифровая, цифровая и аналоговая;
- *внешние связи* – универсальные линии ввода/вывода, программируемые пользователем, набор стандартных интерфейсов;
- *тип памяти программ* – постоянная, однократно программируемая, перепрограммируемая (ЭСППЗУ);
- *фирменные платформы* – Atmel, Microchip, Motorola, TexasInstr., Philips, Fujitsu, Samsung (описание платформ см.: [Рюмик 2010]).

Основу парка микроконтроллеров в настоящее время составляют 8-разрядные МК. Современные 8-разрядные МК отличаются низкой стоимостью, достаточно высокой производительностью, возможностью выбора разнообразной по объёму и типу памяти, широким набором аппаратных средств на кристалле (параллельные порты ввода/вывода, последовательные интерфейсы, таймеры/счётчики, аналоговые компараторы, АЦП, ЦАП, ШИМ и другие специализированные модули). Следует отметить, что, на наш взгляд, знакомство с устройством и особенностями применения МК целесообразно начинать с 8-разрядных моделей.

Наиболее популярными в настоящее время являются семейства 8-разрядных МК [Рюмик 2010]:

- MCS-51 – AT89Cх051, AT89C5х, AT89S (Atmel), DS89 (Maxim/Dallas);
- AVR – ATtiny, AT90S, ATmega, ATXmega (Atmel);
- PIC – PIC10, PIC12, PIC16, PIC18 (Microchip).
- SX – SXxxx (Uvicom, ранее Scenix);
- 68HC – 68HC08, 68HC12 (Freescale Semiconductor, ранее Motorola);
- ST – ST62, ST7 (STMicroelectronics, ранее SGS-THOMSON);
- CIP-51 – C8051 (Silicon Laboratories, ранее Cygnal Integrated);
- 8052 – W78E516 (Winbond);
- «ТЕСЕЙ» – KP1878BE1 («Ангстрем»).

В процессе изучения микроконтроллеров и проектирования систем на их основе приходится использовать как аппаратные, так и программные средства. В качестве аппаратных средств используются специальные учебные лабораторные стенды и промышленные «комплексы развития», «прототипные комплекты» и т.п., в качестве программных средств – специальные учебные программные модели МК и/или промышленные кроссотладочные средства.

В работах А. П. Жмакина [2008, 2010] показана целесообразность использования программных моделей учебных ЭВМ в учебном процессе и сформулированы основные дидактические требования, предъявляемые к ним.

Изучение микроконтроллеров, на наш взгляд, может быть построено по следующей дидактической схеме:

1-й этап – изучение особенностей процессора микроконтроллера;

2-й этап – изучение функциональных возможностей внешних устройств;

3-й этап – построение систем на базе МК и изучение связей кристалла с объектом управления.

Первый этап включает рассмотрение вопросов архитектуры процессорного ядра МК, его системы команд, взаимодействия с различными типами памяти, приобретения навыков низкоуровневого программирования (Ассемблер).

На втором этапе целесообразно рассмотреть функциональные возможности внешних (по отношению к ядру) устройств, расположенных на кристалле МК, программирование их режимов, организацию взаимодействия с ядром, в том числе с использованием подсистемы прерываний.

Заключительный этап касается вопросов организации связи кристалла МК с внешней средой (объектом управления).

Модель, предназначенную для решения вопросов первого этапа обучения микроконтроллерам, можно рассматривать как базовую. Она должна включать в себя:

- модель процессорного ядра МК на уровне программно-доступных объектов;
- модель памяти команд (отображение и загрузка при компиляции программы);
- модель памяти данных (отображение и редактирование), причём в учебной модели не обязательно моделировать весь объём памяти МК;
- средства редактирования текстов на Ассемблере, компилятор и загрузчик кода учебных программ в память модели;
- отладчик, позволяющий запускать программу в автоматическом или пошаговом режиме, с возможностью установки/сброса точек останова.

Расширения этой базовой модели позволят решать вопросы последних двух этапов дидактической схемы и должны обязательно включать модель подсистемы прерывания.

Рассмотрим пример создания программной модели микроконтроллера. В качестве объекта моделирования выбран контроллер семейства «ТЕСЕЙ» – КР1878ВЕ1, интересный прежде всего с методической точки зрения. Действительно, особенности архитектуры процессорного ядра позволяют продемонстрировать управление сегментами памяти, использовать разнообразные способы адресации (в том числе возможность доступа в память программ для извлечения данных), использование «окна интерфейса» для доступа в адресное пространство ЭСППЗУ и другие интересные решения [КР1878ВЕ1 2010].

Базовая модель МК КР1878ВЕ1 включает в себя модель процессорного ядра и памяти. В созданной программной модели они реализованы с помощью стандартных графических элементов, которые иллюстрируют текущее состояние программно-доступных объектов ядра МК.

На рисунке 1 показан вид окна, в котором отображается текущее состояние виртуальной машины: счетчик команд и стек адресов возврата, регистр флагов, служебные регистры процессора. Кроме того, в этом окне отображаются активные сегменты памяти данных и регистры портов ввода/вывода. Изменение этих значений возможно в результате выполнения команд программы, написанной пользователем.



Рис. 1. Окно состояния процессора

Помимо модели процессорного ядра, реализованы модели памяти команд (рис. 2) и памяти данных (рис. 3).

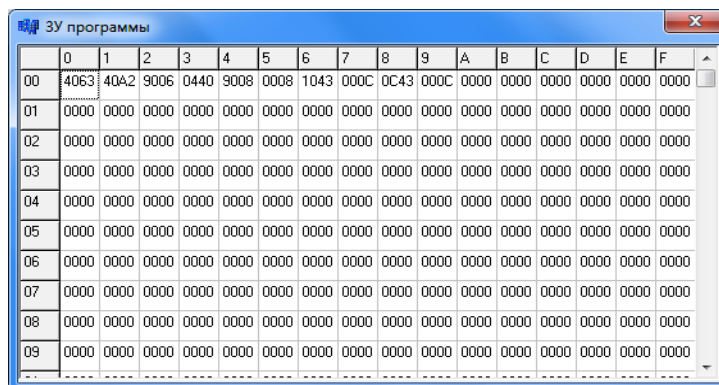


Рис. 2. Память команд

Память команд предназначена для хранения программы пользователя. Длина всех команд процессора КР1878ВЕ1 – 16 бит [КР1878ВЕ1 2010], поэтому содержимое ячеек в памяти команд отображается 4-разрядным шестнадцатеричным числом (в промежутке от 0x0000 до 0xFFFF). Код программы (заполнение таблицы памяти команд) генерируется моделью автоматически. Пользователь не может редактировать содержимое ячеек памяти команд.

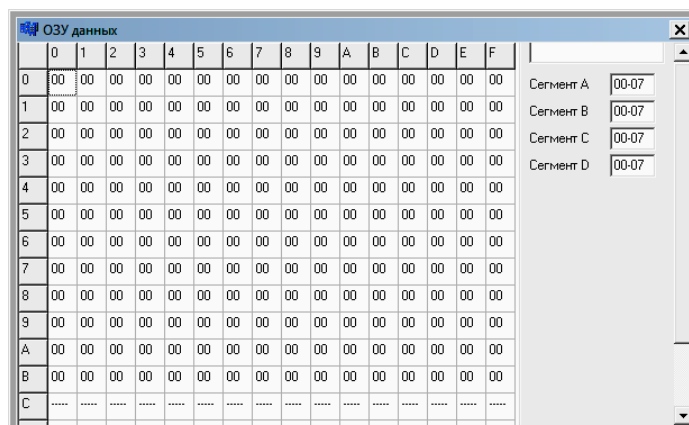


Рис. 3. Память данных

Память данных используется как место хранения результатов различных арифметических операций. Размер ячейки этой памяти составляет 1 байт (что соответствует размеру ячейки памяти данных микро контроллера КР1878ВЕ1). В каждую ячейку может быть записано двухразрядное шестнадцатеричное число от 0x00 до 0xFF. Пользователь имеет возможность вносить любые корректные изменения в эти ячейки.

«Все операнды команд размещаются в едином адресном пространстве памяти данных. Размерность этого пространства в микроконтроллере КР1878ВЕ1 – 256 байт... Команда имеет доступ к 32 байтам этого пространства через четыре сегмента адресации (сегмент А, сегмент В, сегмент С, сегмент D) размером по 8 байт. Каждому сегменту соответствует регистр адреса сегмента, значение которого определяет местонахождение данного сегмента в пространстве адресации памяти данных» [КР1878ВЕ1 2010: 8] и отражается в соответствующих ячейках окна памяти данных программной модели.

На рисунке 4 показано окно текстового редактора, предназначенного для подготовки программ на языке мнемкокодов МК «Тесей». В учебной модели реализован синтаксический анализатор кода программы (синтаксис подобен синтаксису языка Ассемблера), который должен быть исполнен на данной виртуальной машине. Проверка на корректность позволяет определить команды, введенные с нарушением установленных правил.

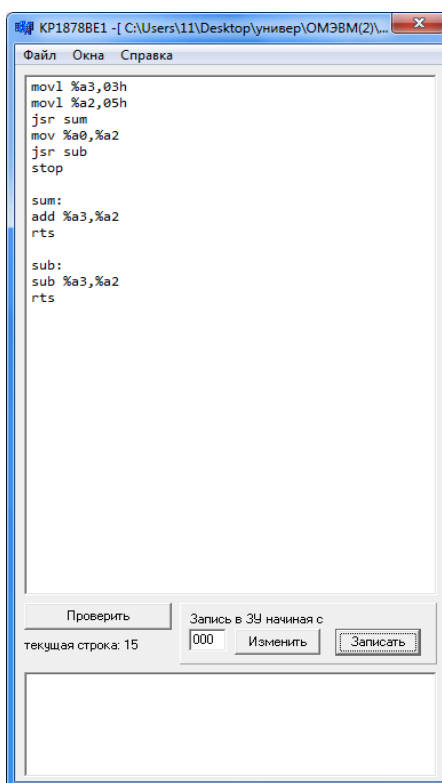


Рис. 4. Главное окно

Текст программы, написанный на языке мнемкокодов МК, вводится в верхнюю часть окна. С помощью кнопки «Проверить» запускается процедура синтаксического анализа программы. Результат проверки – номера ошибочных строк и тип ошибки, найденной в результате работы синтаксического анализатора, – размещается в нижней части окна. В случае отсутствия ошибок формируется сообщение об успешном

прохождении компиляции и выполняется запись кодов в память команд. Задача генерации и записи этих кодов в память решается встроенным в виртуальную машину компилятором. Загрузка в память начинается с адреса, который будет введен с специально отведенное для этого поле (по умолчанию этот адрес равен 000).

Для отладки программы и иллюстрации её работы создан отладчик (рис. 5), который позволяет выполнять трассировку в пошаговом режиме, создавать и удалять контрольные точки и запускать программу в автоматическом режиме. Строка таблицы в окне отладчика включает адрес выполняемой команды, мнемокод команды и её машинный код. В первом столбце таблицы предусмотрена возможность установки и удаления контрольной точки.

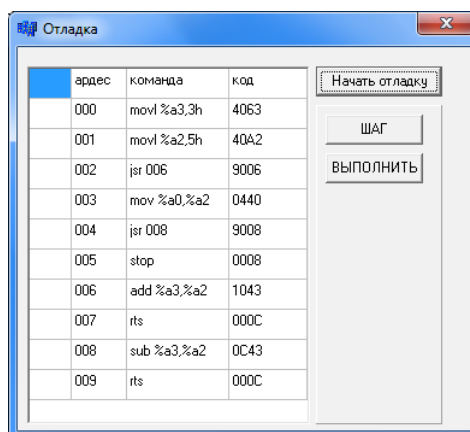


Рис. 5. Окно отладчика программ

Расширение базовой модели возможно за счёт моделирования внешних устройств, расположенных на кристалле МК. Например, порты параллельного обмена реализуются как обычные регистры, а возможность редактирования состояния порта определяется его настройкой. В учебных моделях отображается только логическое состояние выводов в статике, при этом игнорируются проблемы нагрузочной способности, динамических свойств линий. Аналогично можно моделировать и линии последовательных интерфейсов.

При моделировании таймеров/счётчиков следует помнить, что они будут оперировать условными единицами времени, длительность которых, возможно, будет изменяться в зависимости от характеристик ПЭВМ, на которой запускается модель.

При моделировании АЦП или ЦАП «аналоговая» информация будет представлена в цифровой форме, которую на экране можно проиллюстрировать положением движка на «непрерывной» линейке.

В заключение отметим, что предложенная классификация, дидактическая модель обучения и программная модель микроконтроллера КР1878ВЕ1 могут успешно применяться в процессе преподавания дисциплин «Архитектура ЭВМ», «Системы реального времени» и «Системное программирование».

### **Библиографический список**

*Жмакин А.П.* К вопросу проектирования программных моделей ЭВМ // Информационные системы: Теория и практика: сб. науч. работ фак. информатики и вычислит. техники Курск. гос. ун-та. Курск: Курск. гос. ун-т, 2010. С. 55–62.

*Жмакин А.П.* Программные модели устройств ЭВМ // Вестник Московского городского педагогического университета. 2008. № 6 (16). С. 99–103.

*KP1878BE1* 8-разрядный RISC микроконтроллер [Сайт]. URL: <http://www.angstrem.ru> (дата обращения: 10.12.2010).

*Рюмик С.М.* 1000 и одна микроконтроллерная схема. Вып. 1. М.: Додэка-XXI, 2010. 356 с.