

А. А. Оленев, В. В. Малиатаки,
Ставропольский государственный педагогический институт

МОДЕЛИРОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ И ПРОСТЕЙШИХ УЗЛОВ ЭВМ В СИСТЕМЕ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ MAPLE

Аннотация

В статье рассматриваются особенности использования системы компьютерной математики Maple в учебном процессе на примере изучения тем, связанных с логическими основами построения ЭВМ и с порядком использования логических схем при таком построении.

Ключевые слова: системы компьютерной математики, Maple, компьютерное моделирование, моделирование логических элементов.

Контактная информация

Оленев Александр Анатольевич,
канд. тех. наук, доцент, доцент кафедры математики и информатики Ставропольского государственного педагогического института; *адрес:* 355029, г. Ставрополь, ул. Ленина, д. 417 «А»; *телефон:* (8652) 56-08-21; *e-mail:* olenevalexandr@gmail.com

Малиатаки Виктория Викторовна,
канд. пед. наук, ст. преподаватель кафедры математики и информатики Ставропольского государственного педагогического института; *адрес:* 355029, г. Ставрополь, ул. Ленина, д. 417 «А»; *телефон:* (8652) 56-08-21; *e-mail:* maliataki@yandex.ru

A. A. Olenev, V. V. Maliataki,
Stavropol State Pedagogical Institute

MODELING LOGICAL ELEMENTS AND THE ELEMENTARY COMPUTER ELEMENTS IN MAPLE

Abstract

The article discusses the features of using the computer mathematics system Maple in the educational process by the example of the study of topics related to the logical framework of building a computer and with the order of the use of logic schemes in this building.

Keywords: computer mathematic system, Maple, computer simulation, modeling of logical elements.

При изучении курсов информационной направленности в старших классах школы и вузах учебное время делится примерно поровну между теоретическими и практическими занятиями. При этом в ходе практических занятий обучающиеся чаще всего используют технические средства обучения, в первую очередь, компьютеры, что обуславливает построение таких занятий в форме компьютерного практикума. Нужно отметить, что компьютер целесообразно использовать не только при изучении технологических тем курсов, но и при изучении теоретических разделов.

Для освоения теоретических и практических тем курсов информационной направленности целесообразно и эффективно использовать в качестве основного методического подхода математическое моделирование. При этом объектами моделирования могут выступать как сами информационные процессы, входящие в информационную технологию (преобразование, прием и передача, обработка информации), так и отдельные элементы, реализующие отдельные функции преобразования информации. Основу этих процессов составляет именно математическое моделирование. При использовании математического моделирования происходит замена исходного объекта его абстрактным образом — вместо объекта используется математическая модель, а дальнейшее изучение модели выполняется с помощью реализуемых на компьютере алгоритмов и программ. Сам метод соединяет в себе достоинства, присущие как теории, так и экспериментам. При использовании математического моделирования работа будет выполняться не с самим объектом (явлением, процессом), а с его моделью, что и дает возможность относительно быстро и без существенных финансовых и временных затрат понять его основные свойства, оценить изменение его состояний в различных ситуациях. Исходя из этого можно достичь высокого обучающего эффекта при условии, если обучающиеся в ходе выполнения практических или лабораторных работ сами создают соответствующие учебные математические или иные модели на компьютере. Для этого могут использоваться как уже знакомые, так и еще неизвестные им технологии [1–4, 9]. Однако для построения хорошей математической модели целесообразно использовать и программное обеспечение соответствующего уровня. Системы компьютерной математики (СКМ), такие как Maple и Mathematica, наиболее полно соответствуют учебным целям в разработке именно математических моделей [1–5, 7].

В данной статье демонстрируются **возможности использования СКМ Maple при изучении тем, связанных с логическими основами построения ЭВМ и с порядком использования логических схем при таком построении** [5–8].

С представленным в работе элементом практикума обучающиеся могут работать как при изучении основных понятий темы, которые рассматриваются в ходе проведения теоретических занятий, так и в ходе практикума [9]. Интуитивно понятные функции Maple, пояснения и описания последо-

вательности действий позволяют учащимся выполнять работу *самостоятельно*.

Предлагаемые для исполнения задания располагаются в порядке возрастания сложности и в необходимом порядке изложения. Например, в учебнике И. Г. Семкина [9] последовательность изложения такая: основные понятия, логические величины, логические операции, логические формулы, простейшие логические схемы.

Задачами практической работы являются:

- моделирование в Maple логических элементов, в том числе с различным числом входов: инвертора, конъюнктора (элемент «И»), дизъюнктора (элемент «ИЛИ»);
- моделирование в Maple логических элементов универсального базиса, в том числе с различным числом входов: штриха Шеффера (элемент «И–НЕ»), стрелки Пирса (элемент «ИЛИ–НЕ»);
- объединение логических элементов в логические схемы в различных базисах в Maple.

Целью работы является обучение построению математических моделей логических элементов вычислительной техники, простейших комбинационных схем в СКА Maple, а также основным методам анализа работы логических схем. Обновление созданной модели или ее математическое описание при этом будет происходить автоматически: при изменении значений входных переменных будут пересчитываться выходные значения функции. В таблице 1 приведен пример такой программы.

Логический элемент «Инвертор» (возможные используемые обозначения: НЕ, not, ¬, \bar{x}). Логический элемент «НЕ» может иметь только один вход и один выход.

С помощью программы (п. 1, 2) можно получить таблицу истинности (п. 3, 4) для логического элемента «НЕ» в различных формах выходного представления (см. сноску к таблице 1).

Если задать значение переменной для инвертора (истинное или ложное), то схема (программа) среагирует и выдаст соответствующий результат. Пункты

5, 6 демонстрируют выходное значение при заданных значениях входных.

Можно сделать и так, чтобы вместо слов ИСТИНА (true) и ЛОЖЬ (false) в ячейках выводились 1 и 0 (демонстрация п. 4, 6).

Используя эту технологию, можно строить и анализировать работу простейших логических схем.

Для освоения данной технологии и создания библиотеки элементов основного базиса необходимо выполнить несколько заданий.

Задание 1. Построение логического элемента «НЕ».

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «НЕ» в двух вариантах (первый вариант — представление таблицы истинности в общем виде, второй вариант — в форме MOD2, см. сноску к таблице 1).

Получите таблицу истинности для элемента «НЕ» для двух форм представления.

Проследите за работой элемента, изменяя значение входного параметра в п. 5, 6.

Задание 2. Построение логического элемента «И».

Логический элемент «Конъюнктор» (возможные используемые обозначения: И, and, &, \wedge).

Простейший логический элемент «И» должен иметь не менее двух входов и один выход.

Программа для моделирования элемента «И» представлена в таб. лице 2.

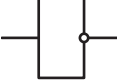
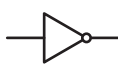
Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от выбранной (заданной) формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «И» в двух вариантах.

Получите таблицу истинности для элемента «И» для двух форм представления, используя функцию Maple TruthTable.

Таблица 1

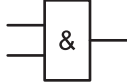

Математическое моделирование элемента «НЕ»

№ п/п	Математическое моделирование	Пояснения	Условно-графическое изображение	
			IEC (ГОСТ)	ANSI
1	<pre>> restart; > with(Logic): > элемент_НЕ:=proc(a) return &not(a); end proc;</pre>	Начало нового сеанса. Подключение библиотеки. Процедура моделирования элемента «НЕ».		
2	<pre>> T1:=элемент_НЕ(x); T1:=&not(x)</pre>	Присвоение элементу «НЕ» имени T1		
3	<pre>> T11:=TruthTable(T1, [x]); T11:=table([true=false, false=true])</pre>	Построение таблицы истинности элемента «НЕ»		
4	<pre>> T12:=TruthTable(T1, [x], form=MOD2); T12:=table([0=1, 1=0])</pre>	Построение таблицы истинности элемента «НЕ» в форме MOD2*		
5	<pre>> T11[true]; false</pre>	Проверка выходного значения отклика при входном значении true		
6	<pre>> T12[0]; 1</pre>	То же в форме MOD2		

* В СКМ Maple при использовании функции TruthTable возможен ввод и вывод на экран в виде значений true и false, а при использовании form=MOD2 значения вводятся и выводятся в виде 0 и 1 соответственно.

Таблица 2

Программа моделирования элемента «И»

Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> > элемент_И:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент И должен иметь не менее двух входов."; end if; result:=a[1] &and a[2]; for i from 3 to nops([a]) do result:=result &and a[i]; end do; return result; end proc: </pre>		

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой, изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Используя приведенную выше программу, произведите моделирование элемента «И» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 3. Построение логического элемента «ИЛИ».

Логический элемент «Дизъюнктор» (возможные используемые обозначения: ИЛИ, or, 1, ∨).

Простейший логический элемент «ИЛИ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «ИЛИ» представлена в таблице 3.

Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от выбранной (заданной) формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «ИЛИ» в двух вариантах.

Получите таблицу истинности для элемента «ИЛИ» для двух форм представления, используя функцию Maple TruthTable.

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой,

изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Создайте элемент «ИЛИ» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 4. Построение логического элемента «И—НЕ».

Логический элемент «Штрих Шеффера» (возможные используемые обозначения: И—НЕ, nand, |).

Простейший логический элемент «И—НЕ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «И—НЕ» представлена в таблице 4.

Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от выбранной (заданной) формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «И—НЕ» в двух вариантах.

Получите таблицу истинности для элемента «И—НЕ» для двух форм представления, используя функцию Maple TruthTable.

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой, изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Таблица 3

Программа моделирования элемента «ИЛИ»

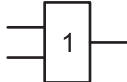

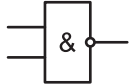
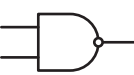
Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> > элемент_ИЛИ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент ИЛИ должен иметь не менее двух входов."; end if; result:=a[1] &or a[2]; for i from 3 to nops([a]) do result:=result &or a[i]; end do; return result; end proc: </pre>		

Таблица 4

Программа моделирования элемента «И—НЕ»

Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre>> элемент_И_НЕ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент И-НЕ должен иметь не менее двух входов."; end if; result:=&not a[1] &or (&not(a[2])); for i from 3 to nops([a]) do result:=result &or (&not(a[i])); end do; return result; end proc:</pre>		

Создайте элемент «И—НЕ» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 5. Построение логического элемента «ИЛИ—НЕ».

Логический элемент «Стрелка Пирса» (возможные используемые обозначения: ИЛИ—НЕ, \downarrow).

Простейший логический элемент «ИЛИ—НЕ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «ИЛИ—НЕ» представлена в таблице 5.

Значения входного параметра могут задаваться как true и false или числами 1 и 0 соответственно, это зависит от формы представления.

Воспроизведите путем копирования программы в рабочее поле СКМ Maple логический элемент «ИЛИ—НЕ» в двух вариантах.

Получите таблицу истинности для элемента «ИЛИ—НЕ» для двух форм представления, используя функцию Maple TruthTable.

Произведите анализ работы логического элемента по таблице истинности и проследите за его работой, изменяя значения входных параметров, как в п. 5, 6 таблицы 1.

Создайте элемент «ИЛИ—НЕ» на три или четыре входа. Получите таблицу истинности и проследите за изменением выходных значений при изменении входных значений.

Задание 6. Организация связи между элементами.

Создайте элемент «И—НЕ» с помощью созданных ранее программ для моделирования элементов «И», «ИЛИ» и «НЕ».

Получите таблицы истинности и сравните с результатами (таблицами истинности) для элементов «И—НЕ» сделайте вывод.

Элемент «И» имеет два входа — x_1 и x_2 . Присвоим этому элементу имя I_2 , а элементу «НЕ» — I_3 . Выход элемента I_2 будет являться входом элемента «НЕ».

Проанализируйте полученную таблицу истинности.

Произведите сравнение с таблицей истинности, полученной для несоставного элемента «И—НЕ».

По аналогии составьте схему для элемента «НЕ—И» и получите таблицу истинности.

Сравните таблицы истинности и сделайте вывод.

Согласно классической пропозициональной логике, справедливы следующие соотношения:

$$\begin{aligned} \text{не } (x \text{ и } y) &= (\text{не } x) \text{ или } (\text{не } y); \\ \text{не } (x \text{ или } y) &= (\text{не } x) \text{ и } (\text{не } y). \end{aligned}$$

Математически это можно представить так:

$$\begin{aligned} \neg(x \wedge y) &= \neg x \vee \neg y; \\ \neg(x \vee y) &= \neg x \wedge \neg y. \end{aligned}$$

Самостоятельно составьте логическую схему для элемента «ИЛИ—НЕ» и выполните для него действия, идентичные предусмотренным для изучения составного элемента «И—НЕ».

Таблица 5

Программа моделирования элемента «ИЛИ—НЕ»

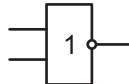

Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre>> Элемент_ИЛИ_НЕ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент ИЛИ-НЕ должен иметь не менее двух входов."; end if; result:=a[1] &nor a[2]; for i from 3 to nops([a]) do result:=result &and (&not(a[i])); end do; return result; end proc:</pre>		

Таблица 6

Пример моделирования составных элементов

Математическое моделирование	Результат	Реализуемые логические элементы	
$I2 := \text{элемент_И}(x1, x2);$ $I3 := \text{элемент_НЕ}(I2);$	$I2 := x1 \ \& \ x2$ $I3 := \text{not}(x1 \ \& \ x2)$	Элемент «И»	
		Соединение элемента «И» и элемента «НЕ» — элемент «И–НЕ»	
$\text{TruthTable}(I3, [x1, x2]);$	$\text{table}([(false, false)=true,$ $(true, true)=false,$ $(false, true)=true,$ $(true, false)=true])$	Элемент «И–НЕ»	

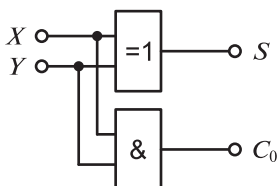
Контрольное задание.

Постройте логические схемы, соответствующие приведенным в таблице 7 логическим формулам.

Для каждой схемы проверьте правильность ее работы путем построения таблицы истинности.

Задание 7. Построение одноразрядного полусумматора.

Полусумматор реализует сложение двух двоичных слагаемых X и Y с получением на выходе младшего разряда их суммы S и старшего (переносимого) разряда C_0 .



На рисунке приведен внешний вид логической схемы полусумматора, двум выходам которого (C_0 и S) соответствуют следующие логические формулы:

$$C_0 = X \ \& \ Y;$$

$$S = X \ \text{XOR} \ Y.$$

Таким образом, схему одноразрядного полусумматора можно реализовать, используя два логических элемента: логического элемента «И», имеющего два входа, и логического элемента «исключающее ИЛИ».

7.1. Построение логического элемента «исключающее ИЛИ».

Логический элемент «исключающее ИЛИ» (возможные используемые обозначения: сложение (сумма) по модулю 2, исключающее_ИЛИ, неравнозначность, MOD 2, =1).

Простейший логический элемент «исключающее ИЛИ» должен иметь не менее двух входов и один выход.

Программа моделирования элемента «исключающее ИЛИ» представлена в таблице 8.

Сопоставьте полученные результаты с таблицей истинности.

7.2. Моделирование логической схемы полусумматора.

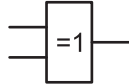
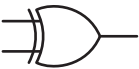
Произведите моделирование логической схемы одноразрядного полусумматора с помощью полученной ранее программы для логического элемента «И» и полученного в пункте 7.1 логического элемента «исключающее ИЛИ». Для организации связей воспользуйтесь рекомендациями задания 6.

Таблица 7

Примеры использования различных обозначений логических функций

Вариант 1 (чтение)	Вариант 2
не x или y	$\neg x \ \text{or} \ y$
x и y или z	$x \ \text{and} \ y \ \text{or} \ z$
x или y и z	$x \ \text{or} \ y \ \text{and} \ z$
не x или y и z	$\neg x \ \text{or} \ y \ \text{and} \ z$
не (x и y и z)	$\neg(x \ \text{and} \ y \ \text{and} \ z)$
(x или y) и (z или y)	$(x \ \text{or} \ y) \ \text{and} \ (z \ \text{or} \ y)$
не(x или y) и (z или y)	$\neg(x \ \text{and} \ y) \ \text{or} \ (z \ \text{or} \ x)$
((не x и y) или x) и не y	$((\neg x \ \text{and} \ y) \ \text{or} \ z) \ \text{and} \ \neg y$

Программа моделирования элемента «Исключающее ИЛИ»

Математическое моделирование	Условно-графическое изображение	
	IEC (ГОСТ)	ANSI
<pre> элемент_ИСКЛИЛИ:=proc(a::seq(anything)) local result, i; if nops([a])<2 then error "Элемент ИЛИ должен иметь не менее двух входов."; end if; result:=a[1] &xor a[2]; for i from 3 to nops([a]) do result:=result &xor a[i]; end do; return result; end proc: </pre>		

Постройте таблицу истинности для выходных параметров C_0 и S полусумматора.

Протестируйте работу логической схемы, перебрав все варианты значений входных параметров X и Y .

Задание 8 (повышенной сложности). Построение одноразрядного полусумматора в универсальных базисах.

Произведите моделирование логической схемы полусумматора в универсальных базисах — штрих Шеффера и стрелка Пирса.

Постройте таблицу истинности для выходных параметров C_0 и S полусумматора в одном из универсальных базисов — штрих Шеффера и стрелка Пирса.

Протестируйте работу логической схемы, перебрав все варианты значений входных параметров X , Y , C_0 .

Сопоставьте полученные результаты с таблицей истинности.

Список использованных источников

1. *Говорухин В. И., Цибулин В. Г.* Введение в Maple. Математический пакет для всех. М.: Мир, 1997.
2. *Дьяконов В. П.* Maple 10/11/12/13/14 в математических расчетах. М.: СОЛОН-Пресс, 2011.
3. *Кирсанов М. Н.* Maple 13 и Maplelet. Решение задач механики. М.: Физматлит, 2010.

4. *Красильников В. В., Оленев А. А., Тоискин В. С.* Проверка знаний по теме «Множество» на основе системы компьютерной математики Maple // Актуальные вопросы инженерного образования — 2015: Сборник научных трудов международной научно-методической конференции (Октябрьский, 27 ноября 2015 г.). Екатеринбург: Универсальная Типография «Альфа Принт», 2016.

5. *Малиатаки В. В., Медведева Л. М., Оленев А. А.* Совершенствование математической подготовки учителя в вузе на основе использования СКА Maple // Актуальные вопросы инженерного образования — 2015: Сборник научных трудов международной научно-методической конференции (Октябрьский, 27 ноября 2015 г.). Екатеринбург: Универсальная Типография «Альфа Принт», 2016.

6. *Оленев А. А.* Особенности реализации алгоритма Евклида в Maple // Актуальные вопросы инженерного образования — 2014: Сборник научных трудов международной научно-методической конференции (Октябрьский, 13 декабря 2014 г.). Челябинск: АРКАИМ, 2014.

7. *Оленев А. А., Сагдеев К. М.* Абстрактная и компьютерная алгебра. Ч. 1. Абстрактная алгебра в Maple: учебно-методическое пособие. Ставрополь: Бюро новостей, 2014.

8. *Сагдеев К. М., Оленев А. А.* Подход к математическому моделированию процесса выработки управляющего воздействия в организационных системах // Современные проблемы науки и образования. 2012. № 3.

9. *Семакин И. Г., Шеина Т. Ю., Шестакова Л. В.* Информатика и ИКТ. Профильный уровень: учебник для 10 класса. М.: БИНОМ. Лаборатория знаний, 2010.