

УДК 004.94

DOI: 10.18413/2518-1092-2024-9-3-0-4

Шатохин М.С.  
Жихарев А.Г.  
Гапицонов И.Ю.

СИСТЕМНО-ОБЪЕКТНОЕ ИМИТАЦИОННОЕ  
МОДЕЛИРОВАНИЕ МИКРОКОНТРОЛЛЕРА ATMEL  
ATMEGA 168

Белгородский государственный национальный исследовательский университет,  
ул. Победы, 85, г. Белгород, 308015, Россия

*e-mail: zhikharev@bsu.edu.ru*

#### Аннотация

В работе представлен процесс разработки имитационной модели, включающий описание функциональности микроконтроллера, определение основных параметров модели, а также выбор инструментов для ее создания. В работе представлено подробное описание разработанной имитационной модели, включая ее архитектуру, основные компоненты и интерфейсы. Полученные результаты представляют собой ценный вклад в области образования, предоставляя учебные материалы и инструменты для обучения студентов работе с микроконтроллером Atmel ATmega 168.

**Ключевые слова:** модель; моделирование; микроконтроллер; имитационное моделирование

**Для цитирования:** Шатохин М.С., Жихарев А.Г., Гапицонов И.Ю. Системно-объектное имитационное моделирование микроконтроллера Atmel ATmega 168 // Научный результат. Информационные технологии. – Т.9, №3, 2024. – С. 34-42. DOI: 10.18413/2518-1092-2024-9-3-0-4

Shatokhin M.S.  
Zhikharev A.G.  
Gapitsonov I.Yu.

SYSTEM-OBJECT SIMULATION OF THE ATMEL  
ATMEGA 168 MICROCONTROLLER

Belgorod State National Research University,  
85 Pobedy St., Belgorod, 308015, Russia

*e-mail: zhikharev@bsu.edu.ru*

#### Abstract

The paper presents the process of developing a simulation model, including a description of the functionality of the microcontroller, determination of the main characteristics and parameters of the model, as well as the selection of tools for its creation. The paper presents a detailed description of the developed simulation model, including its architecture, main components and interfaces. The results provide a valuable contribution to the field of education by providing educational materials and tools for teaching students how to use the Atmel ATmega 168 microcontroller.

**Keywords:** system model; modeling; microcontroller; simulation

**For citation:** Shatokhin M.S., Zhikharev A.G., Gapitsonov I.Yu. System-object simulation of the Atmel ATmega 168 microcontroller // Research result. Information technologies. – Т. 9, №3, 2024. – P. 34-42. DOI: 10.18413/2518-1092-2024-9-3-0-4

#### ВВЕДЕНИЕ

В современном мире микроконтроллеры играют ключевую роль во множестве устройств и систем, обеспечивая их функциональность и автоматизацию. Одним из наиболее популярных микроконтроллеров на рынке является Atmel ATmega 168, который широко используется в различных областях, включая электронику, автоматизацию, робототехнику и другие.

Целью данной работы является проектирование имитационной модели микроконтроллера Atmel ATmega 168 в образовательных целях.

Имитационная модель позволяет создать виртуальное представление реального устройства, которое может быть использовано для обучения студентов работе микроконтроллера без необходимости физического присутствия самого микроконтроллера. В рамках работы будут рассмотрены основные характеристики микроконтроллера ATmega 168, его архитектура и функциональные возможности. Затем будет разработана имитационная модель, которая будет включать в себя эмуляцию памяти, регистров и других компонентов микроконтроллера. Для создания имитационной модели будет использована среда разработки UFOmodeler. В результате выполнения работы будет создана имитационная модель микроконтроллера Atmel ATmega 168, которая позволит преподавателям и студентам изучать работу микроконтроллера в виртуальной среде. Это значительно упростит и ускорит процесс обучения и позволит сэкономить ресурсы и время.

### ТЕОРИЯ ATMEL ATMEGA 168

Atmel ATmega 168 – один из наиболее популярных микроконтроллеров на рынке. Он широко используется в различных областях, таких как электроника, автоматизация и робототехника. ATmega 168 имеет множество функциональных возможностей, которые позволяют разработчикам создавать разнообразные устройства и системы.

Архитектура микроконтроллера ATmega 168 основана на архитектуре Harvard с модификациями, специфичными для данной модели. Он имеет 8-битный AVR-процессор, что обеспечивает высокую производительность и быструю обработку данных. Микроконтроллер также оснащен различными периферийными устройствами, такими как аналого-цифровые преобразователи (АЦП), таймеры, интерфейсы ввода-вывода и другие [1].

Одной из особенностей ATmega 168 является наличие встроенной флэш-памяти размером 16 Кб, которая используется для хранения программного кода. Это позволяет разработчикам загружать и изменять программы на микроконтроллере без необходимости использования внешних устройств. Кроме того, микроконтроллер имеет EEPROM-память размером 512 байт для хранения постоянных данных и SRAM-память размером 1 Кб для временного хранения данных. На рисунке 1 представлена схема микроархитектуры Atmel ATmega168 [2].

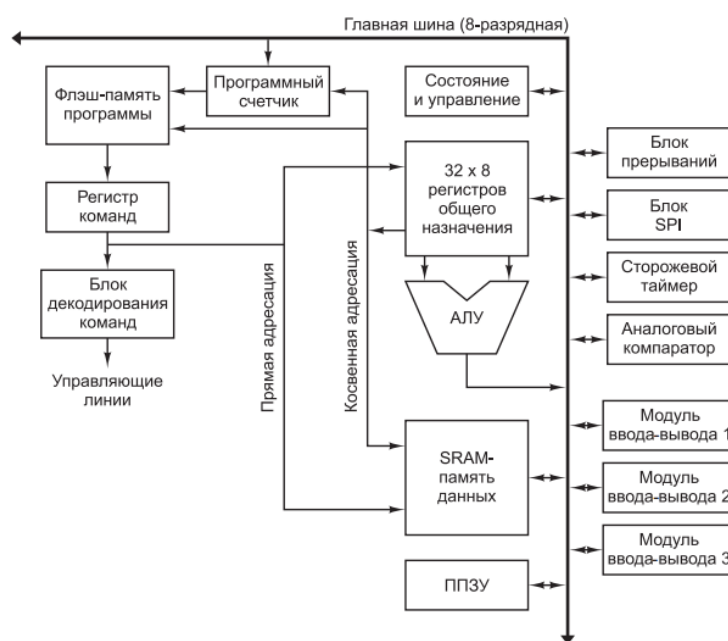


Рис. 1. Схема микроархитектуры Atmel ATmega168  
Fig. 1. Atmel ATmega168 microarchitecture diagram

Центральное положение в микроархитектуре ATmega168 занимает 8-разрядная основная шина. С ней связаны регистры и биты состояния, АЛУ, память и устройства ввода-вывода. Регистровый файл состоит из 32 8-разрядных регистров, используемых для хранения временных значений. В регистре состояния и управляющем регистре содержатся признаки последней операции АЛУ (знак, переполнение, отрицательность, нуль, перенос), а также бит незавершенного прерывания. Программный счетчик содержит адрес команды, выполняемой в настоящий момент. Чтобы выполнить операцию АЛУ, необходимо сначала прочитать операнды из регистров и передать их АЛУ. Выходные данные АЛУ записываются в любые регистры с возможностью записи через главную шину [1].

### ПОСТРОЕНИЕ МОДЕЛИ

Как было отмечено выше, в качестве технологии построения модели в работе используются инструменты теории системно-объектного моделирования [3], в частности, программный инструмент UFOModeler [4, 5].

Для проектирования имитационной модели [6, 7] микроконтроллера ATmega168 необходимо учесть все его особенности и функциональные возможности. Модель должна включать в себя эмуляцию периферийных устройств, памяти, регистров и других компонентов микроконтроллера. Она должна быть способна выполнять программный код и обрабатывать данные так же, как и реальный микроконтроллер.

Начать создание модели стоит непосредственно с узла [7, 8] “ATmega168”, к нему присоединяется узел “Терминал управления” для непосредственного управления микроконтроллером и узел “Преобразователь данных” для контроля результатов его работы.

“Терминал управления” передает 16 битные инструкции в “ATmega168”, тактовый генератор контролирует очередность выполнения инструкций. Структура инструкций принимает следующий вид:

Структура команды	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
команда	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	упр сиг				адрес блока памяти				адрес регистра				адрес команды			

Рис. 2. Структура инструкции тип 1

Fig. 2. Instruction structure type 1

Структура команды	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
команда	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	упр сиг				конкретное значение								адрес команды			

Рис. 3. Структура инструкции тип 2

Fig. 3. Instruction structure type 2

“Преобразователь данных” принимает результат работы микроконтроллера, после чего он преобразует его из двоичной системы счисления в десятичную, для контроля правильности результата.

Далее по схеме рисунка 1 декомпозируем узел ATmega168, для упрощения модели реализуем основные компоненты микроконтроллера:

- Модули ввода вывода;
- Шина;
- Программный счетчик;
- Флэш-память;
- Регистр команд;
- Блок декодирования команд;
- 32x8 регистров общего назначения;

- АЛУ;
- SRAM-память (для упрощения объединена с EEPROM).

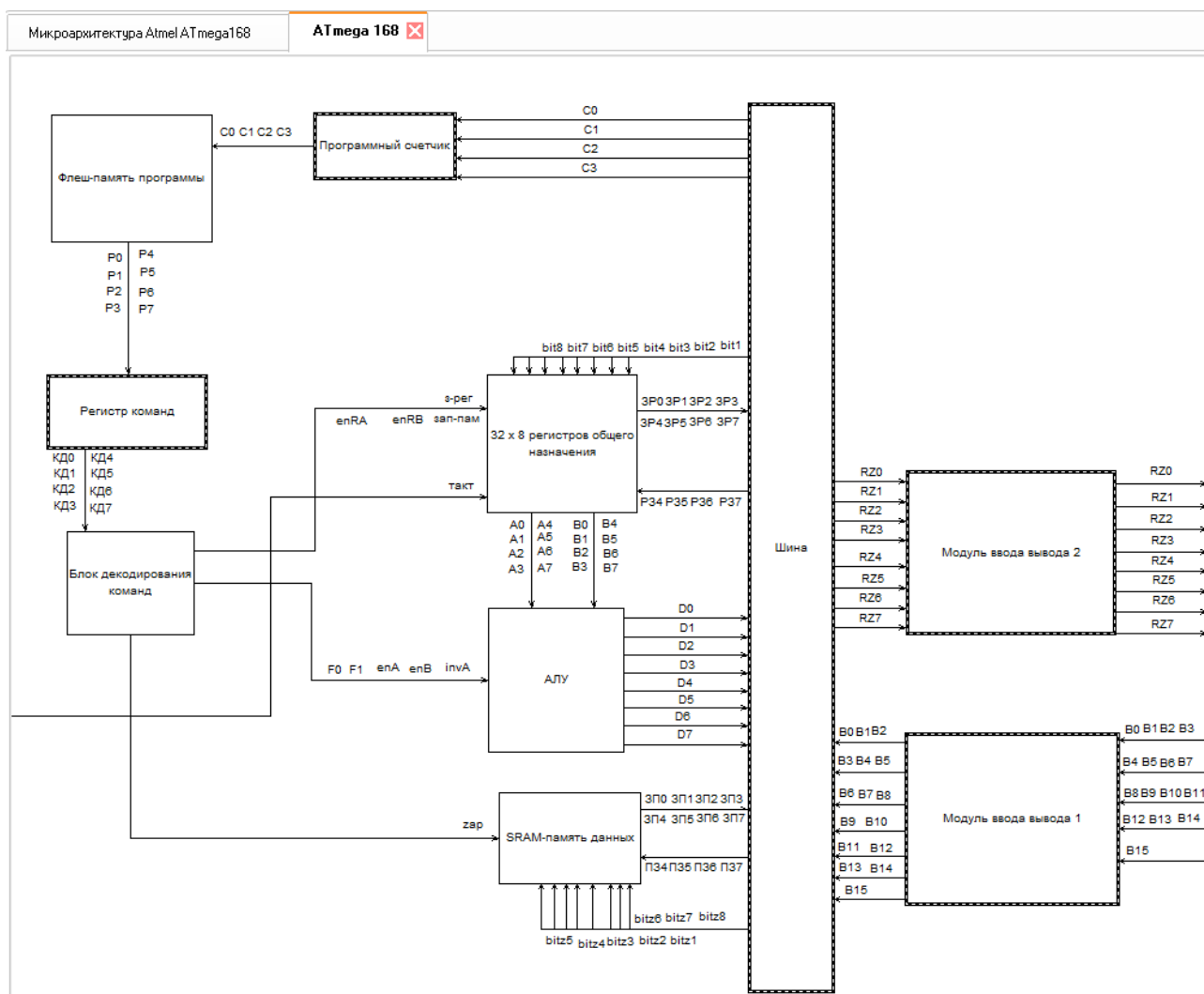


Рис. 4. Общий вид модели ATmega168  
Fig. 4. General view of the ATmega168 model

“Модуль ввода вывода 1” принимает инструкции извне через связи B0-B15 и передает их в “Шина”. “Модуль ввода вывода 2” принимает результат работы из “Шина” и передает их в “Преобразователь данных”. “Шина” передает между узлами значения в зависимости от инструкций. “Программный счетчик” принимает по связям C0-C3 адрес команды и передает его во “Флэш-память”. “Флэш-память” в данной модели содержит десять 8-битных ячеек памяти. Каждая ячейка имеет свой адрес, и исходя из входных значений C0-C3 включится один из 10 блоков памяти и передаст на “ВЫХОД” команду, которую хранит в себе. Адреса блоков памяти представлены на рисунке 5.

Флэш-память		b3	b2	b1	b0
Блок памяти 1	загрузка в регистр	0	0	0	1
Блок памяти 2	Сложение с переносом	0	0	1	0
Блок памяти 3	Логическая операция И	0	0	1	1
Блок памяти 4	Логическая операция ИЛИ	0	1	0	0
Блок памяти 5	Логическая операция ИСКЛЮЧАЮЩЕЕ ИЛИ	0	1	0	1
Блок памяти 6	Инкремент	0	1	1	0
Блок памяти 7	Декремент	0	1	1	1
Блок памяти 8	Вычитание с переносом	1	0	0	0
Блок памяти 9	сохранение в sram	1	0	0	1
Блок памяти 10	Сравнение	1	0	1	0
		адрес команды			

Рис. 5. Адреса блоков памяти

Fig. 5. Memory block addresses

“Блок памяти 1” принимает в переменные k0-k3 значение адреса и когда произведение этих переменных приравняется к 1, то на выход пойдет команда, которая в нем записана.

“Регистр команд” принимает команду из “Флэш-памяти” и передает её на узел “Блок декодирования команд” для дальнейшей обработки.

“Блок декодирования команд” принимает из “Регистр команд” команду, которую декодирует и превращает в управляющие линии, каждая команда включает свой определенный набор управляющих линий, которые включают или выключают “Регистры”, “АЛУ” и “SRAM”. В данной модели закодированы 3 команды: Загрузка в регистр, Сложение с переносом, Сохранение в SRAM. Данное решение обусловлено тем, что для понимания работы модели этого вполне достаточно, но при необходимости можно увеличить набор команд до десяти и более.

Набор управляющих линий:

1. Управляющие SRAM-памятью:
  - Zar – запись новых значений.
2. Управляющие Регистрами:
  - enRA – открыть шину АЛУ A;
  - enRB – открыть шину АЛУ B;
  - зап-пам – запись из регистра в SRAM память;
  - з-рег – запись в регистры.
3. Управляющие АЛУ:
  - F0 – выбор действия АЛУ;
  - F1 – выбор действия АЛУ;
  - enA – открыть вход в АЛУ A;
  - enB – открыть вход в АЛУ B;
  - invA – инвертировать значение A.

“Упр ком 1 загр” (Управление командой 1 загрузка вложенный узел узла “Блок декодирования команд”) блок активируется аналогично “Флэш-памяти” и включает управляющие линии (F0,F1...) необходимые для выполнения данной команды, остальные управляющие линии выключаются.

“SRAM-память” в данной модели представляет собой пять 8-битных блоков памяти, которые хранят в себе значения и могут быть перезаписаны. Работает аналогично “Флэш-памяти”, исходя из пришедшего адреса активируется один из блоков и передает хранившееся в нем значение на “Шина”.

В узел “Блок памяти 1” узла “SRAM-память” запись нового значения осуществляется, когда приходит управляющий сигнал zar и адрес блока, в который необходимо записать значение.

“32x8 регистров общего назначения” для примера работы реализованы 4 самостоятельных регистра из 32. Работают они следующим образом, на вход поступает 8 битное значение и в

зависимости от адреса и сигнала управления значение запишется в один из регистров, после так же в зависимости от сигналов управления значение попадет на шины АЛУ или же запишется в SRAM-память.

“Регистр 1” принимает значение с шины и сохранит его в регистре если придет сигнал управления адреса и записи в регистр, если адрес придет в значении 0 передаст 8 битное слово в следующий регистр, тот в свою очередь действует также, и в зависимости от такта операции передаст слово в шину АЛУ А или В.

“Шина А” принимает значение из регистра в зависимости от сигнала управления который разрешит запись в шину в зависимости от такта операции.

“Управление” принимает значения управляющих сигналов из блока декодирования команд и передает их на регистры и шины АЛУ.

“АЛУ” принимает два 8 битных слова, передает их в однобитные АЛУ по связям А0-А7 и В0-В7, и в зависимости от значений F0 и F1 выбирает операцию логическое И, ИЛИ, неВ или арифметическая сумма А и В, связь invA инвертирует значение пришедшее через связи А, связи enA и enB открывают вход для двух 8 битных слов. Так же стоит отметить, что связи F0, F1, enA, enB, invA передаются от 1 разряда АЛУ к 8 разряду.

“АЛУ 2 – разряд” подается несколько сигналов:

1. F0, F1 управляющие сигналы задают выбор выполняемой команды:
  - 00 – лог. А\*В
  - 01 – лог. А+В
  - 10 – лог. неВ
  - 11 – сумма А+В
2. ENA, ENB разрешают вход сигналов А и В.
3. А и В непосредственно операнды.
4. INVA сигнал инверсии А по умолчанию 0, если необходимо получить неА устанавливается значение 1.

Через сигнал перенос передается бит, переходящий на следующий разряд. Далее в зависимости от поступающих сигналов АЛУ выполняет одну из 4 операций и выводит результат. Элементы АЛУ: И, ИЛИ и искл-ИЛИ работают как стандартные логические операции и не требуют пояснений.

“Декодер” работает как любой стандартный декодер принимая код операции и выдавая сигнал, который её активирует.

“Логическое устройство” принимает бит А и В из двух слов и управляющий сигнал, в зависимости от которого будет выполнена одна из операций.

“Полный сумматор” принимает бит А и В из двух слов и управляющий сигнал, в зависимости от которого будет выполнена операция или нет.

## **ТЕСТИРОВАНИЕ МОДЕЛИ**

Тестирование следует начать с описания некоторой последовательности команд или другими словами сценария, который должен будет выполнить микроконтроллер [9]. Для тестирования модели был написан сценарий из четырех инструкций которые охватят весь функционал текущей версии модели, сценарий представлен на рисунке 6.



Описание сценария

к1 - записать в регистр 2 значение блока памяти 1 (2)  
к2 - записать в регистр 3 значение 15

к3 - провести операцию сложение с переносом  
над этими значениями

к4 - сохранить результат операции в блок памяти 1  
результат операции число 17

Рис. 6. Сценарий тестирования  
Fig. 6. Test scenario

Далее перейдем к тесту модели, после запуска на рисунке 7 можем увидеть значение инструкции, которое терминал управления передаст на ATmega 168.

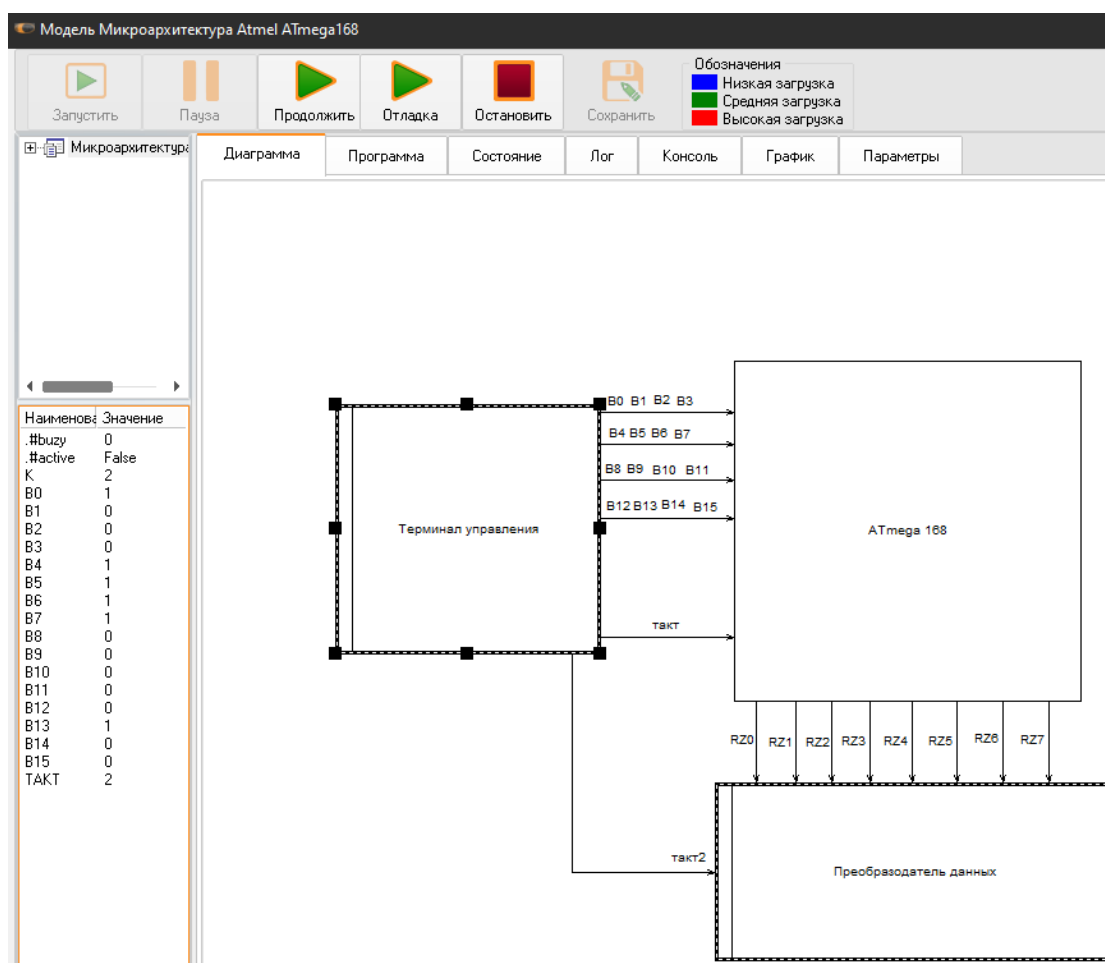


Рис. 7. Значения терминала управления  
Fig. 7. Control terminal values

Дальше инструкция разделяется и расходится на разные блоки “Программный счетчик”, “SRAM-память” и “Регистры”.

Теперь исходя из кода команды мы попадаем в блок управления который включает линии управления, которые позволяют выполнить данную команду. Далее результат выводится из ATmega 168 в Преобразователь данных для контроля правильности результата.

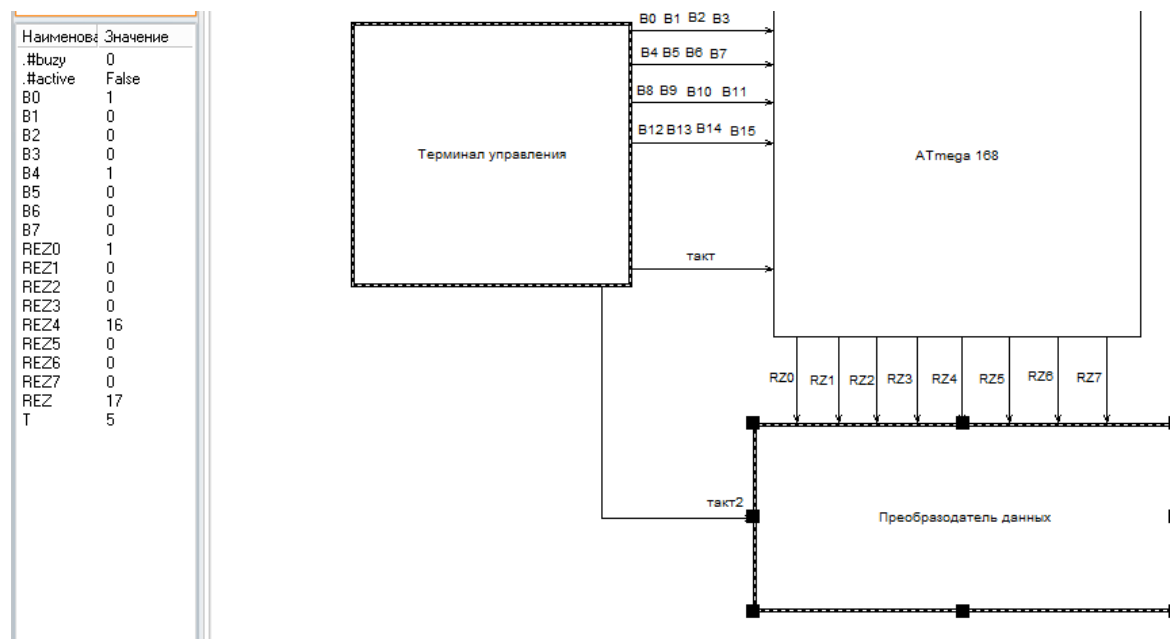


Рис. 8. Результат  
Fig. 8. Result

По результату видно, что микроконтроллер работает правильно.

### ЗАКЛЮЧЕНИЕ

На основе полученных знаний была разработана имитационная модель микроконтроллера, которая позволяет симулировать его работу с разными входными данными.

В ходе выполнения работы была создана модель микроконтроллера Atmel ATmega 168. Данная модель была спроектирована с использованием ПО UFOmodeler.

Основной целью разработки модели было получение возможности обучению принципам работы микроконтроллера ATmega 168. С ее помощью студенты могут практически освоить основные функции и возможности данного микроконтроллера, а также научиться разрабатывать и отлаживать программное обеспечение для него.

Таким образом, разработанная имитационная модель микроконтроллера Atmel ATmega 168 является полезным инструментом для тестирования программного обеспечения и обучения студентов. Она позволяет симулировать работу микроконтроллера в различных режимах и проверить правильность выполнения программы.

### Список литературы

1. Таненбаум Э., Остин Т. Архитектура компьютера 6-е издание: СПб.: Питер, 2013. 816 с.
2. Техническое описание ATMEGA168 (PDF) — Корпорация ATMEL: офиц. сайт. – URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/83753/ATMEL/ATMEGA168.html> (дата обращения: 20.11.2023). – Текст: электронный.
3. Matorin S.I., Zhikharev A.G. Accounting for system-wide regularities in the system-object modeling of organizational knowledge // Scientific and Technical Information Processing. – 2019. – Vol. 46, № 6. – P. 1-9.
4. Жихарев А.Г., Корсунов Н.И., Маматов Р.А., Щербинина Н.В., Пономаренко С.В. О разработке адаптивной образовательной платформы с использованием технологий машинного обучения // Экономика. Информатика. 2022. Т. 49. № 4. С. 810-819.
5. Deeney I. A., Zhikharev A.G., Klyuchnikov D. A., Shurukhina T. N., Gavrilova T. A. Some aspects of AI-technologies in education // Revista San Gregorio. – 2021. – Vol. 44. – P. 186-197.
6. Zhikharev A.G. Formalization of Knowledge by Tools of System-Object Simulation // Lecture Notes in Networks and Systems. – 2022. – Vol. 330. – P. 390- 399.
7. Маторин С.И., Жихарев А.Г. 2019. Системно-объектный подход как основа общей теории систем. Научные ведомости БелГУ. Сер. Экономика. Информатика, 2019, Т. 46, № 4: 717-730.



8. Маторин С.И., Жихарев А.Г. 2018. Формализация системно-объектного подхода «Узел-Функция-Объект». Прикладная информатика, 2018, Т. 13, № 3 (75): 124-135.

9. Бич, М. Микроконтроллеры семейства XC166. Вводный курс разработчика / М. Бич. – М.: ДМК, 2016. – 200 с.

### References

1. Tanenbaum E., Austin T. Computer Architecture 6th edition: SPb.: Piter, 2013.816 p.

2. Technical description ATMEGA168 (PDF) – ATMEL Corporation: official website. – URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/83753/ATMEL/ATMEGA168.html> (accessed: 20.11.2023). – Text: electronic.

3. Matorin S.I., Zhikharev A.G. Accounting for system-wide regularities in the system-object modeling of organizational knowledge // Scientific and Technical Information Processing. – 2019. – Vol. 46, No. 6. – P. 1-9.

4. Zhikharev A.G., Korsunov N.I., Mamatov R.A., Shcherbinina N.V., Ponomarenko S.V. On the development of an adaptive educational platform using machine learning technologies // Economy. Informatics. 2022. Vol. 49. No. 4. P. 810-819.

5. Deeney I. A., Zhikharev A.G., Klyuchnikov D. A., Shurukhina T. N., Gavrilova T. A. Some aspects of AI-technologies in education // Revista San Gregorio. – 2021. – Vol. 44. – P. 186-197.

6. Zhikharev A.G. Formalization of Knowledge by Tools of System-Object Simulation // Lecture Notes in Networks and Systems. – 2022. – Vol. 330. – P. 390- 399.

7. Matorin S.I., Zhikharev A.G. 2019. System-object approach as a basis for general systems theory. Scientific bulletin of BelSU. Series: Economics. Informatics, 2019, Vol. 46, No. 4: 717-730.

8. Matorin S.I., Zhikharev A.G. 2018. Formalization of the system-object approach "Node-Function-Object". Applied informatics, 2018, Vol. 13, No. 3 (75): 124-135.

9. Beach, M. Microcontrollers of the XC166 family. Introductory course for the developer / M. Beach. – М.: ДМК, 2016. – 200 p.

**Шатохин Михаил Сергеевич**, студент 4 курса направления подготовки «Информационно-аналитические системы безопасности», кафедра автоматизированных систем и технологий, институт инженерных и цифровых технологий  
**Жихарев Александр Геннадиевич**, доктор технических наук, доцент, заведующий кафедрой автоматизированных систем и технологий, институт инженерных и цифровых технологий

**Гапицонов Илларион Юрьевич**, соискатель по специальности 3.2.1. Системный анализ, управление и обработка информации, статистика

**Shatokhin Mikhail Sergeevich**, 4th year student in the field of training “Security Information and Analytical Systems”, Department of Automated Systems and Technologies, Institute of Engineering and Digital Technologies

**Zhikharev Alexander Gennadievich**, Doctor of Technical Sciences, Associate Professor, Head of the Department of Automated Systems and Technologies, Institute of Engineering and Digital Technologies

**Gapitsonov Illarion Yuryevich**, applicant in the specialty 3.2.1. Systems analysis, management and information processing, statistics