

КРОСС-ПЛАТФОРМЕННЫЙ ЭМУЛЯТОР МИКРОПРОГРАММИРУЕМОЙ ЭВМ

*Николай Николаевич Коннов¹, Алексей Ильич Пчелинцев²,
Вячеслав Дмитриевич Федулов³, Ольга Викторовна Юрова⁴*

^{1,2,3,4} Пензенский государственный университет, Пенза, Россия

^{1,2,3,4} knnpnz@mail.ru

Аннотация. Приводится описание кросс-платформенного эмулятора 16-разрядной ЭВМ с микропрограммным управлением, предназначенной для изучения функционирования и проектирования микропроцессорных систем на микропрограммном уровне в учебном процессе подготовки бакалавров по направлению «Информатика и вычислительная техника».

Ключевые слова: архитектура, процессор, эмулятор, учебный процесс

Подготовка современного специалиста в области вычислительной техники предполагает изучение архитектуры процессоров на двух уровнях [1].

Ассемблерном, когда особенности функционирования процессора можно изучать по правилам формирования программно доступных регистров и ячеек оперативной памяти на реальной аппаратной платформе с использованием стандартных средств низкоуровневого программирования.

На более низком микропрограммном, когда надо рассматривать особенности имплементации команд процессора, реализуемой как последовательность межрегистровых передач (фазы выборки и дешифрации команд, выборки операндов, исполнения команды, записи результата и др.). Микропрограмма выполняется внутри кристалла современного процессора, что не позволяет студенту ее контролировать или изменять, Это вызвало необходимость создания программ – эмуляторов.

На кафедре вычислительной техники ПГУ были разработаны и активно применяются в лабораторном практикуме и курсовом проектировании DOS и Windows версии программных симуляторов шестнадцатиразрядного процессора с микропрограммным управлением, использующего логическую организацию некогда популярного разрядно-модульного комплекта серии K1804 [2–4]. Программы состоят из программно-логических моделей универсального шестнадцатиразрядного микропрограммируемого процессора с разрядно-модульной организацией, базирующейся на БИС K1804, оперативной памяти с емкостью 64Кбайт и шириной выборки шестнадцать разрядов, а также компоненты графического интерфейса.

Использование программы эмулятора в лабораторном практикуме и курсовом проектировании дисциплин, связанных с изучением архитектуры, позволяет студентам разрабатывать различные алгоритмы выполнения арифметических операций с фиксированной и плавающей точкой, выполнять микропрограммную

имплементацию шестнадцатиразрядных ЭВМ с различной процессорной архитектурой (наиболее просто реализуется архитектура PDP-11).

Однако принятые правительственные решения о переходе органов власти и бюджетных учреждений, в том числе и вузов, на импортозамещенное используемое ПО заставляет вести работу по переводу учебных программ на отечественное либо свободно распространяемое ПО.

Авторами была разработана новая кроссплатформенная версия эмулятора микропрограммируемой ЭВМ, которая позволяет сохранить существующие методические наработки при отказе от ОС ПО Microsoft.

Для реализации поставленных задач был выбран объектно-ориентированный язык программирования Java [5]. Отличительными особенностями Java является: платформонезависимость, архитектурная-нейтральность, интерпретируемость, высокая производительность, динамичность и адаптируемость. Для реализации графического интерфейса был выбран набор графических компонентов Swing, который в Java является частью базового класса и не зависит от платформы, используется для создания оконных приложений.

В данной программе реализовано восемь классов, каждый из которых отвечает за свою область в программе и имеет индивидуальный набор методов.

iTact – класс, реализующий окно эмулятора. Главный класс, с которого начинается выполнение программы, создает объекты других пользовательских классов и обеспечивает взаимодействие между ними.

Поля данного класса: объекты классов Listing, EditPzu, Protocol, gr1804 и поля, которые обеспечивают изменения внешних сигналов эмулятора, блокировку пользовательского интерфейса и форматированного ввода значений в регистры эмулятора.

Методы класса: main – это главный метод класса точка входа в приложение; конструктор класса, в котором инициализируются поля, создаются объекты классов, устанавливаются исходные значения регистров эмулятора. Также содержит методы, которые выполняют: установку блокировки и ее отключение на компоненты пользовательского интерфейса; вывод значений регистров на экран; вывод адреса старта и завершения отладки, а также ограничения количества тактов; установку точки останова и цветовой темы приложения; обновление данных регистров, перед запуском эмуляции; подготовку данных к выполнению микрокоманд; запуск отладки; сохранение микропрограммы в файл; чтение микропрограммы в массив из файла.

Gr1804 – класс, реализующий программно-логическую модель микропроцессора и взаимодействие основного класса с ядром. Поля: t – объект класса Tact. Методы класса: в конструкторе класса инициализируется объект t класса Tact и устанавливает начальные значения для эмулятора микропроцессора. Содержит методы, которые выполняют: извлечение микрокоманды из массива микрокоманд; сброс регистров микропроцессора в исходное значение; эмуляция микропроцессора; сохранение адреса текущей команды; проверка на совпадение с адресом завершения или с точками останова; проверка количества тактов моделирования; выполнение эмуляции в классе Tact; возвращение результата моделирования.

Tact – класс, реализующий ядро микропроцессора. Полей данный класс не содержит. Методы класса выполняют преобразование типов данных, арифмети-

ческие и логические операции, формирование флагов, обработку определённых битов, а также определяют уровень команды и эмулируют работу блока микропрограммного управления и блока обработки данных

Protocol – класс, реализующий протокол отладки микропрограммы. Поле класса – счетчик тактов для протокола. Методы класса выводят начальную информацию об отладке в текстовое поле, сохраняют очередную микрокоманду и состояние регистров в текстовое поле.

Listing – класс, реализующий листинг микропрограммы. Полей данный класс не содержит. Методы данного класса выводят начальную информацию об отладке в текстовое поле, сохраняют очередную микрокоманду и состояние регистров в текстовое поле.

EditPzu – класс, реализующий редактирования микропрограммы. Поля данного класса – модель таблицы, с помощью которой происходит изменение значений в ячейках, а также их проверку на допустимые значения. Методы данного класса выполняют инициализацию компонентов формы, получение модели таблицы и установка шрифтов и цветовой темы, вывода в таблицу микропрограммы

MemoryForm – класс, реализующий эмулятор оперативной памяти ЭВМ в виде таблицы, с возможностью редактирования, сохранения и загрузки. Поля данного класса – модель таблицы, с помощью которой происходит изменение значений в ячейках, а также их проверку на допустимые значения. Методы данного класса выполняют инициализацию компонентов формы, получение модели таблицы и установка шрифтов и цветовой темы, вывода в таблицу данных ячеек памяти, установку значений ячеек по умолчанию и очищение всех ячеек, сохранение в файл и загрузка из файла данных таблицы памяти. Также реализованы методы для получения значения из ячейки памяти и для записи значения в ячейку памяти по указанному адресу по указанному адресу.

Окна программы представлены на рис. 1.

Главный экран содержит поля регистров и флагов процессора, для отслеживания текущей информации в них. При необходимости информацию в них можно изменять вручную. После изменения она сохранится. Меню состоит из трех пунктов: Файл включает в себя функции сохранения и загрузки программы; пункт отчеты содержит окна необходимые для составления отчета по выполненной программе – Протокол и Листинг; пункт Тема позволяет выбрать пользователю интерфейс. Ниже располагаются кнопки управления программой: редактирование полей микрокоманды МК1–МК15, оперативной памяти, запуска и отладки микропрограммы в автоматическом режиме, отладки в потактовом режиме.

Слева располагаются блок для ручной установки флагов FLG, подаваемых в процессор от внешней аппаратуры (готовности ОП и периферии, прерывания, останова и др.), блок добавления точек останова. В самом низу выводится информация о выполненной микрокоманде и микрокоманде, которая будет выполняться следующим тактом.

Окно редактирования микропрограммы представляет память микропрограмм в виде таблицы, с возможностью изменения любого поля любой МК. Микрокоманда включает 64 разряда и разбита на 15 полей, задающих микрооперации.

Окно работы с памятью выводит содержимое оперативной в виде таблицы и содержит кнопки очищения памяти, сохранения и загрузки. Для удобства поль-

зователя имеется подсвечивание цветами полей, которые не равны нулю. При наведении на любую ячейку таблицы внизу будет выведен ее адрес, так же если адрес известен заранее введя число в данную ячейку можно произвести поиск.

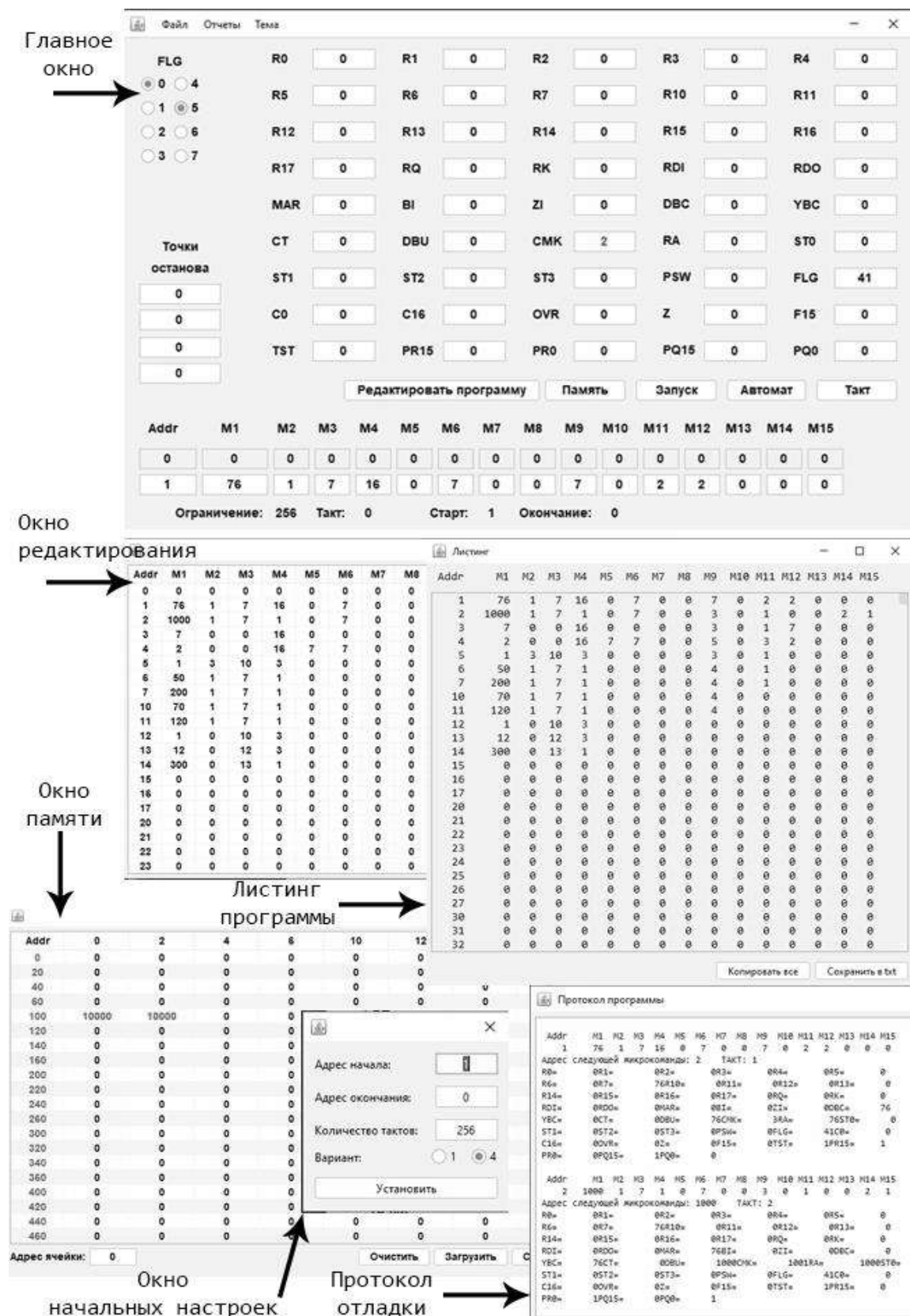


Рис. 1. Рабочие окна программы

Окно запуска отвечает на настройку начальных параметров. Имеет поля для ввода адреса начала и окончания моделирования, задания количества тактов моделирования. Размеры памяти микрокоманд могут быть установлены 256, 512, 1024, 2048, 4096 микрокоманд, что позволяет реализовывать весьма сложные микропрограммы.

В окне протоколирования выводится информация о выполненной МК и текущей информации, хранящейся в регистрах.

В окне листинга выведена каждая МК. Внизу располагаются кнопки копирования и сохранения в текстовом виде.

Программа эмулятора разработана в пакете NetBeans 12.2 [6] и может исполняться в операционных средах Windows 7/8/10 и Linux Ubuntu/Astra/Альт Образование/Manjaro с установленной средой выполнения Java Runtime Environment (версия не менее 1.8.0). Требуемый объем оперативной памяти не менее 256 Мбайт, размер программы – 2 Мбайт. Программа получила свидетельство о государственной регистрации программ для ЭВМ [7].

Дальнейшее развитие программы будет направлено на интеграцию с графическим редактором микропрограммы обеспечивающий независимость этапов подготовки и отладки, оптимизацию программного кода.

Список литературы

1. Таненбаум Э. Архитектура компьютера. СПб. : Питер, 2018. С. 816.
2. Гурин Е. И., Коннов Н. Н., Механов В. Б. Проектирование процессора ЭВМ на секционированных микропроцессорных БИС : метод. указания к курсовому проектированию. URL: <http://window.edu.ru/resource/316/66316>
3. Коннов Н., Филин С. Е., Вирясов П. П., Барабаш К. О. Учебный программный эмулятор ЭВМ с микропрограммным управлением // Университетское образование (МКУО–2014) : сб. ст. XVIII Междунар. науч.-метод. конф., посвящ. 200-летию со дня рождения М. Ю. Лермонтова (г. Пенза, 10–11 апреля 2014 г.) / под ред. А. Д. Гулякова, Р. М. Печерской. Пенза : Изд-во ПГУ, 2014. С. 183–184.
4. Коннов Н., Исаков Н. В., Прокуров Д. А. Программный симулятор ЭВМ с микропрограммным управлением // Актуальные вопросы современной науки: теория и практика научных исследований : сб. науч. ст. II Всерос. науч.-практ. конф. (Пенза, 1–15 ноября 2018 г.). Пенза : Изд-во Пенз. гос. технол. ун-та, 2018. С. 24–27.
5. Шилдт Г. Java 8. Полное руководство. М. : Питер : Вильямс, 2015. 816 с.
6. Монахов В. В. Язык программирования Java и среда NetBeans : учеб. пособие. СПб. : БХВ-Петербург, 2011. С. 703.
7. Свидетельство о государственной регистрации программы для ЭВМ № 2021662392. Эмулятор микропрограммируемой ЭВМ «Такт-3» / А. И. Пчелинцев, Н. Н. Коннов, В. Д. Федулов, О. В. Юрова. Заявка : 2021661545 от 22.07.2021 № 2012618426 (дата регистрации: 27.07.2021).